



Universidad Internacional de La Rioja
Escuela Superior de Ingeniería y Tecnología - ESIT

Máster Universitario en Internet De Las Cosas / INTERNET OF
THINGS (IoT)

Propuesta de estudio del sistema de aparcamiento tarifado utilizando un modelo de simulación sensorial

Trabajo fin de estudio presentado por:	Casas i del Olmo, Ricard
Tipo de trabajo:	Planteamiento de un proyecto de IoT
Director:	Dr. Alonso Rincón, Ricardo Serafín
Codirector:	Dr. Pous Sabadi, Carles
Fecha:	20/7/2023

Resumen

Las Smart Cities se enfrentan constantemente a retos de movilidad, como aglomeraciones y contaminación. Enfoques para gestionar el tráfico a menudo descuidan las tecnologías generadoras de datos. Es crucial considerar estos elementos para mejorar eficazmente la movilidad en una Smart City.

En consecuencia, el ayuntamiento de Girona plantea un sistema de aparcamiento tarifado en plaza Catalunya empleando tecnologías IoT por lo que pretende saber el uso e impacto que van a tener previamente a su despliegue. Por esta razón, con la herramienta ns-3, se realizan escenarios de simulación junto al protocolo de comunicaciones LoRa/LoRaWAN, el dispositivo seleccionado y el entorno de la ciudad.

El conocimiento generado valida la implementación del dispositivo sensorial Winext con una configuración LoRa/LoRaWAN con la potencia de transmisión (Tx) en 14 dBm, el factor de dispersión a SF7 junto al ADR (*Adaptive Data Rate*) y ACK (*Acknowledge*) desactivados. Además, se presumen errores de transmisión inferiores al 1% juntamente con una vida útil máxima del dispositivo oscilando los cinco años.

Finalmente, se concluye que la implantación del sensor y las comunicaciones LoRa cumplen con las exigencias del ayuntamiento de Girona, generando un marco de conformidad para ello.

Palabras clave: Smart Cities, simulación, protocolo de comunicaciones LoRa/LoRaWAN, aparcamiento, sensores.

Abstract

Smart Cities constantly face mobility challenges, such as congestion and pollution. Approaches which manage traffic overlook data-generating technologies often. Therefore, it is crucial to consider these elements to enhance mobility in a Smart City effectively.

Consequently, the Girona city council proposes a time-based parking system in Catalunya Square, employing IoT technologies to understand the service and impact it will have before its implementation. For this reason, simulation scenarios are implemented using the ns-3 tool, in conjunction with the LoRa/LoRaWAN communication protocol, the selected device, and the city environment.

The knowledge generated validates the implementation of the Winext sensor device with a LoRa/LoRaWAN configuration, like a transmission power (Tx) set at 14 dBm, a spreading factor of SF7, with the ADR (Adaptive Data Rate) and the ACK (Acknowledge) disabled. Additionally, transmission errors below 1% are presumed, with the maximum device lifetime of approximately five years.

In conclusion, the sensor's deployment and LoRa communications meet the demands of the Girona city council, creating a framework for this purpose.

Keywords: Smart Cities, simulation, LoRa/LoRaWAN communication protocol, parking, sensors.

Agradecimientos

El trabajo aquí presente ha requerido un esfuerzo inconmensurable, el cual no hubiera sido posible sin la ayuda incondicional de instituciones, profesionales, amigos, familiares y finalmente, de mi pareja. Es por ello por lo que quiero darles mi eterna gratitud.

En primer lugar, agradecer la oportunidad que me han otorgado las instituciones de la UNIR (Universidad Internacional de La Rioja) y la UdG (Universidad de Girona) para poder realizar este tipo de trabajo en el máster oficial del Internet de las Cosas.

A mi director Dr. Ricardo Serafín Alonso Rincón, por todo el seguimiento y excepcionales recomendaciones personales, las cuales me han permitido perfilar y marcar correctamente el hito final en este trabajo.

Al codirector Dr. Carles Pous Sabadi, al cual le tengo una gran admiración y estima, principal causante de que desarrollara este trabajo. No solo eso, su capacidad analítica y de razonamiento han permitido que pudiese llegar tan lejos. Aparte, me gustaría añadir, de forma personal, mi eterna gratitud como la persona que es y la relación que se ha formado a lo largo de estos años, por entenderme como soy y la confianza depositada. Relación la cual me gustaría que durase para siempre, tanto a nivel profesional como de amistad.

A mi familia, en concreto a mis padres, siempre adaptándose y entendiendo mi manera de ser, apoyándose en todo y en cualquier lugar donde vaya y esté. Agradecerles sus puntos de vista, su experiencia y esfuerzo, fundamentales para mí a lo largo de mi vida, siempre que los he necesitado han estado allí.

Por último y más importante a mi pareja, este año en especial hemos pasado por muchos retos juntos y este trabajo ha hecho que fueran un poco más difíciles. Aun con ello, tu paciencia, tus palabras. Asimismo, has hecho que pudiera llegar más lejos de lo que hubiera esperado, haciéndome sentir muy agradecido de que hayas permanecido a mi lado todo este tiempo, nunca lo olvidaré. No sé qué nos deparará el futuro, pero pase lo que pase, haré todo lo posible para seguir estando a tu lado.

Índice de contenidos

1. Introducción	19
1.1. Motivación	19
1.2. Planteamiento del trabajo	22
1.3. Estructura del trabajo	23
2. Contexto y estado del arte	25
2.1. Smart Cities: contexto e implicaciones actuales	25
2.2. Proyectos enlazados y relevantes.....	31
2.2.1. Implementaciones dispares en el ámbito de aparcamientos inteligentes	31
2.2.2. Modelos de simulación para escenarios dispares.....	33
2.2.3. Modelos de simulación en el ámbito de aparcamientos inteligentes	35
2.3. Conclusiones	37
3. Descripción general de la contribución.....	39
3.1. Objetivos	39
3.2. Metodología del trabajo	40
3.2.1. Fase 1: Definición del caso de uso.....	40
3.2.2. Fase 2: Selección de la información necesaria	41
3.2.3. Fase 3: Diseñar e implementar la simulación.....	43
3.2.4. Fase 4: Análisis de los resultados	46
4. Desarrollo específico de la contribución.....	49
4.1. Fase 1: Definición del caso de uso	49
4.1.1. Zona de aplicación del proyecto.....	49
4.1.2. Dispositivos y tecnología	53
4.1.3. Objetivos y requerimientos del ayuntamiento	60

4.2.	Fase 2: Selección de la información.....	61
4.2.1.	Estudio del sensor candidato	62
4.2.2.	Métricas y características asociadas a la simulación.....	69
4.3.	Fase 3: Diseño e implementación la simulación.....	71
4.3.1.	Modelaje de los escenarios base ns-3.....	71
4.3.2.	Implementación para los distintos escenarios.....	83
4.4.	Fase 4: Análisis de los datos.....	89
4.4.1.	Creación del digestor de datos de Python	91
4.4.2.	Fiabilidad del primer escenario	92
4.4.3.	Fiabilidad del segundo escenario	94
4.4.4.	Análisis previo al escenario del consumo.....	99
4.4.5.	Consumo del tercer escenario para el dispositivo	104
5.	Conclusiones y trabajos futuros.....	107
5.1.	Conclusiones	107
5.2.	Líneas de trabajo futuras	109
	Referencias bibliográficas.....	111
	Anexo A. Planos.....	123
A.1.	Situación.....	123
	Anexo B. Datos de implementación	124
B.1.	Ubicaciones de los dispositivos	124
B.2.	Métricas y características de los dispositivos.....	125
B.3.	Datos de las simulaciones	127
	Anexo C. Visualizaciones y gráficos	131
	Anexo D. Código implementado.....	145
D.1.	Código de los escenarios simulados	145

D.2.	Representación de los edificios	155
D.3.	Digestor de datos para los distintos escenarios	156

Índice de figuras

Figura 1. <i>Descripción del ecosistema IoT genérico.</i>	20
Figura 2. <i>Evolución de la población rural y la residente en ciudades a nivel mundial.</i>	25
Figura 3. <i>Dominio de aplicaciones IoT para Smart Cities.</i>	26
Figura 4. <i>Reducción del transporte por carretera durante el COVID-19 en distintos países.</i> ...	29
Figura 5. <i>Relación entre palabras clave en movilidad y transporte.</i>	30
Figura 6. <i>Demora media según número de UAV y servidores FOG.</i>	34
Figura 7. <i>Escalabilidad y tiempo de respuesta del proyecto de simulación para São Paulo.</i> ...	36
Figura 8. <i>Guía de las distintas fases para este proyecto de simulación de aparcamiento.</i>	40
Figura 9. <i>Comparativa entre lenguajes de programación y la mediana en líneas de código.</i> .	47
Figura 10. <i>Área de despliegue sensorial en Plaza Catalunya.</i>	50
Figura 11. <i>Mapa de la ubicación de los sensores en Plaza Catalunya.</i>	51
Figura 12. <i>Ubicación de la Gateway en la Torre de Alfons XII de la ciudad de Girona.</i>	52
Figura 13. <i>Trazado longitudinal de Plaza Catalunya y colindantes.</i>	52
Figura 14. <i>Perfil longitudinal de Plaza Catalunya y colindantes.</i>	53
Figura 15. <i>Modelo AN-101D Winext.</i>	55
Figura 16. <i>Modelo Parking Lot Sensor BOSCH.</i>	56
Figura 17. <i>Componentes del dispositivo BOSCH.</i>	57
Figura 18. <i>Modelo Dual Libelium.</i>	59
Figura 19. <i>Pasarela MultiTech Conduit.</i>	60
Figura 20. <i>Resultados de las desviaciones generadas por los modelos de propagación de la señal en la ciudad de Brno.</i>	72
Figura 21. <i>Captura del consumo del dispositivo sensorial, envío más recepción ACK en Rx1.</i> .	81
Figura 22. <i>Arquitectura de gestión y conectividad de borde planteada para la simulación.</i> ...	84
Figura 23. <i>Visualización de los edificios creados.</i>	88

Figura 24. <i>Gráfico PER-Periodo para el primer escenario.</i>	93
Figura 25. <i>Gráfico PER-Periodo para el primer escenario con vehículo.</i>	93
Figura 26. <i>Gráfico PER-Periodo para el segundo escenario.</i>	95
Figura 27. <i>Gráfico PER-Periodo para el segundo escenario con coche.</i>	96
Figura 28. <i>Mediana del PER según SF en el segundo escenario.</i>	97
Figura 29. <i>Desviación del PER según SF en el segundo escenario.</i>	98
Figura 30. <i>Gráfico PER-Periodo para el segundo escenario con ADR y ACK.</i>	99
Figura 31. <i>Mapa de la posición de los dispositivos sensoriales en plaza Catalunya según el Rx sin vehículo.</i>	100
Figura 32. <i>Mapa de la posición de los dispositivos sensoriales en plaza Catalunya según el Rx con vehículo.</i>	101
Figura 33. <i>Visualización de los promedios del Rx según ED.</i>	102
Figura 34. <i>Datos cruzados del Tx según PER y Rx sin vehículo según configuración analizada.</i>	103
Figura 35. <i>Datos cruzados del Tx según PER y Rx con vehículo según configuración analizada.</i>	103
Figura 36. <i>Vida útil del dispositivo sensorial respecto los periodos entre eventos.</i>	105
Figura 37. <i>Plano de situación para el proyecto de estacionamiento tarifado de Plaza Catalunya.</i>	123
Figura 38. <i>Conjunto de gráficos comparativos PER según Periodo para el primer escenario.</i>	131
Figura 39. <i>Conjunto de gráficos comparativos PER según Periodo para el segundo escenario.</i>	131
Figura 40. <i>Comparativos del PER según Periodo para el primer y segundo escenario.</i>	132
Figura 41. <i>Comparativo del DC según Periodo y SF7 para el primer y segundo escenario.</i> ...	132
Figura 42. <i>Comparativo del DC según Periodo y SF8 para el primer y segundo escenario.</i> ...	133
Figura 43. <i>Comparativo del DC según Periodo y SF9 para el primer y segundo escenario.</i> ...	133

Figura 44. <i>Comparativo del DC según Periodo y SF10 para el primer y segundo escenario.</i>	134
Figura 45. <i>Comparativo del DC según Periodo y SF11 para el primer y segundo escenario.</i>	134
Figura 46. <i>Comparativo del DC según Periodo y SF12 para el primer y segundo escenario.</i>	135
Figura 47. <i>Variación del Tx según funcionalidad ADR con periodo de 900 segundos.</i>	135
Figura 48. <i>Variación del Tx según funcionalidad ADR con periodo de 720 segundos.</i>	136
Figura 49. <i>Variación del Tx según funcionalidad ADR con periodo de 540 segundos.</i>	136
Figura 50. <i>Variación del Tx según funcionalidad ADR con periodo de 360 segundos.</i>	137
Figura 51. <i>Variación del Tx según funcionalidad ADR con periodo de 180 segundos.</i>	137
Figura 52. <i>Variación del Tx según funcionalidad ADR con periodo de 20 segundos.</i>	138
Figura 53. <i>Variación del SF según funcionalidad ADR con periodo de 900 segundos.</i>	138
Figura 54. <i>Variación del SF según funcionalidad ADR con periodo de 720 segundos.</i>	139
Figura 55. <i>Variación del SF según funcionalidad ADR con periodo de 540 segundos.</i>	139
Figura 56. <i>Variación del SF según funcionalidad ADR con periodo de 360 segundos.</i>	140
Figura 57. <i>Variación del SF según funcionalidad ADR con periodo de 180 segundos.</i>	140
Figura 58. <i>Variación del SF según funcionalidad ADR con periodo de 20 segundos.</i>	141
Figura 59. <i>Mapa del SNR de los sensores sin vehículo en Plaza Catalunya.</i>	141
Figura 60. <i>Mapa del SNR de los sensores con vehículo en Plaza Catalunya.</i>	142
Figura 61. <i>Visualización de los promedios del SNR según ED según configuración del análisis.</i>	142
Figura 62. <i>Relación del Tx 2 y 4 según Rx y ED según configuración del análisis.</i>	143
Figura 63. <i>Relación del Tx 6 y 8 según Rx y ED según configuración del análisis.</i>	143
Figura 64. <i>Relación del Tx 10 y 12 según Rx y ED según configuración del análisis.</i>	144
Figura 65. <i>Relación del Tx 14 según Rx y ED según configuración del análisis.</i>	144

Índice de tablas

Tabla 1. <i>Palabras clave asociadas a 166 definiciones de Smart Cities (2016)</i>	27
Tabla 2. <i>Métricas destacables para la generación de conocimiento.</i>	42
Tabla 3. <i>Comparativa de simuladores para este trabajo.</i>	43
Tabla 4. <i>Características de los distintos escenarios para la simulación de Plaza Catalunya.</i> ..	45
Tabla 5. <i>Comparativa de herramientas de visualización.</i>	48
Tabla 6. <i>Dispositivos propuestos para el proyecto.</i>	54
Tabla 7. <i>OKR y KPI del ayuntamiento de Girona.</i>	61
Tabla 8. <i>Modelos de propagación de la señal nombrados por Stusek et al. (2020) e implementados en el ns-3.</i>	73
Tabla 9. <i>Datos de la desviación estándar según el RSSI extraído de las pruebas realizadas con LoRa y el sensor.</i>	78
Tabla 10. <i>Comparativa de los consumos para la implementación del modelo de energía de las comunicaciones.</i>	83
Tabla 11. <i>Periodos de duración de la vida útil del dispositivo sensorial.</i>	104
Tabla 12. <i>Ubicación de los dispositivos en Plaza Catalunya.</i>	124
Tabla 13. <i>Características de los sensores candidatos propuestos por el Ayuntamiento.</i>	125
Tabla 14. <i>Métricas de los sensores candidatos propuestos por el Ayuntamiento.</i>	125
Tabla 15. <i>Características de la Gateway propuesta por el Ayuntamiento.</i>	125
Tabla 16. <i>Payload dispositivo Winext.</i>	126
Tabla 17. <i>Payload dispositivo BOSCH.</i>	126
Tabla 18. <i>Payload dispositivo Libelium.</i>	126
Tabla 19. <i>Sensibilidades y Duty Cycle de los dispositivos según Payload de Winext.</i>	127
Tabla 20. <i>Canales de comunicación LoRaWAN.</i>	127
Tabla 21. <i>Periodos de tiempo empleados para las simulaciones.</i>	127

Tabla 22. <i>Set de datos (Dataset) generado por el digester de datos de Python.</i>	128
Tabla 23. <i>Valores PER-Periodo del primer escenario.</i>	129
Tabla 24. <i>Valores PER-Periodo del segundo escenario.</i>	129
Tabla 25. <i>Promedios del Rx y SNR de los ED con y sin vehículo.</i>	130

Índice de listados

Listado 1. <i>Código de configuración de la simulación.</i>	84
Listado 2. <i>Código de elaboración de la configuración para el modelo de propagación de la señal.</i>	85
Listado 3. <i>Extracto de código de la pérdida de la potencia de la señal.</i>	85
Listado 4. <i>Código de configuración del ED dentro de la simulación.</i>	85
Listado 5. <i>Creación de los canales para LoRa/LoRaWAN.</i>	86
Listado 6. <i>Código de creación del escenario.</i>	86
Listado 7. <i>Distribución de los edificios dentro de la simulación.</i>	88
Listado 8. <i>Configuración de los consumos en el tercer escenario.</i>	89
Listado 9. <i>Ventanas de tiempo para la recepción de mensajes.</i>	89
Listado 10. <i>Definición de las palabras clave para la búsqueda de información del script de Python.</i>	91
Listado 11. <i>Algoritmo de computación del SNR.</i>	92
Listado 12. <i>Etiquetas de estructuración de los datos.</i>	92

1. INTRODUCCIÓN

En este capítulo se emprende la casuística principal del trabajo, detallando distintos elementos separados en tres secciones.

La Sección 1.1 organiza la idea principal detrás de dicha implementación de sistemas IoT empleando simulaciones hasta llegar al caso concreto de este trabajo. A continuación, la Sección 1.2 describe el móvil por el cual se realiza dicho trabajo final de máster. Finalizando con la Sección 1.3 en la que se enumera la estructura del trabajo de manera descriptiva, detallando aquellos aspectos más característicos.

1.1. Motivación

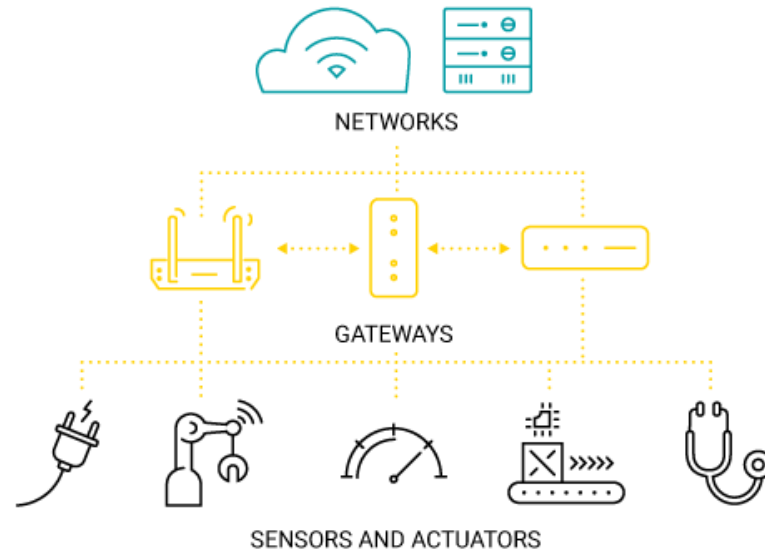
El Internet de las cosas o IoT representa un componente crucial para la transformación digital, las posibilidades e impacto que tiene la implantación de dicha tecnología en el día a día de las personas y en los modelos de negocios son numerosas (Allam & Dhunny, 2019). En este sentido, Elnashar (2019) va más allá y argumenta que el IoT representa la vanguardia para la última revolución industrial y da paso a nuevas tecnologías de tratamiento de datos (Big Data) y análisis (ML) las cuales, representan los futuros revolucionarios para la década por venir.

Malhotra (2021) define la dinámica principal del IoT como el establecimiento en el uso de millones, e incluso de billones de dispositivos a finales del año 2020, todos ellos interconectados, unidos en un ecosistema que engloba una gran variedad de estos a través de protocolos de comunicación ya sea por cable o inalámbricos. Además, el mismo autor añade que el ecosistema más básico para un sistema IoT queda representado en la Figura 1. De esta forma se crea una red de malla o “Mesh” de dispositivos generadores de datos e información, la cual pretende ser aprovechada para generar ciertos beneficios y comforts para los humanos.

La transformación digital no solo afecta a la industria, sino también a los elementos constitutivos de la ciudadanía como lo son las Smart Cities. Saravanan (2019) argumenta que las Smart Cities pueden aprovechar distintos avances revolucionarios en diversos campos de desarrollo del IoT para proveer un mejor servicio a la ciudadanía. Además, alienta a los responsables y dirigentes de dichas Smart Cities a emplear esta oportunidad para tomar

caminos estratégicos preventivos, todo lo contrario que hasta ahora, en que se creaban procesos de toma de decisiones (*Decision-Making*) después de que se generasen incidentes.

Figura 1. Descripción del ecosistema IoT genérico.



Fuente: IoT Ecosystem, s. f.

En el entorno de Smart Cities se aprecian una multitud de áreas en las que se pueden implementar sistemas IoT. Entre todas las existentes, la gestión urbanística, concretamente la gestión del tráfico lleva asociada diversas controversias.

En el trabajo de Chang (2020) plantea el paradigma de los beneficios que pueden aportar la implementación de sistemas IoT e ICT (*Information and Communications Technology*) para la gestión del tráfico. De hecho, en dicho trabajo se argumentan razones por las cuales se asocia la necesidad de crear sistemas de gestión para reducir o mejorar el flujo del tránsito, ya que, debido a las necesidades del trabajo de la ciudadanía, pero también el uso de edificios públicos, tales como hospitales, bibliotecas, entre otros, se incrementa el uso del transporte y por ello las emisiones de CO₂. En este sentido también se generan mayores retardos en el tráfico, empeorando las condiciones generales de la población.

Estos problemas son más notorios cuanto mayor se trata la ciudad, un ejemplo es Beijing, donde los retardos provocados por las congestiones de tráfico superaron la media de 256 h por conductor en el año 2017 (Zhao & Hu, 2019). Además, se añade el factor social y característico de cada ciudad o país, ya que existe una relación con los horarios establecidos

como el trabajo, eventos sociales, entre otros, y los barrios donde se encuentren dichos trabajos o lugares públicos o privados.

En esta línea, muchas ciudades deciden solventar dicha problemática abordando la gestión de los estacionamientos como parte de la solución final. En el trabajo de Nagowah (2019) se presentan un modelo de gestión de un aparcamiento, en este se establecen procesos para la ingestión y tratamiento de los datos, pero no se establecen parámetros de evaluación para la implementación de la tecnología sensorial o de comunicaciones. Es por ello por lo que encontrar una solución para la evaluación e implementación de dichas tecnologías IoT (dispositivos sensoriales y protocolos de comunicación) es tan importante como la gestión que se le dé más adelante ya que aporta valor y cimienta firmemente las bases para su implementación.

Para que el IoT pueda realizar su función dentro de las Smart Cities es necesario que genere gran cantidad de datos, los cuales puedan ser transportados de manera segura y a través de los protocolos adecuados. Es por ello por lo que Saravanan (2019) alega el empleo de dispositivos "Smart IoT" para que estos ayuden en la generación de nuevas soluciones que aporten valor a la ciudadanía. Con respecto a esto, es necesario emplear multitud de tecnologías, algunas novedosas, que sean capaces de gestionar dichos datos, así como de los dispositivos que los generan, empleando el hardware y software necesario en cada ocasión, proveyendo adaptabilidad y disponibilidad.

Algunas de las comunicaciones más importantes y destacadas por Elnashar (2019) son las M2M (*Machine to Machine*), como el LTE/4G/3G/2G, enfocadas en un entorno donde las necesidades consisten en traspasar grandes cantidades de datos sin tener en cuenta el consumo de los dispositivos, y IoT, como el NB-IoT y LTE-M, entre otras.

En el último caso, se antepone el bajo consumo que puedan requerir este tipo de comunicaciones, ya que los dispositivos para los cuales han sido diseñados no generan un volumen tan alto de datos (menor ancho de banda) y normalmente se encuentran en entornos sin alimentación. Este hecho crea la necesidad de sistemas de almacenamiento de energía para funcionar, como baterías u otros. Además, el hecho de que puedan trasladarse e instalarse en cualquier sitio supone una mayor distancia entre el emisor de datos y el remitente, por ello, dichas comunicaciones deben tener alcances mucho mayores. Por esta razón, las comunicaciones con dispositivos de estas características requieren emplear las

comunicaciones de tipo LPWAN (*Low Power Wide Area Network*) o redes de bajo consumo y rango amplio como LoRa, Sigfox, LTE-M, entre otras.

En esta línea existe una motivación por parte del autor para involucrarse en un proyecto de IoT y Smart Cities que ayude a dar una mayor gestión y control del tráfico. Desplegar un sistema IoT en una ciudad donde su prioridad es mejorar la congestión de los vehículos y por consiguiente a la salud de su ciudadanía resulta clave.

1.2. Planteamiento del trabajo

La distinta problemática asociada en la gestión y control del tráfico dentro del paradigma de Smart Cities y IoT supone un reto que requiere especial atención a los problemas de salud, congestiones, entre otras y determinar su resolución es imperante.

Este Trabajo de Fin de Máster propone evaluar la proposición del sistema de aparcamiento tarifado del Ayuntamiento de Girona. El sistema de aparcamiento, el cual se instalaría en una zona cercana al casco antiguo, en concreto se trata de Plaza Catalunya, pretende reducir los tiempos de búsqueda de aparcamiento en la zona, aumentando la fluidez del tráfico y, por consiguiente, reduciendo la contaminación (acústica y ambiental) provocada por ello.

Por ende, se expresa la intención para ejercer un rol el cual permita establecer un criterio de evaluación de la tecnología que se pretende implementar. El Ayuntamiento determina a su juicio cuál es la implementación de los posibles dispositivos y el protocolo de comunicación, determinando una lista de posibles dispositivos sensoriales para la detección de los vehículos y el uso de LoRa para las comunicaciones. Por este motivo, este trabajo requiere de la recolección y posterior evaluación de los datos sensoriales y del protocolo de comunicación para determinar cuál de los sensores propuestos encuentra un mayor encaje para el proyecto.

Aparte, el hecho de que las comunicaciones LoRa se establezcan como predeterminadas hace que la evaluación requiera del uso de simulaciones para determinar su impacto en un proyecto de estas características. En virtud de ello, se trata de hallar posibles limitaciones, características y modos de utilidad, determinando de esta manera modelos de configuración para el protocolo de comunicaciones.

1.3. Estructura del trabajo

El Trabajo de Fin de Máster desarrollado se compone con una totalidad de cinco capítulos, esta sección se halla en el primero de ellos. A partir de este punto se describen la totalidad de estos:

- **Capítulo 1. Introducción:** En este se encuentra la disposición de la motivación, donde se argumenta la problemática principal del proyecto. A continuación, se encuentra el planteamiento por el cual se pretende solventar dicha problemática, finalizando con el orden que sigue este documento.
- **Capítulo 2. Contexto y estado del arte:** Este capítulo contempla el contexto e implicaciones generales de este proyecto, como la coyuntura actual de las Smart Cities. Dentro del paradigma creciente de la Movilidad, como el tráfico y los efectos perjudiciales en las personas, se introducen ejemplos comparativos en escenarios distintivos y parentescos que muestran diversas soluciones desde el ámbito de las simulaciones. Incluso, en enfoques más directos con el uso de Blockchain o técnicas de Inteligencia Artificial.
- **Capítulo 3. Descripción general de la contribución:** Partiendo con la descripción del objetivo principal, se enumeran los objetivos específicos y los pasos necesarios que se pretenden seguir para conseguirlo. En este sentido, la evaluación del sensor para el caso de uso del aparcamiento y la simulación que se realiza con la herramienta ns-3, permite determinar una configuración específica para estos dispositivos y evaluar el comportamiento de las comunicaciones.
- **Capítulo 4. Desarrollo específico de la contribución:** Los pasos descritos en el capítulo anterior son llevados a cabo. Por esta razón, en este capítulo se encuentran las distintas fases de implementación imprescindibles de manera detallada.
 - **Sección 4.1. Fase 1: Definición del caso de uso:** En esta primera fase se desarrolla la búsqueda, revisión y contraste de la información para todos aquellos aspectos necesarios para la selección del sensor y la realización de la simulación. Finalizando con un listado de todos aquellos OKR (*Objective Key Results*) y KPI (*Key Performance Indicators*) por los cuales el ayuntamiento pretende encontrar una respuesta.

- **Sección 4.2. Fase 2: Selección de la información:** La segunda fase lleva a cabo dos procesos, en primer lugar, una evaluación exhaustiva en la selección de los sensores candidatos propuestos y en paralelo una valoración de las métricas necesarias para la realización de la simulación.
- **Sección 4.3. Fase 3: Diseño e implementación la simulación:** Como indica el título, con el sensor seleccionado, se procede a la selección del algoritmo de propagación de la señal y la reproducción del máximo de factores ambientales y tecnológicas que requiere una simulación de protocolos de transmisiones.
- **Sección 4.4. Fase 4: Análisis de los datos:** En esta última fase se elabora la totalidad del estudio junto a los resultados obtenidos de la simulación para los distintos escenarios que se plantean, indicando de manera argumental las razones de estos.
- **Capítulo 5. Conclusiones y trabajos futuros:** En este capítulo se evalúa la relación entre los resultados finales de este proyecto junto a los objetivos marcados. Por ende, se interpretan de pertinente los distintos hitos específicos junto a los OKR y KPI logrados o no. Además, se añade una sección para los trabajos futuros en los cuales este proyecto puede desarrollarse para mejorar o expandir sus resultados.

Finalmente, al final de este documento se hallan las referencias empleadas y los distintos anexos donde se encuentran los distintos planos, tablas, imágenes y descripciones auxiliares, así como los códigos empleados para este proyecto.

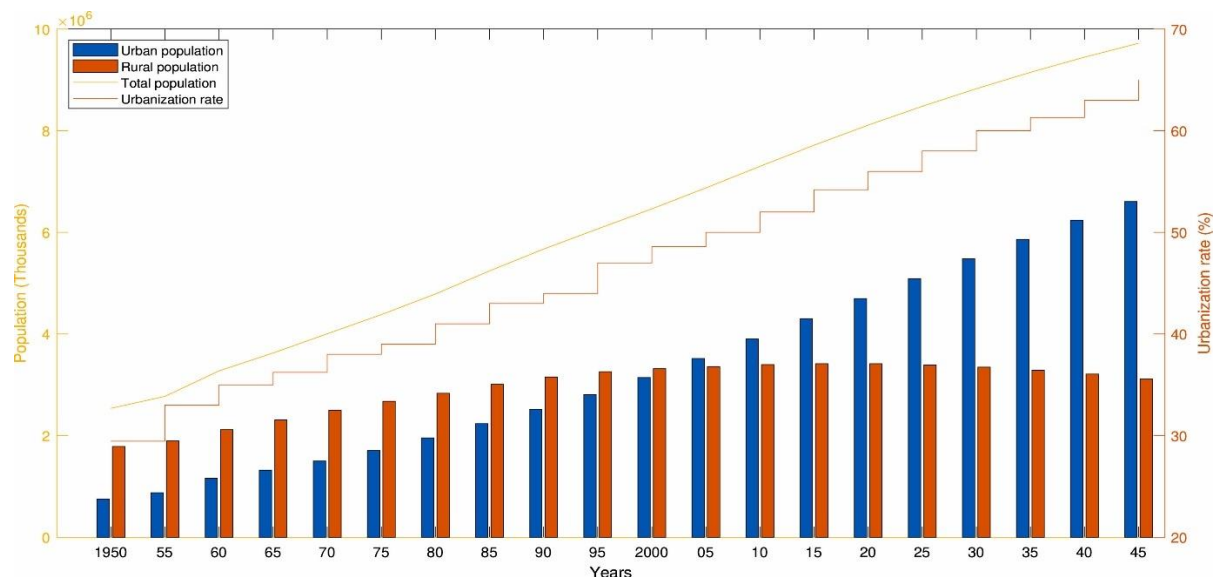
2. Contexto y estado del arte

En este capítulo introduce la temática envolvente del proyecto. La Sección 2.1 describe el contexto y la situación de las ciudades, la definición del concepto de ciudad, llamado Smart Cities y esta se orienta hacia el transporte y la movilidad junto a la problemática del proyecto. A continuación, en la Sección 2.2 se abordan trabajos relacionados, describiendo los procesos novedosos empleados para el sistema y de esta forma dar paso a las tecnologías empleadas y sus características intrínsecas. Finalmente, en la Sección 2.3 se describe, de manera argumental, las conclusiones del capítulo.

2.1. Smart Cities: contexto e implicaciones actuales

Las ciudades han experimentado un crecimiento sin igual de sus habitantes en las últimas décadas, hasta el punto en que el 50% de la población mundial ya reside en ellas. Junto a este hecho, a principios del siglo XXI, empezaron a surgir las “Megaciudades”, ciudades con más de 10 millones de habitantes (Pelton, 2019). La mejora de oportunidades y el crecimiento económico inició un éxodo de la población rural hacia estas y se espera que esta tendencia continúe los próximos años (Khan et al., 2022). En la Figura 2 se describe dicha tendencia y la posible fluctuación futura.

Figura 2. Evolución de la población rural y la residente en ciudades a nivel mundial.



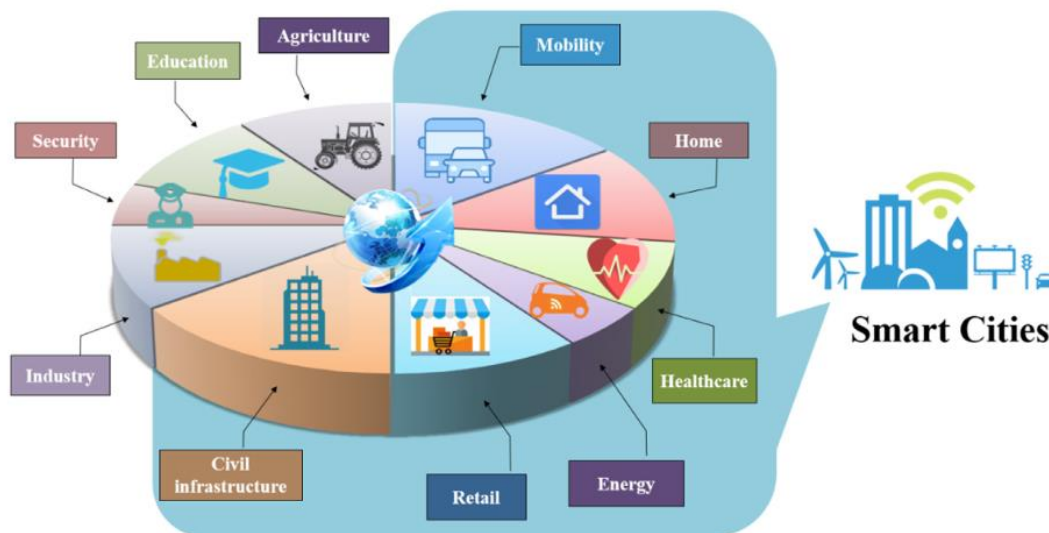
Fuente: Li et al. (2019).

Estos acontecimientos han hecho que en todos los ámbitos como el transporte, la salud y asistencia, en infraestructuras y energía, etcétera, se viesen afectados drásticamente (Alavi et al., 2018).

No es de extrañar que, las ciudades sean un aspecto fundamental e intrínseco en la resolución del cambio climático y que a raíz de esto, junto a otras necesidades, se apremie el desarrollo de nuevas tecnologías (Orejon-Sanchez et al., 2022). El control y uso de tecnologías para las ciudades son consideradas un aspecto crucial en la reducción de emisión de gases de efecto invernadero y redundantes en la mejora de la sostenibilidad energética, siempre que sean dirigidas por estrategias políticas conscientes (Allam & Dhunny, 2019).

Las ciudades más modernas intuyen el fin de un modelo usado y se ven forzadas a modernizar, tanto sus sistemas como metodologías y usar gran variedad de tecnologías y dispositivos electrónicos como parte intrínseca de la solución (Piracha & Chaudhary, 2022). Esto obliga a la diversificación y virar el enfoque hacia nuevos conceptos y áreas de desarrollo dando a conocer el nuevo concepto “Smart Cities” (Figura 3).

Figura 3. Dominio de aplicaciones IoT para Smart Cities.



Fuente: Alavi et al. (2018).

El concepto de Smart Cities no es nuevo, aun así, no ha sido hasta hace poco que la población ha introducido este lenguaje en su día a día. La demanda de una ciudad más respetuosa con el medio ambiente, ágil y dinámica ha crecido exponencialmente, además, en las últimas dos décadas, los artículos relacionados con Smart Cities se multiplicaron por 600 (Duygan et al.,

2022). Este nuevo concepto es ampliamente asociado a numerosos aspectos, sin embargo, contiene un propósito común, la mejora en calidad de vida, los más relevantes se muestran en la Tabla 1.

Algunos de los aspectos relacionados con la mejora de la calidad de vida se vieron más impactados que otros, como la calidad del aire, empeorada por el uso masivo de transportes dentro de la ciudad. Anenberg (2019) informa que la muerte prematura de 3.5 millones de personas en todo el mundo en el año 2017 por diabetes, infecciones pulmonares y respiratorias, enfermedades cardiovasculares y por obstrucción, así como los cánceres asociados son causa directa del aire contaminado por el transporte de movilidad.

Tabla 1. Palabras clave asociadas a 166 definiciones de Smart Cities (2016).

Palabra clave	Repeticiones	Porcentaje de apariciones en definiciones (%)
Calidad de vida	44	37.9
Tecnología	42	36.2
Gente	36	31.0
Gobierno y administración	35	30.2
Sistemas	34	29.3
Economía	34	29.3
Sostenibilidad	28	24.1
ICT (Infraestructura Común de Telecomunicaciones)	27	23.3
Recursos	27	23.3
Salud, seguridad y protección	26	22.4

Fuente: UNE 178201 Ciudades inteligentes. s. f..

En esta línea argumental, uno de los dominios que empieza a tomar especial relevancia pública es el transporte y la movilidad, representando uno de los mayores retos ya sea por su gran implicación (directa o indirecta) en la economía, la sostenibilidad, en salud, entre otras. Este dominio tomó especial transcendencia con los estudios realizados durante y después de la pandemia del COVID-19.

El COVID-19 otorgó a la población una mayor percepción de la calidad del aire y sus beneficios exigiendo una mayor demanda de espacios verdes y legislación, del año 2020, por parte de las entidades gubernamentales (Rousseau & Deschacht, 2020). Algunas legislaciones ya conocidas como las zonas de bajas emisiones (ZBE) empezaban ya a ser frecuentes y, ciertamente estas ayudan a reducir las más de muertes prematuras, estimadas en 40 000 en la Área Económica Europea (EEA) para el año 2017 (*Environmental Implementation Review*, s. f.).

Un ejemplo claro de ZBE se puede encontrar en el centro de Madrid la cual, en el año 2021, permitió aumentar en un 66% el tiempo en que los gases (especialmente el dióxido de nitrógeno, NO₂) permanecían por debajo de los niveles marcados por la Unión Europea (Lebrusán & Toutouh, 2021). De la misma forma, su impacto fue notado en todos aquellos componentes más habituales en el cóctel de gases perjudiciales para la salud producidos por el transporte. En el aire de las ciudades se pueden encontrar elementos como el dióxido de azufre (SO₂), el dióxido de carbono (CO₂) y el monóxido de carbono (CO), ozono (O₃) y dióxido de nitrógeno (NO₂) (Miller & Newby, 2020) así como partículas en suspensión de varios tamaños (PM).

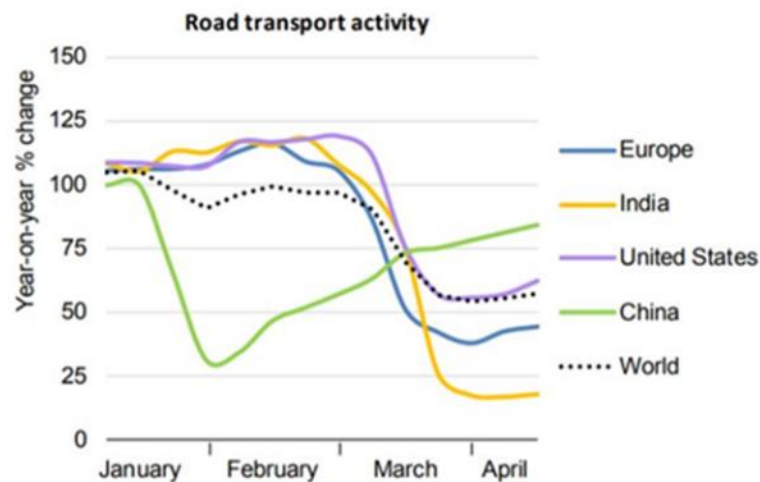
La contaminación no es el único aspecto perjudicial, Sugiyama (2020) señala que el uso continuo y constante del transporte (en especial atención a los coches) junto a una vida sedentaria agravada por el sobrepeso, propicia el aumento del riesgo de enfermedades cardiovasculares. Añadir la importancia en la gestión de las congestiones para reducir el tiempo que las personas se pasan dentro de los vehículos.

Un ejemplo de contexto es la ciudad de Los Ángeles, la cual fue coronada durante seis años consecutivos por ser la ciudad con más tráfico del mundo por el estudio INRIX Global Traffic Scorecard (Cookson & Pishue, 2017). En el 2017 los conductores de Los Ángeles se pasaban un promedio de 102 horas en el coche dentro de las congestiones de la ciudad, además,

suponiendo un gasto promedio de 2828 \$ por conductor y unos 19 000 millones de dólares anuales para la ciudad (Eiza et al., 2020).

Después del COVID-19, el cual provocó que el transporte por carretera sufriera una disminución promedia del 50% entre el año 2019 y marzo de 2020 (Europa Publications, 2020), expuesto en la Figura 4, los países más avanzados tomaron este evento para aplicar nuevos métodos y paradigmas de la movilidad y transporte.

Figura 4. Reducción del transporte por carretera durante el COVID-19 en distintos países.



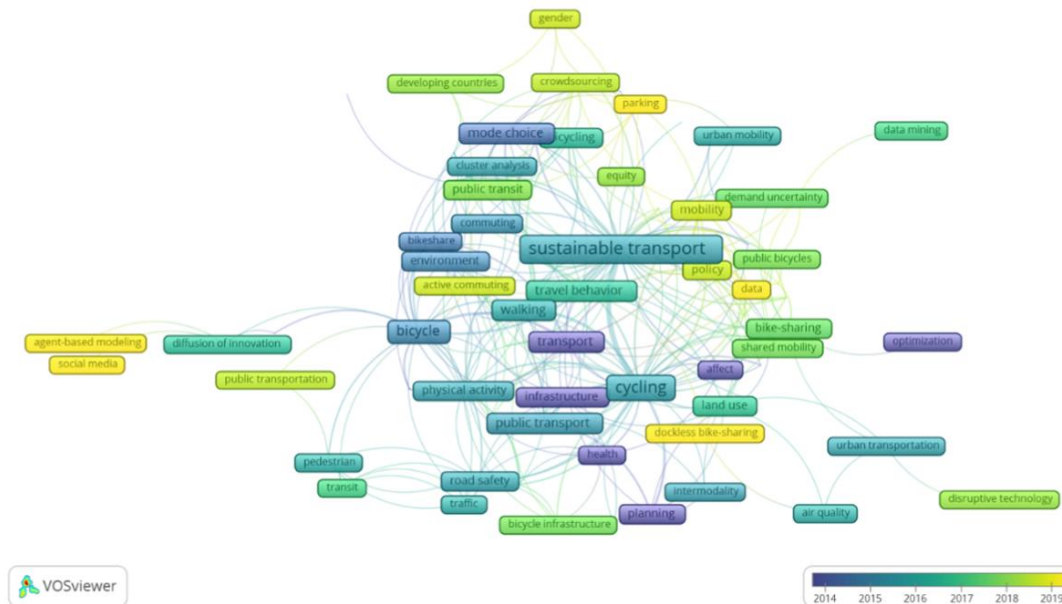
Fuente: Abu-Rayash & Dincer (2020).

Ciudades de todo el mundo promulgaron diferentes medidas, algunas de corto recorrido y otras con más dilatación en el tiempo, todas ellas en entorno al concepto de transporte sostenible (Figura 5). Thombre & Agarwal (2021) daba muestras de cambios de modelo de transportes, tales como el crecimiento del uso de la bicicleta, pasando de un 31% al 44% o uno más reducido en el uso de coches y motos en las ciudades después del COVID-19. Los mismos autores añaden que el transporte público, por otra parte, despertó gran interés durante los confinamientos, pero demostró no ser tan eficiente y reveló ciertas precariedades, hecho que hizo menguar drásticamente el crecimiento de uso esperado.

Un elemento fundamental es requerir un control sobre el número total de vehículos que son usados en las ciudades, ya sean de ámbito privado o público. Especialmente, cuando gran cantidad de estos requieren el acceso a zonas muy confluídas, creando la necesidad del uso de Sistemas Transporte Inteligente (ITS) (Alavi et al., 2018), Sistemas de Parking Inteligentes

(SPS) y Técnicas de Detección de Vehículos (VDT) (Díaz Ogás et al., 2020). Todos estos sustentados por Tecnologías de la Información y Comunicaciones (TIC).

Figura 5. Relación entre palabras clave en movilidad y transporte.



Fuente: Abduljabbar et al. (2021).

Así pues, dentro del paradigma de las Smart Cities un aspecto confluyente entre todos los citados hasta ahora son los aparcamientos, de cualquier tipología. Una de las primeras razones por las cuales implementar sistemas IoT en los aparcamientos, desarrollando así sistemas de aparcamiento inteligentes (SPS, *Smart Parking Systems*), es reducir el tiempo de búsqueda de estos. De esta manera se ayuda a reducir los problemas de salud (menos contaminación, problemas pulmonares, etc.), así como los riesgos asociados (cardiovasculares, etc.) y los costes económicos, ya sean de las personas, empresas y la ciudad (Sugiyama et al., 2020).

A medida que la sensorización de las ciudades va siendo una realidad, los aparcamientos pueden generar una información clave para la gestión del tráfico y ser usada como solución para ciertas congestiones. Esta información que se genera puede ser ingerida y almacenada por sistemas en tiempo real (Plataformas IoT) para luego ser tratada en modelos de inteligencia artificial (IA), como en las redes neuronales (RNN). El uso de estas permiten gestionar a los sensores, actuadores y comunicaciones de manera más eficiente, aumentando el ahorro de su energía hasta en un 50% (Bhattacharya et al., 2022).

Aun así, las tecnologías IoT son muchas y diversas, programas que permiten la simulación en comunicaciones IoT (algunas específicas, como LPWAN) añaden un valor importante a esta selección y resultan indispensables (Idris et al., 2022). En este sentido, el mismo autor argumenta que estos simuladores, en general, permiten visualizar resultados de implementación como, consumo de energía, pérdida de datos o cobertura, cuáles deberían ser las calibraciones necesarias, tanto para dispositivos como comunicaciones y demás.

En esta línea, la ciudad de Girona quiere empezar a realizar las implementaciones necesarias para convertirse en una Smart City. Entre otros proyectos, el equipo de la UMAT (Unidad Municipal de Análisis Territorial) del ayuntamiento de Girona ha empezado a distribuir una red de comunicaciones IoT inalámbricas (LoRa), siendo estas de pruebas y situándolas en lugares estratégicos de la ciudad. De esta forma desean dar cobertura a la zona céntrica de la ciudad.

El proyecto central en el cual pivota este proyecto es el aparcamiento tarifado (tiempo limitado) contiguo de plaza Catalunya, céntrica y perteneciente al casco antiguo. Debido a su patrimonio y otras diversas razones el despliegue sensorial, actuadores incluidos, deben tener una fuente de alimentación autónoma y comunicaciones inalámbricas.

Este aparcamiento se encuentra tarifado por un tiempo limitado y el Ayuntamiento quiere monitorearlo con la detección de los vehículos y, así, poder enviar avisos a los conductores una vez superado el umbral de tiempo usando indicadores luminiscentes para cada una de las plazas. De este modo, quieren fomentar una mayor fluidez de vehículos en la zona durante el día y concienciar a los conductores para realizar sus labores con prontitud.

2.2. Proyectos enlazados y relevantes

En esta sección se encuentran descritos los aspectos más relevantes de proyectos llevados a cabo con aspectos similares. En la Subsección 2.2.1 se encuentran aquellos con tecnologías diversas. A continuación, en la Subsección 2.2.2 se describen proyectos que usan tecnologías relativas al uso de simulaciones, pero en entornos diferentes. La Subsección 2.2.3 finaliza explicando trabajos con tecnologías y escenarios similares a este proyecto.

2.2.1. Implementaciones dispares en el ámbito de aparcamientos inteligentes

En proyectos de aparcamientos para Smart Cities es muy habitual encontrarse con distintos métodos y tecnologías para tratar de solucionar diversas problemáticas del mismo vertical.

Un ejemplo es Amiri (2019), en su trabajo presenta una solución de implementación directa. Esta consiste en la creación de un sistema de privacidad de los datos para una aplicación de reservas de aparcamiento usando Blockchain. El sistema pretende separar la información proveída por los sensores de detección del vehículo, cifrando y anonimizando sus datos.

La ciudad de Nashville (Estados Unidos) es planteada para esta solución, para ello implementan la arquitectura DLT (*Distributed Ledger Technology*) o Tecnología de Registro Distribuida usando Blockchain para permitir separar la identidad del usuario de la localización del aparcamiento solicitado. Como resultado, obtienen una implementación capaz de asegurar, de manera transparente y con alta disponibilidad, la anonimización de datos a un precio relativamente bajo, aun así, no implementan la conexión con la aplicación de reservas.

La casuística de los aparcamientos es mucha y diversa, en el siguiente ejemplo se muestra una problemática relacionada con los modelos de aprendizaje automático (ML, *Machine Learning*).

Awan (2020) plantea una solución para determinar el mejor algoritmo de ML o DL (*Deep Learning*) para su uso específico en predicciones de disponibilidad de plazas de aparcamiento. Este proyecto se basa en los datos recogidos de fuentes públicas, en concreto de los aparcamientos de la ciudad de Santander (España), generados dentro del proyecto H2020 WISE-IoT (*Worldwide Interoperability for SEmantics IoT | Wise-IoT Project | Fact Sheet | H2020 | CORDIS | European Commission, 2022*).

En este trabajo se realiza una comparativa entre distintas metodologías usando el mismo set de datos, como *K-Nearest Neighbors*, árboles de decisiones (*Decision Tree*), perceptrón multicapa, entre otros. La finalidad es encontrar aquel método con mayor predicción. Métricas como la exactitud, la precisión o el *recall* (Ratio de verdaderos positivos), y demás son estudiados y considerados exhaustivamente.

En los análisis resultantes, Awan (2020) observa que los algoritmos más sencillos, como el árbol de decisiones, ofrecen una mayor predicción y desbancan aquellos más complejos, como el perceptrón multicapa. Además, observa la peculiaridad de, llegado a cierto punto del tamaño del set de datos, este deja de ser relevante para el entrenamiento (*overfitting*).

Otros aspectos relevantes para el *Smart Parking* es poder encontrar aquel espacio (reservado o indicado) rápidamente, sin tener que estar buscándolo mucho tiempo. Por ello, Jemmali

(2022) presenta un trabajo para tratar de encontrar el mejor algoritmo con el cual, distribuir los vehículos de manera eficiente y con prioridad según espacio, tiempo de llegada, ocupantes, entre otros.

Diferentes algoritmos son presentados, como el algoritmo aleatorio de repetición múltiple, del inglés *Multi-repeating randomized algorithm* (MR) o algoritmo aleatorio iterativo llamado, *Iterative randomized-NI algorithm* (NI). Luego, desarrolla un sistema de distribución individual para las plazas de aparcamiento y realiza su implementación usando datos públicos o sacados de literatura (medidas estándares para plazas de aparcamiento, nombre medio de personas por vehículo, etcétera), añadir que, algunos datos son concebidos para escenarios concretos.

Finalmente, examina los resultados y los compara. El algoritmo con mayor fiabilidad es el MR con un 96% junto al menor tiempo de respuesta, de esta forma en el caso de implementación real el MR sería el candidato escogido. En futuros trabajos propone la mezcla de algoritmos y la posible implementación en sistemas de inteligencia artificial.

2.2.2. Modelos de simulación para escenarios dispares

Las simulaciones no solo son usadas e implementadas para este marco o ambiente, hoy día casi cualquier empresa o administración usa las simulaciones para evaluar costes, riesgos, determinar barreras (límites), etcétera, para distintas casuísticas (de Paula Ferreira et al., 2020).

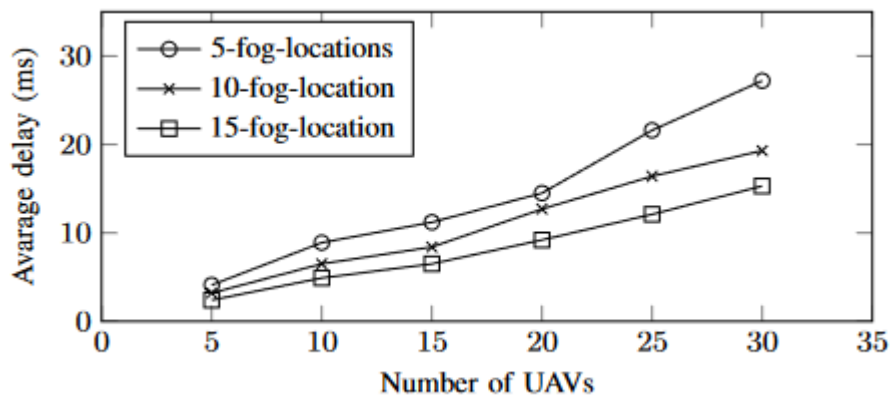
Malik (2020) muestra una simulación de servidores en la niebla (*fog*) para una aplicación del sector agroalimentario, esta, emplea distintos protocolos de comunicación, sensores, actuadores y demás, para englobar la totalidad del sistema. El planteamiento es poder determinar eficazmente un despliegue sensorial sobre el terreno agrario considerando los requisitos del servidor.

La simulación usa distintos algoritmos para contemplar y evaluar diferentes aspectos. Estos son, el uso de recursos, el número de sensores y UAV (*Unmanned Aerial Vehicle*) o Vehículo Aéreo no Tripulado (peculiares para esta implementación) así como la energía empleada por estos durante las transmisiones y, para terminar, la demora de procesamiento de los eventos del servidor.

Finalmente, los resultados son divididos en dos partes, primero muestran la falta herramientas con mayor configuración, no pudiéndose adaptar mejor a la problemática y

formando una guía de posibles mejoras para simulaciones futuras. En segundo lugar, dan paso al producto de la simulación, en este presentan los datos, algunos de ellos representados en la Figura 6, preservándolos para una futura implementación de ML o Aprendizaje por Refuerzo. De la misma manera delimitan los aspectos cruciales del sistema, pudiendo determinar cuáles son los límites del sistema, que número de recursos son necesarios, así como el consumo de los sensores y UAV para determinar su vida útil.

Figura 6. Demora media según número de UAV y servidores FOG.



Fuente: Malik et al. (2020).

Las simulaciones también pueden ser empleadas para la fabricación de nuevos productos en líneas de producción de fábricas, un ejemplo es el siguiente.

Ruane (2023) emplea una simulación para caracterizar un nuevo producto para la farmacéutica Johnson & Johnson. En este caso, la problemática a resolver consta de tres factores decisivos, prever la capacidad dinámica de la línea de producción, optimizar los tiempos de entrega del producto y, finalmente, poder proveer a la simulación de nuevos datos para la mejora continua.

Para llevar a cabo tal simulación, Ruane (2023) determina primero el modelo, en segundo lugar, verifica la información y termina, validando el modelo establecido, comparándolo con información real de la misma compañía en aquellos procesos similares o iguales.

Este trabajo constata dos hechos importantes y destacados por el uso de simulaciones. Primero, es posible realizar simulaciones en entornos de producción dinámicos y, segundo, que los modelos y algoritmos empleados puedan ser contrastados con datos reales, para luego ser implementados en programas especializados para el sector productivo.

Al encontrarnos en entornos IoT, no solo importan las plataformas, dispositivos o protocolos, sino también el conocimiento extraído de la cantidad ingente de información generada, también llamada minería de datos o *Data Mining*. Savaglio & Fortino (2021) pretenden, en su labor, encontrar la manera de recoger el máximo de requisitos necesarios para una implementación de minería de datos antes que esta se llegue a implementar, teniendo en cuenta el paradigma de computación Edge y Cloud.

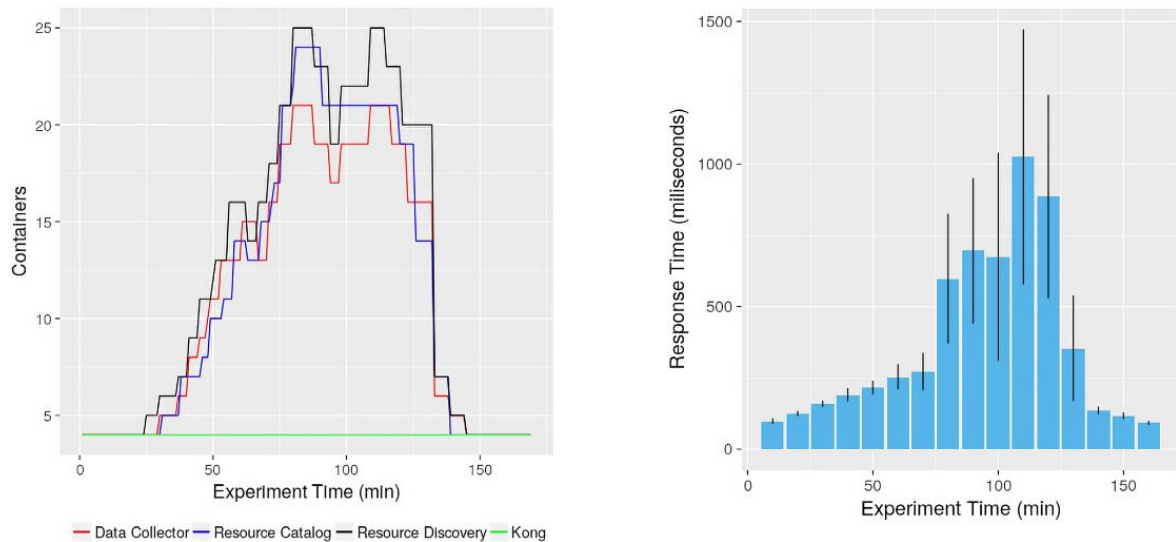
Por consiguiente, primero analiza el escenario de implementación IoT, analiza los requerimientos imprescindibles para el sistema de minería y contempla aspectos de computación en el borde y en la nube. En segundo lugar, implementa la herramienta de simulación EdgeMiningSim con los aspectos de la primera fase. Finalmente, aplica esta metodología en un escenario de medioambiente.

La solución presenta distintos datos, algunos de los más llamativos son los tiempos de respuesta a peticiones de información, así como, de computación a lo largo de diferentes iteraciones para dar una muestra comparativa entre los sistemas desplegados en la nube o en dos tipologías de borde. Entre estos resultados y demás, se concluye que el programa utilizado (EdgeMiningSim) resulta determinar un papel clave, novedoso e intuitivo para el despliegue de soluciones de este estilo detallando los recursos y características del sistema.

2.2.3. Modelos de simulación en el ámbito de aparcamientos inteligentes

La verticalidad del escenario y la implementación de las simulaciones permiten explorar una gran variedad de problemáticas. En un primer proyecto, Del Esposte (2019) analiza una simulación de escalabilidad horizontal de la plataforma IoT (Middleware) "InterSCity" usada para una aplicación de aparcamiento en São Paulo.

En este escenario se da solución a una supuesta implementación de la plataforma IoT en la nube pública de Google y como respondería esta según diferentes escenarios. Así pues, este proyecto usa la simulación de la plataforma con datos de fuentes públicas (como OpenStreetMap, entre otros), para generar una cantidad de eventos para la plataforma y de esta forma capturar los datos generados. Con estos se permiten delimitar los límites del sistema, gastos asociados a la nube, etcétera. Algunos de los datos obtenidos se encuentran en la Figura 7.

Figura 7. Escalabilidad y tiempo de respuesta del proyecto de simulación para São Paulo.

Fuente: Del Esposte et al. (2019).

En otro proyecto, se implementa una solución de simulación de aparcamiento para Monterrey (México) previa a su implementación. Este proyecto realiza una simulación de las comunicaciones inalámbricas de los sensores de aparcamiento para determinar su efectividad en el caso práctico. Determina diferentes aspectos clave de la zona de implementación, como la cobertura, potencia de transmisión y demás (Celaya-Echarri et al., 2020).

Finalmente, compara los datos con la implementación real y determina que el sistema ha sido contrastado, demostrando el uso de herramientas de simulación como uno de los pasos necesarios para la determinación de sistemas IoT.

En otro trabajo, empleando una casuística de sensores, Bock (2019) realiza una simulación para contrastar un sistema de aparcamientos variables (*Parking Crowd-Sensing*). Este proyecto se desarrolla en la ciudad de San Francisco (Estados Unidos), el objetivo es realizar esta simulación en flotas de taxis de la ciudad y prever si los sistemas actuales de control son capaces de alcanzar una cierta fiabilidad y gestión.

De la misma forma que en proyectos anteriores, este también recoge datos de fuentes públicas para determinar el tráfico de ciertos segmentos de la ciudad y la trayectoria registrada de la flota de taxis, añadiendo distintas casuísticas como, fallos de detección, de guía o modelos estadísticos y demás.

Los resultados de esta simulación demuestran que tan importante es la fiabilidad del sensor, como la cantidad de vehículos a gestionar. Entre otros, se argumenta que los filtros de Kalman (Grewal et al., 2020), usados habitualmente en aplicaciones de GPS, no representan una mejora estadística significativa para este tipo de soluciones.

Para terminar, Pasolini (2018) presenta los resultados obtenidos de una simulación para un escenario de iluminación urbana inteligente de los aparcamientos. Este trabajo utiliza datos reales del entorno, como calles, distribución, entre otros y, emplea las comunicaciones LoRa para simular este escenario con la intención de encontrar la mejor tipología de red LoRa para un despliegue futuro que abarca la mayor cobertura posible.

El proyecto se basa en una instalación funcional de comunicaciones LoRa de la ciudad de Bolonia y abarca ciertos distritos como Zaragoza o Navile. Con esta premisa de antemano, Pasolini (2018) usa el simulador ns-3 y se enfoca principalmente en la capa MAC de LoRa con pequeñas insinuaciones en la capa física.

Los resultados presentados muestran una gran decaída en la cobertura de este tipo de comunicaciones, de unos 15 km estipulados por el fabricante hasta los 1 – 2 km, debido en gran medida por el entorno urbano. Terminando con la clara indicación de la necesidad del uso de múltiples pasarelas (Gateway) para mejorar la cobertura.

2.3. Conclusiones

La casuística de las ciudades es cada vez más compleja y requiere de acciones más concisas, estas se ven afectadas por elementos externos como el crecimiento poblacional debido a movimientos migratorios rurales, el cambio climático, apremiando la escasez de recursos y generando climas más extremos, entre otros. A raíz de estos acontecimientos surge el nuevo concepto de Smart Cities el cual, trata de modernizar los servicios y las metodologías de gestión actuales. Estas engloban varios aspectos, del que cabe destacar la movilidad la cual, es percibida desde muchos ámbitos, pero contiene una problemática general, la salud.

Es por ello, que este proyecto quiere abordar el sistema de aparcamiento tarifado de Girona, en su modernización para mejorar la circulación de los coches y por ende a una menor contaminación.

El uso de las herramientas de simulación para evaluar el impacto de las comunicaciones y sensores empleados en el proyecto antes de que se llegue a implementar genera un valor

añadido para la solución final. Hasta donde existe conocimiento del autor, aun existiendo la implementación de herramientas de simulación para casos de evaluación de sistemas IoT, el uso de estas empleando datos e información real, de los sensores y las comunicaciones candidatas, así como del escenario, con el fin de determinar la efectividad y cumplimiento de los requerimientos no ha sido llevado a cabo en ningún aparcamiento visto hasta la fecha.

3. Descripción general de la contribución

Este capítulo atiende los objetivos a alcanzar y las fases por las cuales se desarrolla este trabajo, es por ello se reparte en dos secciones. La primera Sección 3.1 desglosa el objetivo principal de este trabajo y aquellos puntos primordiales para hacer su alcance posible, mientras que en la Sección 3.2 se enumeran y describen las fases que atiende este trabajo, exponiendo las razones de elección de herramientas cuando es necesario.

3.1. Objetivos

La realización de este trabajo de final de máster contempla la problemática para la selección del sensor y poder encontrar la configuración efectiva de las comunicaciones. Para ello, el objetivo principal trata de implementar y evaluar los resultados de la simulación, en la que se emplea el simulador ns-3, para la tecnología LoRa dentro del sistema de aparcamiento tarifado de plaza Catalunya en Girona. Con ello, se permite emular el comportamiento del sensor y sus comunicaciones, se aplica un modelo de simulación basado en el entorno urbanístico junto a las características intrínsecas de las comunicaciones y del sensor seleccionado.

Para conseguir este objetivo general se desglosan los siguientes objetivos específicos:

- Estudiar la problemática del escenario, concretamente, la realización de una catalogación de los sensores, comunicaciones y terreno, desglosando aquellos aspectos más relevantes para el proyecto y caso de uso.
- Seleccionar el sensor candidato, que cumpla con las mejores características de uso para el entorno y aplicación, para que este sea empleado como referencia en la simulación.
- Diseñar e implementar el modelo de simulación, buscar un modelo de propagación y pérdida de la señal e integrarlo en el simulador ns-3. Es de vital importancia contemplar los datos y características relevantes para conseguir un entorno simulado con el máximo detalle posible de los nodos (sensores, gateway, entre otros) para obtener resultados más fidedignos.
- Implementar *scripts* de Python para digerir la información y, de esta manera, extraer las métricas de salida del simulador ns-3 para la ingesta del programa de visualización. Los datos obtenidos por el simulador ns-3 son varios y no se encuentran normalizados. La programación con Python ayuda a generar prefiltros y seleccionar los datos de los

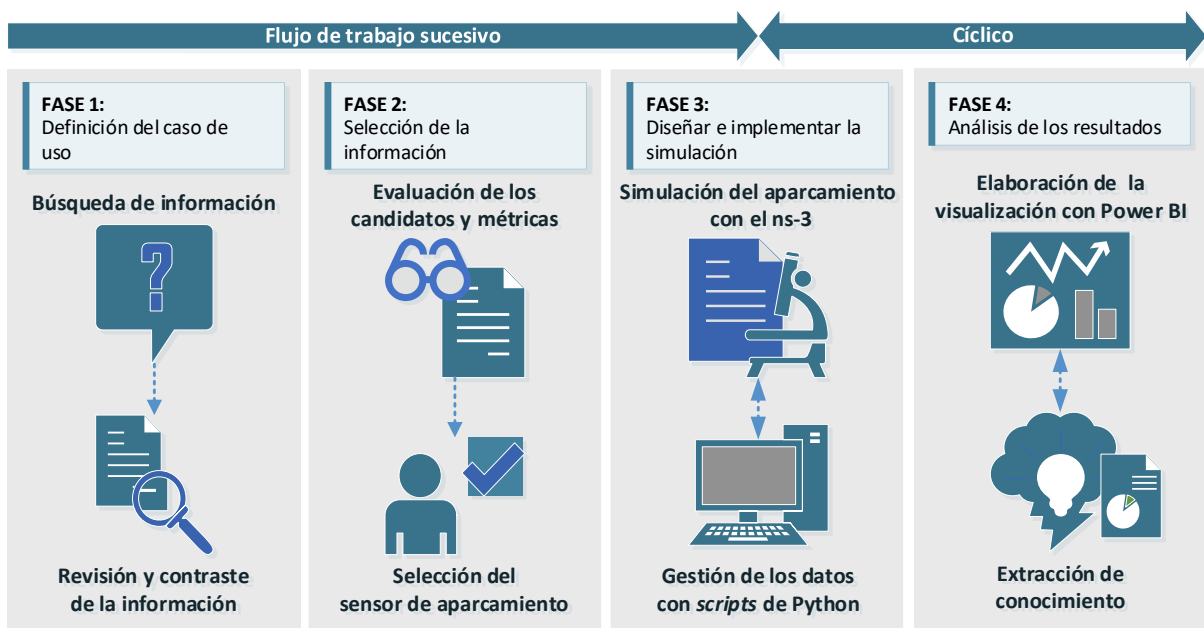
archivos de registro (LOG) generados por el simulador, de este modo, transformarlos en un formato relacional.

- Modelar e implementar la visualización de los datos, a la vez llevar a cabo un proceso de visualización para poder seleccionar, filtrar y digerirlos. Finalizando con la creación de distintos modelos visualización dashboard para su posterior análisis, facilitando su percepción.
- Analizar los resultados finales de la simulación y aflorar detalles relevantes para la implementación y configuración de los sensores y comunicaciones afines al consumo y cobertura, entre otros.

3.2. Metodología del trabajo

La ejecución de este estudio se realiza siguiendo las diferentes etapas descritas en las siguientes subsecciones. Aun así, estas fases se encuentran resumidas en la Figura 8, se realizan por orden enumerativo y argumentadas en secciones posteriores.

Figura 8. Guía de las distintas fases para este proyecto de simulación de aparcamiento.



Fuente: Elaboración propia.

3.2.1. Fase 1: Definición del caso de uso

Esta primera fase destaca por recoger y detallar toda aquella información relevante del caso de uso como su entorno (ciudad, terreno...), los sensores y comunicaciones empleadas. Las

aplicaciones de aparcamientos inteligentes para Smart Cities pueden ser muchas y diversas, como se ha podido observar en el estado del arte de este documento en la Sección 2.2.

En este caso, al tratarse de un despliegue para gestionar un aparcamiento tarifado por tiempo limitado, se describen los requisitos del sistema, como el área de despliegue sensorial, las especificaciones técnicas, la tipología de los sensores y sus comunicaciones, entre otros.

Además, se estipulan los aspectos imprescindibles que el ayuntamiento de Girona quiere implementar para el despliegue. Por esta razón se generan tablas de Objetivos y Resultados Clave u OKR (*Objective and Key Results*) junto a sus Indicadores Clave de Rendimiento o KPI (*Key Performance Indicator*).

La recogida de toda la información permite generar un marco de referencia para el proyecto y se definen claramente los elementos de la evaluación. A partir de este punto no se añade ninguna información que no sea imprescindible y, por consiguiente, se da paso a la segunda fase.

3.2.2. Fase 2: Selección de la información necesaria

Una vez se ha planteado el escenario, el siguiente paso es seleccionar y relacionar la información obtenida con los modelos de simulación que se quieren llevar a cabo, siguiendo el criterio base, establecido en la Fase 1 Subsección 3.2.1. Algunas métricas empleadas y de vital importancia en las comunicaciones LoRa y sensoriales son brevemente descritas en la Tabla 2.

La selección del sensor se realiza en esta fase, los criterios de evaluación se enfocan en tres pilares, la detección, las comunicaciones y el consumo. Las métricas sensoriales para la elección del mejor candidato son evaluadas desde un punto de vista comparativo donde se exponen, tiempo y rango de detección del vehículo, capacidad de la batería, protección y modelo de instalación (superficie, enterrado, u otros). Además, todas aquellas que tengan relación con el módulo de comunicaciones LoRa, configuración de la modulación de la señal como el modelo de chip, antena, etcétera.

En el caso de las comunicaciones LoRa se da vital importancia a las capas físicas (PHY) y MAC, referente a LoRaWAN. Aunque, en el caso de LoRaWAN, esta se aleja de la capa de Aplicación para centrarse en aquellos aspectos de unión y referencia hacia la capa física PHY perteneciente de LoRa. Con este enfoque se deja de lado la plataforma “The Things Network”,

la gestión y administración de los dispositivos, velocidades de transmisión entre servidores (“Network Server”, “Gateway Server” y demás), entre otros.

Tabla 2. Métricas destacables para la generación de conocimiento.

Nombre	Unidades	Acrónimo	Descripción
Potencia de transmisión, “Power Transmission”	dBm	Tx / TP	Energía empleada para la modulación de la señal
Potencia recibida, “Receive Power ”	dBm	Rx	Energía de la señal en el receptor, pasarela u otros
Relación entre señal y ruido, “Signal to noise ratio”	dB	SNR	Proporción entre la potencia de la señal transmitida y la potencia del ruido.
Ancho de banda, “Bandwith”	Hz	BW	Diferencia entre la máxima y mínima frecuencia de modulación de la señal
Factor de dispersión, “Spreading Factor”	N/A	SF	Controlador de la velocidad de transmisión de la señal de LoRa (The Things Network)
Capacidad de la batería	A·h/mA·h	A·h/mA·h	Cantidad de la carga eléctrica del sensor
Voltaje de alimentación	V	Vcc	Alimentación de corriente continua del sensor
Corriente en modo “Deep Sleep”	mA	DP	Corriente mínima del sensor en modo ahorro de energía

Fuente: Elaboración propia.

3.2.3. Fase 3: Diseñar e implementar la simulación

Para esta fase se ha seleccionado el simulador ns-3 el cual, se define como un simulador de redes de eventos discretos. Idris (2022) aglomera los tres entornos de simulación más usados para LoRa: LoRaSim, el OMNeT++ (FLoRa) y el ns-3. Estos quedan reflejados en la Tabla 3.

Tabla 3. Comparativa de simuladores para este trabajo.

Características	ns-3	OMNet (FLoRa)	LoRaSim
Lenguaje	C++ y Python	C++	Python
Eventos	Discretos	Discretos	Discretos
Licencia	Abierta	Abierta	Abierta
Modelo de energía	Sí	Sí	Sí
Soporte ADR LoRa ("Adaptative Data Rate")	Sí	Sí	No
Soporte ACK ("Acknowledge")	Sí	Sí	No
Confirmación "Uplink"	Sí	No	Sí
Tráfico "Downlink"	Sí	Sí	No
Modelos de propagación urbanos	Sí	Sí	Sí
Soporte de la comunidad	Muy buena	Limitada	Limitada

Fuente: Extracto de Idris et al (2022).

El ns-3 permite implementar modelos más sofisticados, con mayor versatilidad, y existe más información al respecto, aunque también es necesario argumentar que tiene una curva de aprendizaje de mayor dificultad y requiere de conocimientos más profundos de las tecnologías

a implementar. Esto alarga el periodo de implementación y modelaje, pero, aun con esto, los resultados son más fiables y representativos de la realidad.

Con el simulador de ns-3 se diseñan las simulaciones requeridas usando las métricas de despliegue del aparcamiento como, el número de sensores, la cantidad de datos que envían ("Payload", según sensor), las pasarelas de LoRa (gateway) y los edificios colindantes.

Los escenarios principales de la simulación para este proyecto son tres. En un primer escenario, de punto de partida, se encara el diseño de la simulación para ver el efecto de los edificios colindantes junto al efecto del SF (*Spread Factor*) sobre la potencia recibida y la relación señal y ruido o SNR. Partiendo con el uso de todos los sensores estratégicamente colocados en el área de despliegue junto a la gateway en la torre Alfons XII, se configura el SF más bajo y se anulan los algoritmos ADR, *Adaptative Data Rate* y ACK, *Acknowledge*. A continuación, se realizan las simulaciones con un mismo SF y Tx para un repertorio de periodos de detección del sensor, repitiendo esta iteración por todos los SF. Este entorno se simula primero para "Uplinks" sin coche y luego con la atenuación correspondiente a este.

Finalmente, esta prueba se vuelve a realizar una vez analizado sus datos en la Fase 4, pero esta vez se escoge el mejor resultado y solo se itera cambiando, esta vez sí, el poder de transmisión, Tx, con ello, se espera generar hasta 61 conjuntos de datos para este primer escenario.

El segundo escenario parte de la misma premisa que el anterior, pero con el modo ACK activo, por lo que se pretende poder comparar los resultados de ambos escenarios. Además, se plantea la opción con el ADR finalmente activado (ambos, ACK y ADR), para observar cuales son las características elegidas por el sistema (SF y Tx), generando hasta 63 conjuntos de datos para el segundo escenario.

Tercer escenario, se recogen los datos resultantes de los escenarios anteriores y también aquellos que afectan a la configuración de los sensores para diseñar un modelo de consumo de estos. Configuraciones como la potencia de envío (Tx), SF, periodo de transmisión, entre otros y, datos recogidos físicamente del sensor seleccionado como, consumo en estado "DeepSleep" (mA) o voltaje, son integrados en la simulación. Con toda esta información, el modelo de simulación pretende estimar el consumo real del sensor y de esta manera obtener

una predicción de la vida útil de la batería del sensor. Todos los escenarios quedan resumidos en la Tabla 4.

Tabla 4. Características de los distintos escenarios para la simulación de Plaza Catalunya.

Escenario	Descripción
Primero	<ul style="list-style-type: none"> • Uso de los 33 sensores, la gateway y el servidor junto al entorno • Repetición de la simulación desde el SF 7 hasta el 12, con y sin vehículo • Tx a 14 dBm • ADR y ACK desactivados
Segundo	<ul style="list-style-type: none"> • Mismos dispositivos y entorno • Exacta repetición para los SF con ACK activado. • Tx a 14 dBm • Simulación extra del escenario con ADR y ACK activados, con y sin vehículo
Tercero	<ul style="list-style-type: none"> • Mismos dispositivos y entorno • Activación del algoritmo para el cálculo de la duración de la batería junto a la configuración seleccionada (SF, modalidades ACK o ADR, Tx, entre otros)

Fuente: Elaboración propia.

En paralelo a esta tercera fase, se dispone del sensor escogido durante la Fase 2 y realizan pruebas de consumo reales. El sensor es desarmado y se realizan capturas del consumo total del sensor en diferentes estados como, "Deep Sleep", detección del vehículo y, finalmente, durante la transmisión de datos. La prueba se realiza calculando la caída de voltaje de una Shunt enlazada en la entrada del voltaje de alimentación del dispositivo, "Pull-Up", aprovechando la ley de Ohm, se convierte en corriente (Amperios).

Otra de las pruebas que se realiza es la detección de vehículos, empleando distintas modalidades de coche se recogen los datos de la señal recibida por una gateway LoRa, con la intencionalidad de visualizar las distintas atenuaciones provocadas por el estacionamiento.

Toda esta información es luego utilizada tanto en la Fase 3 como en la 4 durante el análisis, de esta manera tratan de justificar ciertos usos de los modelos de predicción de la señal y da una visión más fidedigna de la realidad.

3.2.4. Fase 4: Análisis de los resultados

En esta última fase se estudian todos aquellos aspectos relevantes del caso de uso. Principalmente, se seleccionan los datos obtenidos, se filtran y digieren, gracias al diseño y programación de *scripts* con Python. A continuación, se emplea la herramienta de visualización de los datos, con previo proceso de visualización. La función principal es evaluar y proveer el conocimiento para la futura implementación de este proyecto.

Los datos recogidos del simulador ns-3 son datos no estructurados, lo que conlleva un trabajo extra antes de realizar su visualización, posterior análisis y extracción de conocimiento. Python permite transformar estos datos y expórtalos en estructuras como, archivos en coma flotante (.csv), archivos de marcado extensible (.xml) y, Excel (.xlsx), entre otros.

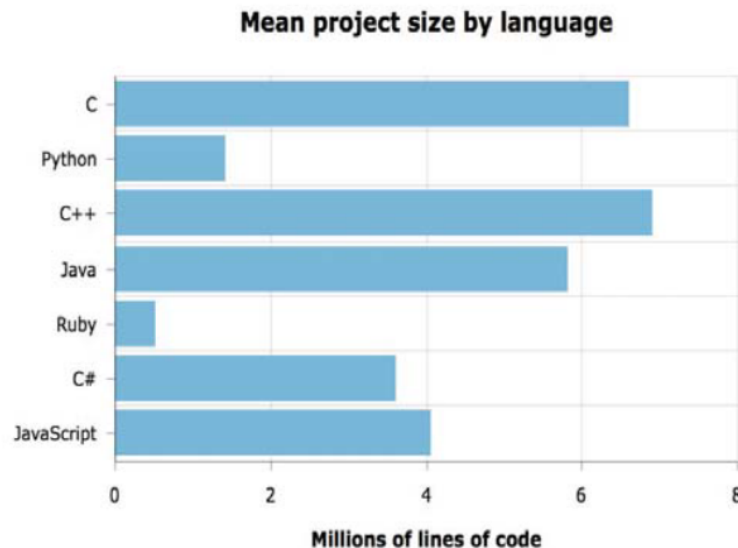
Este lenguaje de programación es muy versátil y presenta muchas ventajas respecto a otros lenguajes de programación. Nagpal & Gabrani (2019) presenta un trabajo donde expone la utilidad de Python para el ámbito de analítica de datos. Una de las primeras y buenas razones para usar este programa es su balance entre medio y alto nivel de abstracción que permite su fácil interpretación e implementación respecto a otros lenguajes, como puede ser observado en la Figura 9. Además, contempla una gran variedad de librerías (Numpy, TensorFlow, Pandas, etcétera) “Open Source”, libre de sistema operativo (lenguaje interpretado, portabilidad) y gran variedad de estructuras de datos.

Por otro lado, no todo son ventajas y bien es cierto que Python contempla algunas desventajas como la velocidad de cómputo, notorio en procesado de grandes cantidades de datos, comparado con C, C++ o Go (Nagpal & Gabrani, 2019). También consume una gran cantidad de memoria, lo que representa un impedimento para realizar ciertas tareas en paralelo si, se quieren procesar grandes cantidades de datos y se utiliza un ordenador de media o baja calidad.

Las herramientas de visualización actuales son abundantes y extensibles, este trabajo contempla un enfoque genérico de los datos, por ello es necesario que la herramienta de visualización contemple un gran abanico de posibilidades. Por otro lado, los *scripts* presentan

los datos necesarios para su visualización, pero no se descarta una posterior transformación durante la visualización, la posibilidad de manejar datos a través de la herramienta de visualización es clave.

Figura 9. Comparativa entre lenguajes de programación y la mediana en líneas de código.



Fuente: Extracto de Nagpal & Gabrani et al (2019).

Para finalizar, hay dos aspectos más para tener en cuenta y que son de gran apreciación, la compatibilidad con elementos Office y licencia gratuita. Debido a la naturalidad de este proyecto es interesante poder importar e implementar los datos de forma nativa en la herramienta, mejorando la interacción con estos y la experiencia del usuario. De la misma forma si la herramienta presenta una versión gratuita con gran variedad de gráficos e implementaciones (filtros, entre otros) se valora muy positivamente.

Con todos estos propósitos se desarrolla la Tabla 5, que enumera algunas de las diferencias más importantes de las herramientas equiparadas desde el punto de vista del cuadrante mágico de Richardson et al. (2020).

La herramienta de visualización escogida es Power BI, destacado por la facilidad de digestión de los datos de archivos de Excel el cual permitirá no dedicar muchos recursos a la programación de los *scripts* de Python, además de facilitar el modelaje final de estos datos. A parte, se observa la posibilidad de complementar cierta carencia de gráficos, junto al Excel o elementos de terceros gracias a la tienda online, con gran variedad y gratuita.

Tabla 5. Comparativa de herramientas de visualización.

Características	Tableau	Google Looker Studio	Microsoft Power BI
Versión sin licencia	Prueba temporal	Sí (Limitada)	Sí (Limitada)
Integración con herramientas Office	No	No	Sí
Tipo de variables	Univariante, Multivariante	Univariante, Multivariante	Univariante, Multivariante
Modificación de datos	Herramienta a parte	Sí	Sí
Modelos mixtos de datos	No	No	Sí
Acceso a modelos de datos de terceros	No	No	Sí
Soporte e información	Muy buena	Buena	Buena

Fuente: Elaboración propia e información extraída de Richardson et al. (2020).

En comparativa con Tableau o Google Looker Studio, Power BI tiene algunas carencias en cuanto a gráficos, aunque, como se ha explicado anteriormente, tiene solución. En otros aspectos, no es la herramienta con la mejor curva de aprendizaje y presenta ciertos retos, aun así, la experiencia previa con herramientas de este estilo facilita su uso. Para finalizar, Power BI tiene una facilidad para la importación de datos que no es recíproca en la exportación de gráficos a archivos Office, al menos con la versión gratuita, pero existen métodos que lo permiten.

Durante esta equiparación también se han tenido en cuenta muchas otras herramientas de visualización ampliamente utilizadas para aplicaciones IoT, como Grafana o Demo, entre otros, las cuales son descartadas debido a su especialización, en series temporales u otros. Esto dificulta drásticamente la elaboración e importación, así como su futuro análisis de los datos de este trabajo.

4. Desarrollo específico de la contribución

En este capítulo se desglosan las fases del proyecto, se demuestra, aclara y define el trabajo realizado durante la mayor parte del tiempo de desarrollo. En este sentido las fases quedan contempladas por secciones individuales, siendo cuatro las fases a realizar.

La primera Sección 4.1 recoge la información relevante e imprescindible para las futuras etapas y define la base sustancial del proyecto. En la Sección 4.2, primero se realiza la evaluación de los sensores candidatos, escogiendo el mejor y a continuación, se estipula aquella información relevante y necesaria para la Fase 3 de la que no. En la tercera Sección 4.3 se formula el escenario, se describen los modelos de comunicación (pérdidas y transmisión, entre otros) y se estipulan las razones y necesidades para la implementación y realización de los distintos escenarios. Finalizando con la Sección 4.4, en la que se realiza la visualización de los datos y la extracción de conocimiento con respecto a los objetivos marcados por el ayuntamiento.

4.1. Fase 1: Definición del caso de uso

Es imprescindible dejar estipulada toda la información necesaria para el correcto desarrollo del proyecto en esta fase, por ello, en esta sección, se dividen diversas subsecciones.

En la primera Subsección 4.1.1 aparece desglosada la información más relevante del proyecto, como el emplazamiento, la posición de los sensores, la pasarela, entre otros. En la Subsección 4.1.2 se recoge el máximo de información pública disponible de los sensores candidatos y la pasarela, se generan tablas con las métricas asociadas. Finalmente, en la última Subsección 4.1.3 se desglosan los OKR y sus respectivos KPI.

4.1.1. Zona de aplicación del proyecto

Como ya se ha mencionado en capítulos anteriores, este despliegue afecta al aparcamiento de plaza Catalunya en la ciudad de Girona donde, en un futuro, se pretende implementar los sensores para esta aplicación.

Para hallar con mayor precisión el emplazamiento se ha desarrollado un plano de situación representado en la Figura 37 de los anexos (Todos los mapas y planos que aparezcan son orientados al Norte geográfico si no se indica lo contrario), remarcando las zonas de despliegue de los sensores, en la plaza y la pasarela (gateway) en la Torre de Alfons XII.

Centrado ahora el foco en el despliegue de los sensores, los aparcamientos pertenecientes a la zona de tiempo limitado son los perimetrales a la plaza, quedando reflejados en las áreas rojizas de la Figura 10, los aparcamientos colindantes no competen. Además, en esta misma figura quedan reflejadas las alturas geodésicas del terreno, entre 68.20 m y 68.40 m (siempre que aparezca una referencia de altitud, esta pertenecerá al *Institut Cartogràfic i Geològic de Catalunya*, s. f., en caso contrario se especificará la fuente alternativa).

Las áreas se encuentran divididas en dos, con distinto repertorio de tipos de aparcamientos, los estándares y los destinados a personas de movilidad reducida, con una totalidad de 33 plazas, repartidas en 29 estándares y 4 de movilidad reducida.

Figura 10. Área de despliegue sensorial en Plaza Catalunya.



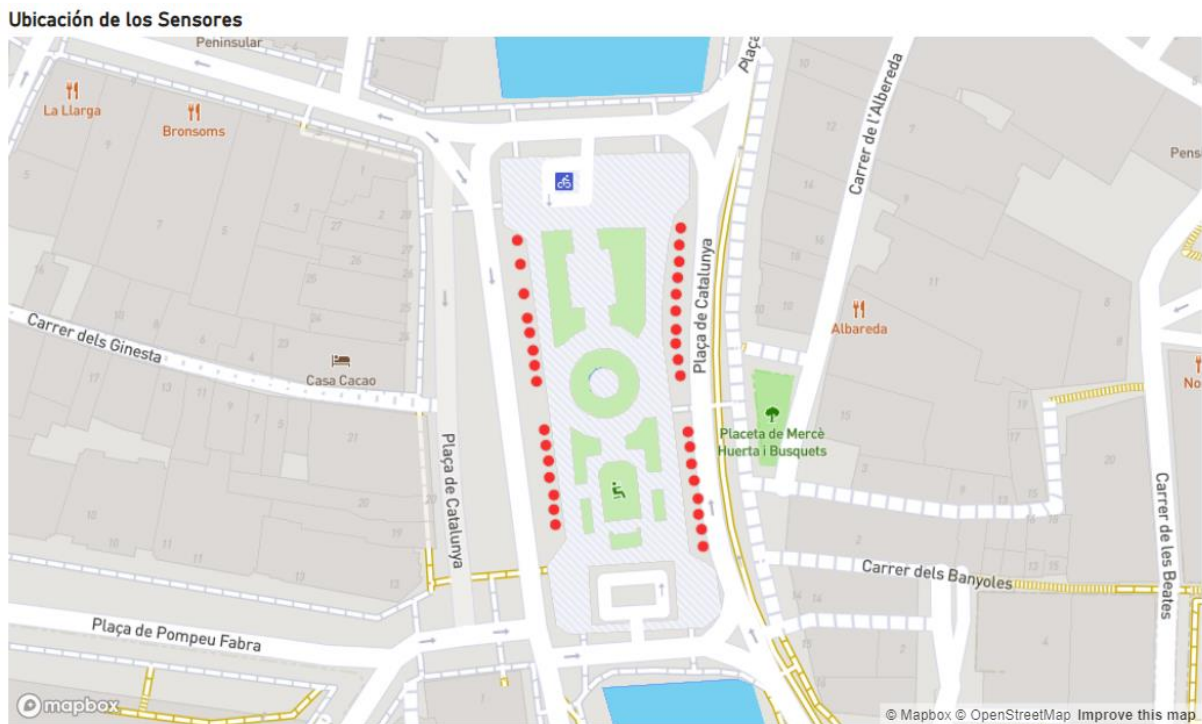
Fuente: Elaboración propia y mapa del *Institut Cartogràfic i Geològic de Catalunya*, s. f..

La distribución de los sensores con respecto a los aparcamientos queda reflejada en la Figura 11 en formato de coordenadas geográficas, latitudinales y longitudinales. Los valores de las coordenadas de los sensores, junto a la gateway, se encuentran recogidos en la Tabla 12 de los anexos.

Resulta interesante recoger información de las dimensiones de los aparcamientos, ya que ayudan a tener una mayor precisión en la posición de los sensores, la cual queda determinada una vez se selecciona el candidato final.

Estas dimensiones se rigen por la ordenanza vigente del ayuntamiento, en este caso la ordenanza de edificación de abril del 2015 (*Ordenances d'edificació, 2015*) establece que las dimensiones mínimas son de 2.30 m de ancho por 4.80 m de largo. Después de una medición *in situ*, se aprecia un ancho de 2.50 m por 6.50 m de largo, sin separación extra entre los de modalidad estándar, a excepción de los reservados para movilidad reducida, siendo de las mismas dimensiones, pero obteniendo un espaciado extra entre ellos de 1.90 metros.

Figura 11. Mapa de la ubicación de los sensores en Plaza Catalunya.

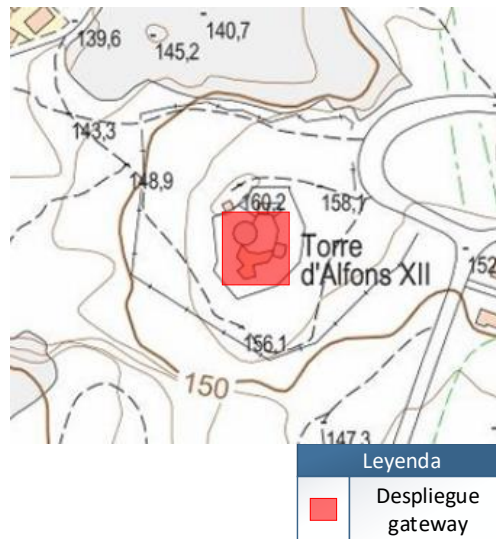


Fuente: Elaboración propia y mapa del *Instituto Geográfico Nacional, s. f.*

Con el despliegue cubierto la gateway es la siguiente en ser mencionada. Esta se encuentra desplegada en la Torre de Alfons XII, uno de los mayores puntos elevados de la ciudad, según la señal geodésica, a una altura de 160.20 metros, reflejado en la Figura 12. La torre en sí tiene una altura total de 18 metros respecto al suelo, la pasarela se encuentra instalada en un mástil de 2.50 metros de largo, en su punto más álgido.

Los datos de elevación del terreno indican una diferencia sustancial entre dichas zonas, para ello resulta interesante registrar tal información. Para ello se debe trazar un perfil longitudinal que recoja las diferentes alturas. Con esto en mente se accede a la herramienta de Visualización y Análisis (SIGNA) del *Instituto Geográfico Nacional, s. f.*

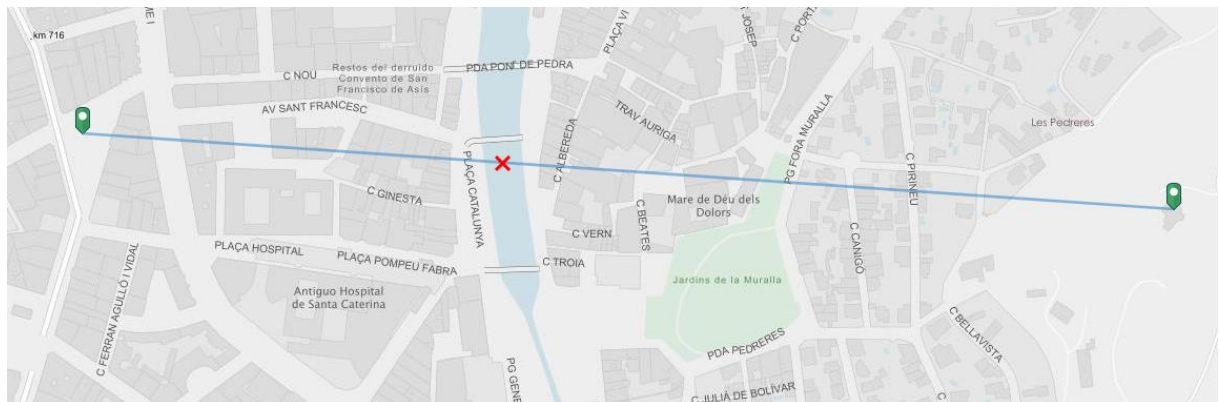
Figura 12. Ubicación de la Gateway en la Torre de Alfons XII de la ciudad de Girona.



Fuente: Elaboración propia y mapa del *Institut Cartogràfic i Geològic de Catalunya*, s. f..

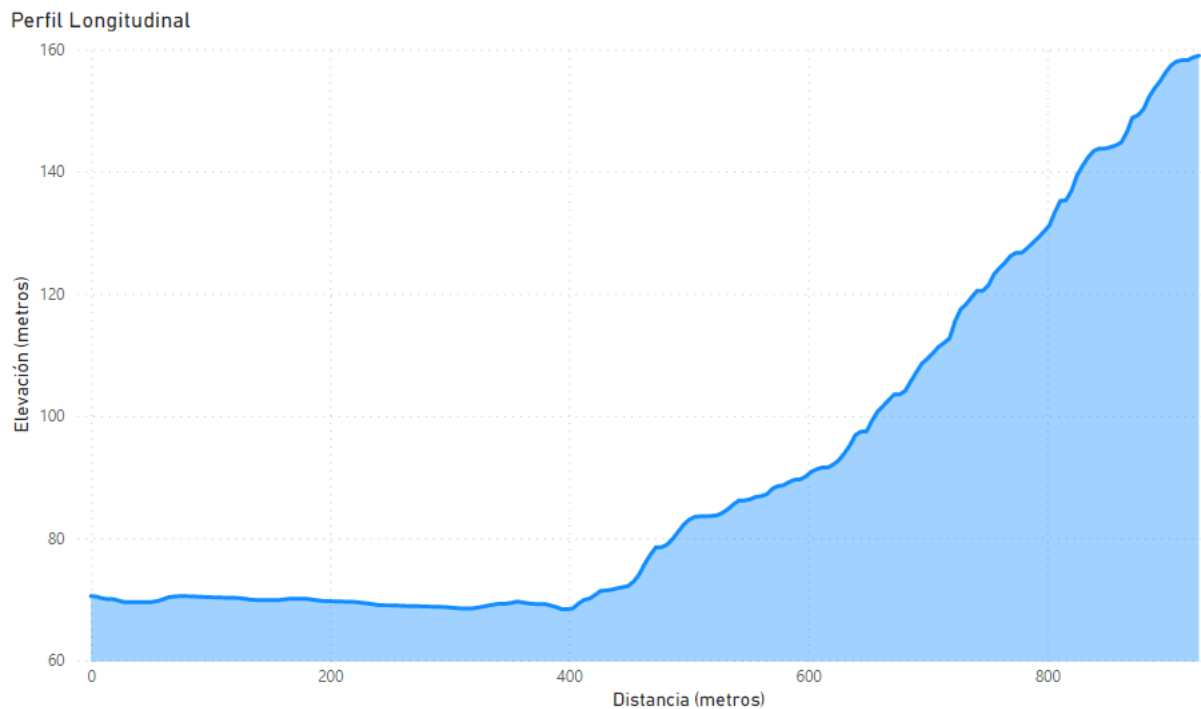
Con la herramienta de SIGNA se traza una línea que une la Torre de Alfons XII, Plaza Catalunya y un punto sustancial de la ciudad a unos centenares de metros. La línea referida se muestra en la Figura 13, donde el aspa roja (x) roja marca la plaza.

Figura 13. Trazado longitudinal de Plaza Catalunya y colindantes.



Fuente: Elaboración propia, mapa de *Instituto Geográfico Nacional*, s. f.

Los datos son extraídos en formato coma flotante (.csv) quedando representados en la gráfica de la Figura 14. Como se puede observar, en la parte Oeste de la ciudad el terreno es prácticamente homogéneo, analizando los datos de cerca se aprecian variaciones del entorno de uno a dos metros. Por otra parte, en el Este la inclinación del terreno es más abrupta, pasando de los 68.40 metros de la plaza hasta los 160.20 donde se encuentra la torre. La plaza se encuentra justo en la línea referencial de los 400 metros en el eje de la distancia.

Figura 14. Perfil longitudinal de Plaza Catalunya y colindantes.

Fuente: Elaboración propia.

Referente al proyecto, se espera que las plazas de aparcamiento tengan un tiempo limitado de 15 minutos, no quedando especificado su uso concreto, así que para este proyecto se interpreta como uso genérico para todas las plazas de aparcamiento. Además, el horario de implementación de esta tarifa se aplica en el horario restringido desde las 9:00 de la mañana hasta las 21:00 de la noche, 12 horas, el resto del día no importa si se detectan coches. El resto de información relevante al sistema quedan descartados por la nulidad total de su uso en este proyecto como, tiempos de respuesta y actuación del servidor, y demás.

4.1.2. Dispositivos y tecnología

Los dispositivos pertenecientes a este proyecto son cuatro, tres sensores (los candidatos) y la pasarela, contenidos en la Tabla 6. A continuación, se detallan algunas de las características y métricas más importantes de cada uno de ellos.

En el Anexo C *Datos de implementación*, se encuentran distintas tablas con todos los elementos resumidos, la Tabla 13 engloba las características de los sensores (modelo, FOTA y calibración, entre otros) mientras que en la Tabla 14 se describen las métricas asociadas (Tiempo de detección, duración teórica de la batería, capacidad de esta y demás). Finalizando con la Tabla 15 reuniendo los aspectos relevantes de la pasarela.

Empezando con el sensor ofrecido por *IoTens*, cabe remarcar que esta empresa se ofrecía hasta hace poco como distribuidor del modelo de aparcamiento AN-101D del fabricante Winext Technologies y se está empezando a sustituir este modelo por uno propio. Este dato, de especial relevancia, no conforma un papel decisivo y se desestima para este proyecto, ya que solo se tienen en cuenta los detalles técnicos referentes a su aplicación para el proyecto del aparcamiento. Aun así, es de vital importancia detallarlo para evitar compromisos.

Tabla 6. *Dispositivos propuestos para el proyecto.*

Dispositivo	Distribuidor	Modelo
Sensor	IOTSens	AN-101D Winext
	BOSCH	Parking Lot Sensor
	Libelium	Modelo Radar/Magnético Dual
Pasarela	MultiTech	MTCDTIP-266A-868

Fuente: *Ajuntament de Girona*, s.f. (2022).

El modelo de *Winext* s. f. (Figura 15) emplea un sensor magnético para la detección de los vehículos de distintas dimensiones. Su montaje en superficie (con una protección IP 67) requiere del uso de tornillos que van sujetos a la superficie de acoplamiento, aunque también recomiendan el uso de resinas adhesivas para un máximo agarre y seguridad. Además, este dispositivo contiene una batería Li-SOCL2, intercambiable, con una capacidad considerable de 16 000 mA·h. El fabricante, incluso, se atreve dar una relación en su duración, de 4 años, con la media de mensajes por día, unos 20 diarios, cosa que en la mayoría de los casos nunca se hace.

El módulo del dispositivo funciona con una alimentación regulada a 3.30 V (variando entre los 2.90 V y 3.60 V con la batería cargada) y tiene unos consumos teóricos (sin especificar tiempos) en funcionamiento de ≤ 125 mA, de envío ≤ 15 mA (corriente Rx) y de "Idle current" ≤ 15 μ A.

El módulo de comunicaciones LoRa se especifica con el uso del chip de SX1276 (*Semtech LoRa Technology Overview | Semtech, 2023*) el cual, define el dispositivo de tipología A dentro de las comunicaciones LoRa. Las características básicas de estas comunicaciones son el rango del SF del 7 hasta el 12, la potencia de transmisión (Tx) de 2 a 14 dBm y las funcionalidades de ADR y ACK. A parte, ofrece diferentes características de funcionamiento que ayudan en la gestión y mantenimiento de los dispositivos, la función “Heartbeat”, por ejemplo, puede ser programada desde los pocos minutos a diversas horas, para que mande un mensaje a través de las comunicaciones LoRa indicando su estado.

Este dispositivo se configura y calibra usando un software junto a un dongle específico, el cual se incluye con el pedido de los dispositivos. Este programa permite configurar características tales como la banda de transmisión (como la ISM-Band 863-868MHz EU), el periodo de “Heartbeat”, la activación del ACK o ADR, entre otros. Este se conecta usando el *dongle* a través del módulo de comunicaciones Bluetooth integrado en este y el sensor.

Figura 15. Modelo AN-101D Winext.



Fuente: *Winext*, s.f.

Volviendo al sensor en sí, este destaca por tener un rango de detección máximo en torno a 1 metro de distancia, aunque el mínimo es de unos 10 centímetros. El ciclo de detección de este dispositivo es de unos 20 segundos (detección, más la modulación de la señal) y cuenta con un sensor extra de temperatura. Para finalizar, cuenta con un “Payload” de 11 bytes de información no modificables, representados en la Tabla 16 de los anexos.

El modelo de BOSCH (Figura 16) emplea dos sensores para la detección de vehículos, un radar y un magnetómetro, aunque el radar es el elemento principal de detección, mientras que el magnetómetro se usa a modo de respaldo con tiempos de detección inferiores a los 35 segundos y rango de entre 20 centímetros y 1 metro (según documentación técnica). La

instalación de este dispositivo también se realiza en la superficie, pero a diferencia de los demás, solo usa resinas, directamente en la zona de implementación, además ofrece una protección contra elementos IP 67. A diferencia de los otros dos modelos, este se compone de cuatro elementos, la base, el sensor (Core), el tornillo de unión y su tapa, estos se muestran en la Figura 17.

La instalación de este dispositivo no es tan invasiva como el resto de los sensores aquí presentados. Este solo requiere de resinas específicas para hacer de unión entre la superficie y la base, el sensor se encaja en la base y se sujeta con el tornillo, haciendo uso de su tapa para imposibilitar el acceso a este último.

Figura 16. *Modelo Parking Lot Sensor BOSCH.*



Fuente: *BOSCH*, s.f.

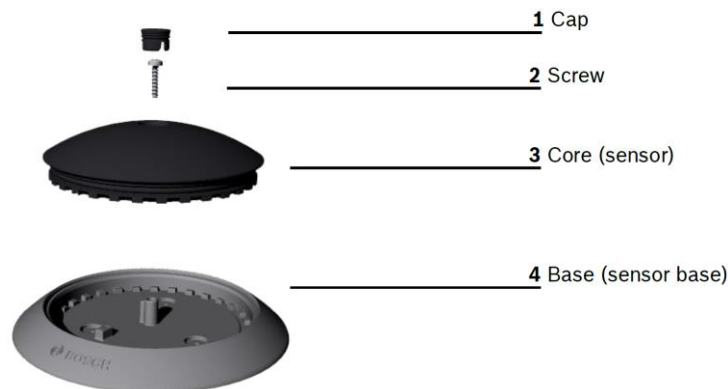
En cuanto a la calibración del dispositivo, esta es automática. El dispositivo lleva integrado un sistema de aprendizaje (no especificado), previamente entrenado, que adapta su detección a los 10 primeros vehículos que realizan un estacionamiento encima del sensor. Esta calibración se activa al acoplar el sensor en la base, detectando un imán situado en esta. Además, no requiere de mantenimiento, ya que es capaz de detectar una variedad de errores relacionados con la calibración, los cuales, en caso de producirse, generan el reinicio de esta y, lo interesante, también en las comunicaciones LoRa. En esta situación se vuelven a precisar de los 10 estacionamientos anteriormente indicados.

La batería de este dispositivo es la menor de los tres, se presenta con una batería Li-SOCL2 muy pequeña (comparada con el resto) de tan sólo 1200 mA·h. La autonomía se espera que sea de hasta 5 años, suponiendo una configuración del dispositivo proclive a la batería, con SF 7 y 200 mensajes (“Uplinks/Downlinks”) a la semana, más o menos 28 mensajes al día.

El módulo sensorial del dispositivo se alimenta a través del voltaje de la batería y a la misma tensión que el dispositivo de Winext. Los consumos de este sensor no aparecen reflejados en ningún documento que este autor haya podido encontrar.

Por parte del módulo de comunicaciones LoRa del dispositivo BOSCH, este no informa del chip usado, pero si da las características típicas de un dispositivo modelo A de LoRa, como el poder de transmisión máximo (Tx) de 14 dBm y demás. Además, a diferencia de los otros modelos observados en este trabajo, BOSCH da las condiciones para las cuales este sensor fue diseñado. Estas son, determinar un factor de dispersión (SF) ≤ 10 (aunque el rango es total, cubre del 7 al 12), escenarios donde el nivel de potencia de la señal recibida (RSSI, *Received Signal Strength Indicator*) sea > -120 dBm y un SNR > 7 dBm.

Figura 17. Componentes del dispositivo BOSCH.



Fuente: *MultiTech Conduit IP67 LoRa Gateway | Outdoor LoRa Deployment*, s. f.

Antes de terminar con el sensor de BOSCH, es imperante remarcar la actual realización de pruebas hechas por el ayuntamiento. Este dispositivo ya ha sido adquirido por el ayuntamiento y se han realizado algunas pruebas en distintas zonas de la ciudad de Girona. Los resultados de estas destacan por su fácil instalación e implementación en el sistema, pero a la vez, destaca un alto fallo de detección de ciertos vehículos junto a mayores tiempos de detección de estos, entre los 40 y 60 segundos, en este caso contradiciendo los tiempos típicos de detección de la documentación los cuales, indican un tiempo habitual de alrededor de los 35 segundos. Estas pruebas no disciernen entre vehículos utilitarios o de transporte de mercaderías de peso liviano (furgonetas pequeñas o de reparto).

Finalizando con el modelo sensorial Dual de Libelium (Figura 18), como su nombre indica, este emplea una modalidad dual entre radar y magnetómetro y su funcionalidad es la misma que

para el dispositivo de BOSCH. La instalación es en superficie, de la misma manera que el modelo de Winext, va atornillado al suelo junto al empleo de resinas. La protección de este dispositivo aumenta respecto a los modelos anteriores, ofreciendo la IP 68. El modelo de la batería de este dispositivo tampoco varía (Li-SOCL2) respecto a los anteriores, aun así, esta es intercambiable, su capacidad es de 10 400 mA·h y su duración se estima del entorno de 4 hasta los 10 años dependiendo de los ciclos de “DeepSleep” y funcionalidades (ACK, “Payload”, entre otros).

El dispositivo es alimentado a un voltaje de 3.30 V (Vcc), el rango de operación se estima entre los 2.90 y 3.60 V de alimentación de la batería. Los consumos varían según uso, estos se encuentran mayormente documentados, en este caso se dan los consumos en el peor escenario y mayor tiempo de transmisión (SF12). Para el envío (“Uplink”) se especifica una media de 17 mA en 6 s y cuando este entra en “DeepSleep” el consumo se reduce a 5.50 μ A.

Este modelo contiene funcionalidades programables para el ahorro de energía, tales como el modo de noche. Este permite determinar un horario donde generar periodos de “DeepSleep” más prolongados en el tiempo y que además desactiven totalmente las funcionalidades de detección, consiguiendo un consumo aún menor. Esto se consigue gracias a un módulo RTC integrado, el cual, permite obtener series temporales de tiempo independientes de la recepción en la pasarela. Además, cuenta con la funcionalidad “Heartbeat”, intervalos programables desde minutos hasta horas.

La información respecto el módulo no es muy abundante, pero tampoco es poca, ya que se indica la posibilidad de implementación y uso de este sensor en el Cloud de Libelium o el uso del programa de configuración inicial. En este caso las comunicaciones están adaptadas para la banda ISM (Industrial, Scientific and Medical) de la EU. El dispositivo se caracteriza por ser de la tipología A en las comunicaciones LoRa y contiene los mismos aspectos básicos que los modelos anteriormente descritos.

Libelium, s. f. llama a los 9 bytes de información “Payload”, aunque en este trabajo este nombre se reserva para indicar la carga de datos que es enviada por las comunicaciones LoRa los cuales, en este dispositivo se emplean una totalidad de 11 bytes, 2 bytes para un “Header” y 9 de datos.

Figura 18. *Modelo Dual Libelium.*



Fuente: Libelium, s. f.

El “Payload” estándar para una transmisión para la detección de vehículos está conformada siempre por 11 bytes de datos, representados en la Tabla 18 de los anexos. Su información no es muy detallada, pero los 2 primeros bytes que representan el cabezal indican la modalidad del mensaje, mientras que los 9 bytes restantes contienen de métricas del sensor. En esta información se encuentran la detección del vehículo, el estado del sensor y distinta información referente a la configuración de este o del módulo de comunicaciones LoRa, entre otros.

La configuración del dispositivo emplea el uso de la aplicación de Libelium a través de OTA o directa (cables micro-USB) hacia el sensor. Hasta donde se ha podido observar la calibración es automática cada vez que realiza una configuración nueva. Cabe tener en cuenta que ciertos aspectos clave de la configuración hacen imprescindible realizarlas a través del cable micro-USB y no se pueden realizar a través de OTA. Asimismo, una vez instalado en la ubicación, la activación de este dispositivo requiere del uso de un imán para que quede en estado activo. Esta última singularidad lo deja en un estado donde cada vez que se quiera realizar un “reset” del dispositivo solo hará falta pasar una sola vez el imán por encima y el sensor cargará el Firmware guardado en su memoria.

Con los sensores descritos y su información recogida se procede a recoger la información para la pasarela utilizada en este trabajo. Así pues, MultiTech cuenta con una modalidad llamada Conduit, en concreto el modelo utilizado se trata del MTCDTIP-266A-868. Las características y métricas más importantes son recogidas en la Tabla 15 de los anexos. Además, MultiTech emplea este tipo de pasarelas para la tecnología de LoRa/LoRaWAN permitiendo la creación de un enlace entre las comunicaciones LoRa y Ethernet redirigiendo los mensajes de la red

privada de LoRa, hacia el servidor pertinente (Network Server), en este caso propio del ayuntamiento.

Este modelo de pasarela en concreto pertenece a la tipología de uso para exteriores y cuenta con una protección contra los elementos IP 67. Su alimentación se hace a través del mismo cable Ethernet, aprovechando la tecnología PoE (Power over Ethernet), por el cual es necesario el uso de un modelo PoE+ con una potencia de 30W para garantizar el correcto funcionamiento de esta.

Figura 19. Pasarela MultiTech Conduit.



Fuente: *MultiTech Conduit IP67 LoRa Gateway | Outdoor LoRa Deployment*, s. f.

El módulo de comunicaciones de este dispositivo cuenta con un chip de Semtech SX1301 capacitado para el uso de las bandas ISM en EU. Su poder de transmisión máximo puede establecerse en 14 o incluso 27 dBm, en aquellos casos puntuales. Este modelo incorpora una antena de comunicaciones LoRa omnidireccional con una ganancia de 3 dBi. Además, añade una antena GPS que permite la creación de “Metadatos” para cada trama de mensajes y que de esta manera se añade información extra en aquellos dispositivos IoT sin estas capacidades, como es el caso de todos los dispositivos sensoriales de este proyecto. Por último, el chip de Semtech implementado en esta pasarela es el SX1301.

4.1.3. Objetivos y requerimientos del ayuntamiento

Principalmente, existen dos objetivos clave para el ayuntamiento, la selección del mejor sensor dentro de los candidatos propuestos y, encontrar un modelo que dé con un equilibrio entre cobertura y consumo del sensor. Los OKR y KPI que el ayuntamiento requiere quedan expuestos en la Tabla 7.

Algunas preferencias no salen en esta tabla debido a su carácter y relevancia general. Por ejemplo, el ayuntamiento considera de gran aprecio no tener que desmontar los sensores, a

excepción de cuando sea necesario (cambio de batería, cesación del dispositivo, entre otros). En otros aspectos, el ayuntamiento comprende la posibilidad del empleo del sistema ACK (*Acknowledge*, para la reducción de pérdida de paquetes) de LoRa, pero no lo considera fundamental siempre que los mensajes tengan una bajísima tasa de error. Además, que el sensor disponga de información referente a su estado (“Metadatos”) como el nivel de la batería u otros resulta interesante.

Tabla 7. OKR y KPI del ayuntamiento de Girona.

<i>Objective Key Results</i>	<i>Key Performance Indicators</i>
Elección del mejor sensor candidato	• Duración de la batería ≥ 2 años
	• Calibración por OTA
	• Tiempo de detección del vehículo inferior a los 30 segundos
	• Rango de detección >500 mm
	• Protección contra los elementos \geq IP 67
Encontrar un equilibrio entre el modelo de cobertura y el consumo del sensor	• Error de transmisión $\leq 1\%$
	• Determinar el menor SF posible, \leq SF12
	• Determinar el menor Tx posible, \leq Tx 14 dBm
	• Disponer del ADR (<i>Adaptive Data Rate</i>)
	• Disponer del ACK (<i>Acknowledge</i>)

Fuente: Elaboración propia.

4.2. Fase 2: Selección de la información

En esta segunda fase se abordan dos temas fundamentales para la proyección final de los resultados, la selección del dispositivo sensorial candidato y los límites y datos referentes en la simulación, quedando dividido en las secciones pertinentes. La Subsección 4.2.1 procede

con la elección del candidato para su futura representación en la simulación, mientras que en la Subsección 4.2.2 comprende el marco referencial sobre el cual se basan los modelos y visualización de los datos empleados.

4.2.1. Estudio del sensor candidato

La realización de la primera fase origina una visión de los aspectos verdaderamente interesantes o que generan repudio para la selección de los distintos candidatos, llevada a cabo en esta Subsección. Por este motivo, en primer lugar, se argumentan los elementos característicos que conciben la eliminación de uno de los candidatos (BOSCH), posteriormente, genera una comparativa entre los candidatos restantes (Winext y Libelium) y finaliza con la elección del dispositivo (Winext).

Así pues, el primer candidato por el cual se descarta su uso es el dispositivo BOSCH, empleando como uno de los principales razonamientos la seguridad en la implementación y colocación de este. Aunque presenta una mayor facilidad de implementación (resina) el hecho de poder quitar la tapa y desatornillar el sensor de su base empleando un simple destornillador lo deja demasiado expuesto al vandalismo. Esto lo hace ideal en implementaciones con entornos controlados como, cámaras de vigilancia, entornos cerrados u otros, no siendo este el caso.

El segundo motivo son las pruebas realizadas por el ayuntamiento, previas a este trabajo, ratificando tiempos de detección superiores a los previstos por la documentación y, una elevada tasa de error en la detección de vehículos de la cual, existen indicios que sean debidos al entrenamiento que recibe el dispositivo durante su calibración. Esto lo hace sensible a los aparcamientos aleatorios que realicen las personas en una futura implementación, añadiendo una arriesgada e incierta aleatoriedad en su comportamiento. Asimismo, este factor imposibilita conseguir el KPI relacionado con los tiempos de detección inferiores a los 35 segundos.

Referente a los dos primeros motivos, se añade la modalidad y uso del algoritmo de entrenamiento para la calibración automática del dispositivo y el reinicio de esta con el imán situado en la base. El hecho que pueda ser separado de su base de una forma tan fácil lo hace vulnerable y reinicia su calibración instantáneamente, ya que se separa el imán entre base y sensor. Aunque, también es cierto que sucedería algo parecido para los otros candidatos si se mueven cierta distancia respecto su centro de calibración.

Dicho esto, en segundo lugar, existe la reactivación del dispositivo dentro de la red de LoRa cada vez que se reinicia esta calibración. Es necesario detallar que el dispositivo no manda mensajes LoRa durante las 10 detecciones realizadas durante la calibración, sino que lo hace al finalizar. Esto añade mensajes innecesarios (“Uplinks” y “Downlinks”), a diferencia del resto de dispositivos que se pueden calibrar sin reiniciar las comunicaciones, cargando la red de tráfico y afectando al consumo del dispositivo.

Por otra parte, el hecho de generar un solo byte de “Payload” le favorece drásticamente frente a los otros candidatos, aunque queda desaprovechado en gran medida por la baja capacidad de la batería. Además, aunque los consumos no se especifican, las condiciones para la duración de la batería son estrictamente reseñadas para alguno de los mejores casos y más infrecuentes como $SF \leq 10$ y los 28 mensajes de media al día, hecho contraproducente porque añade una mayor incertidumbre en la posible duración de la vida útil del sensor que si lo fuese en el peor de los casos.

Con el dispositivo BOSCH descartado, a continuación, se analizan los puntos fuertes y débiles de los sensores de Winext y Libelium. En primer lugar, el dispositivo de Winext presenta la mayor capacidad de batería lo que en principio indica una mayor duración. A este hecho cabe remarcar la durabilidad del dispositivo, de 4 años, con una media de 20 mensajes por día (Uplinks o Downlinks), con el mínimo de calibraciones posibles y el peor escenario (SF12 y Tx a 14 dBm), aumentando la fiabilidad de este dato.

Aun así, la documentación parece indicar un alto consumo por parte del módulo de computación, añadiéndole el hecho de que se trata del mayor “Payload” de los candidatos (junto a Libelium) el cual, no puede ser modificado. Esto puede ser indicación para una menor duración, justificando el aumento de la capacidad de su batería para mantener la duración del dispositivo y no para alargarlo.

El “Payload” representa un factor clave en este dispositivo, su envergadura no se puede modificar, esto aboca a dos enunciaciones, tiene un mayor consumo y, los datos que se envían deben tener una mayor relevancia, aumentando la necesidad para la justificación de estos. Por ello, de todos los datos mostrados en la Tabla 16 de los anexos, existen 10 bytes drásticamente justificados, mientras que existe uno más difícil.

El byte más delicado es el primero, ya que solo indica el tipo del dispositivo. Winext, cuenta con una amplia variedad de dispositivos, con la intención de normalizar los datos de envío entre todos ellos se emplea este dato. Esto lo hace ideal en aquellas aplicaciones y proyectos que tengan dos cosas, complejidad de sistemas, por lo tanto, multitud de dispositivos y que, además, todos ellos sean del mismo fabricante, Winext. Al no ser este el caso provoca una ambigüedad en su uso.

Seguidamente se dan razones para la justificación del resto del "Payload". El byte número 1 contiene 4 bits de "Frame" y 1 de "Uplink" o "Downlink". El primero indica el patrón del mensaje, facilitando su identificación dentro del sistema de gestión de los dispositivos, permitiendo identificar si se trata de un mensaje, de detección, fallo interno u otros. El segundo, sencillamente indica direccionalidad. A continuación, del segundo byte hasta el noveno se indica la calibración en los ejes X, Y, Z, así como la temperatura del dispositivo. Todos ellos resultan interesantes porque abren la puerta a su monitorización para posibles sistemas de mantenimiento predictivo (u otras) y, de esta manera detectar, desde posibles derivas en la calibración, hasta futuros errores en su funcionamiento.

Terminando con el último byte el cual indica la detección de un vehículo y el voltaje de la batería, indispensables por el simple hecho de contener el motivo principal de su uso y dar más motivos a una posible utilización de sistemas de mantenimiento.

El uso de un magnetómetro de 3 ejes para la detección de vehículos en superficie representa una elección ideal para este tipo de aplicaciones (Al-Turjman & Malekloo, 2019), no le afectan las condiciones ambientales y resulta ser un sensor barato. Por el contrario que los demás candidatos solo emplean un sensor, por ello no cuenta con un doble factor de detección.

Por otra parte, existen pocos modos de funcionalidad (como el "Heartbeat") y el dispositivo se trata de una caja negra más allá de la configuración posible, como el recambio de la batería o la configuración a través del programa. Esto abre incertidumbres, como la poca capacidad de reparación y modificación en su comportamiento para escenarios concretos.

La instalación de este dispositivo presenta una mayor complejidad, en comparación con el dispositivo de BOSCH, pero su activación y calibración ofrecen una mayor agilidad. En este sentido, una vez desplegado, se emplea el *dongle* junto al programa gratuito para su configuración y activación en la red LoRa. La configuración puede realizarse con

independencia a su instalación, mientras que la calibración se debe realizar una vez ya hecha. Si es la primera vez, se puede hacer al mismo tiempo, reduciendo tiempos de uso en el Bluetooth y en personal por lo cual, tener un menor consumo energético y económico.

En concreto, la calibración puede realizarse en cualquier momento sin tener que reiniciar las comunicaciones LoRa, manteniendo la activación dentro de la red, ahorrando mensajes innecesarios. Esta calibración se realiza con la condición de no tener ningún vehículo en la plaza de aparcamiento y, a ser posible, en los colindantes, para una mayor precisión en su detección. Al no tener ningún otro método de interacción con el dispositivo, este queda a merced del programa y especialmente el *dongle* el cual emula un único punto de fallo o *Single Point Of Failure* (SPOF) en su utilidad.

Una vez terminada la evaluación del dispositivo de Winext se procede con el de Libelium. En primera instancia, cabe remarcar la similitud de la batería con la de Winext aunque con una capacidad menor, 10 400 mA·h. Este fabricante no indica los mensajes por día, pero si argumenta que los años mínimos de duración (4 años) se han calculado respecto el peor de los casos (SF12 y Tx de 14 dBm). Así pues, se encuentra en una fiabilidad razonable, situado en medio del Winext y BOSCH, ya que la capacidad de la batería es menor que la de Winext, un menor consumo y a la vez mantiene la misma extensión de “Payload”.

El empleo de dos sensores para la detección de vehículos es interesante ya que añade un doble factor de seguridad. El radar presenta unos resultados aceptables para este tipo de aplicaciones (sin saber el modelo específico) y que mejora en situaciones de climatología adversa, aun así, el magnetómetro representa ya de por sí muy buena solución para esta aplicación, en las mismas condiciones que el radar, este ofrece una mayor precisión (aun sin saber el modelo de ejes que utiliza) y a un menor coste (Al-Turjman & Malekloo, 2019).

El “Payload”, para el caso de Libelium, cuenta con el mismo paradigma que el de Winext, es imperioso tratar de justificar su uso. Era de saber que la versatilidad en su configuración tendría alguna que otra desventaja, una de ellas se ve reflejado en el “Payload”, si no se requiere llevar un control de su funcionalidad en cada momento, los dos primeros bytes son bastante innecesarios, incidiendo directamente en el consumo del dispositivo.

Además, el tipo de “Payload” generado para la detección del vehículo, como es el caso en la Tabla 18, no indica directamente la detección del vehículo, sino que debe ser interpretada por

la distancia, amplitud y número de reflexiones. Dificultando la computación de los datos para la detección de un vehículo entrante o saliente.

El resto de los bytes se dividen en errores del sensor, Temperatura y Timestamp. En cuanto al Timestamp y el error, cuesta encontrar un uso directo para ellos. En primer lugar, el Timestamp solo contempla las horas y los minutos, los cuales pueden ser sustituidos por los metadatos que incorpora la gateway, incluso si llega a no utilizarse el ACK de las comunicaciones LoRa, esta pierde aún más razón. En el lugar del error del sensor, Libelium opta por priorizar la funcionalidad y no parece tener un mensaje (“Uplink”) dedicado específicamente para informar de los posibles errores, como si hacen el resto de los candidatos. Para finalizar, la Temperatura puede ser empleada de la misma forma que se ha argumentado para el dispositivo Winext.

La gran versatilidad de uso y configuración del dispositivo se asemeja a un antagonista respecto al dispositivo de Winext. Sus configuraciones en distintos modos de energía (modo de noche) y otros, ofrecen una mayor versatilidad en su configuración, uso y ahorro de energía cuando, por lo contrario, requieren de un mayor tiempo en su configuración junto a una curva de aprendizaje mayor.

Un factor relevante de las distintas funcionalidades es el uso del modo nocturno permitiendo desactivar totalmente las funcionalidades de detección y demás. Esto permite minimizar el consumo durante periodos largos de inactividad. Muy adecuado y apreciado para este proyecto ya que podría ser implementado durante el horario estipulado en la descripción del escenario de la Subsección 4.1.1.

A parte, este dispositivo permite acceder a su interior de forma fácil e intuitiva. Libelium lo recomienda y además es necesario para la configuración inicial del dispositivo. Este dispone de un puerto micro-USB en su interior junto a un interruptor para su encendido, permitiendo realizar cambios de batería o carga de configuraciones y actualizaciones del Software, entre otros.

Aunque es cierto que para la configuración inicial es imprescindible acceder al dispositivo por micro-USB y hacer uso de un programa de escritorio hecho con Java, una vez realizada, se pueden realizar ciertos cambios de configuración por OTA e incluso FOTA. Para realizarlo, hay herramientas online propias de Libelium, existen dos, una permite conectarse a la red de

LoRaWAN directamente y enviar los cambios de la configuración a través de mensajes LoRa. La segunda, es una plataforma Cloud de Libelium que se une a los sistemas de gestión de dispositivos que puedan albergarse tanto en la nube pública, como Azure, AWS u otros y la privada, donde le manda la información para que prosiga el camino allí estipulado.

Los programas que permiten la configuración por OTA, incluso FOTA, son de gran valor para proyectos en que se requiera gestionar una gran cantidad de dispositivos, permitiendo generar programas específicos de configuración y actualización automática. Al ayuntamiento le resultaría muy interesante este hecho, especialmente con una visión de futuro, si decidiese implementar más sistemas parecidos.

En cuanto a la calibración del dispositivo, este cuenta con un sistema automático que calibra el dispositivo una vez detecta las pasadas de un imán por la carcasa superior del dispositivo. Esto lo reinicia (in argumentar específicamente si se reinician las comunicaciones LoRa), cargando de nuevo los datos de configuración inicial y la calibración. Esto requiere que el emplazamiento este vacío y a ser posible que los colindantes también, mejorando la precisión.

Junto a esta funcionalidad se ahorra tiempo en calibraciones, pero deja una puerta abierta a posibles vandalismos que debería ser explorado con más profundidad si este llega a ser el dispositivo escogido. A parte, el consumo de energía no es excesivamente reducido como uno se puede esperar ya que el sistema reinicia el dispositivo completamente, cargando incluso el "firmware" de nuevo.

Con los aspectos más importantes reseñados, a continuación, se dan las razones finales para la elección del dispositivo. Antes de nada, cabe destacar que ambos dispositivos tienen sus pros y contras, destacando en aspectos distintos el uno del otro, aunque parten de una base bastante parecida.

En cuanto el consumo, Winext tiene uno mucho mayor que Libelium, aun así, la batería no sigue la misma proporcionalidad su capacidad, en otras palabras, aun con un consumo más de tres veces superior, su batería no lo es, quedando con una vida útil teórica igual a la de Libelium. Si bien es verdad que Winext requiere de un módulo de Bluetooth que puede generar aumentos de consumo durante las calibraciones o configuraciones, estas pueden ser asimiladas a la configuración por LoRa que usa Libelium. A parte, Libelium puede hacer uso de su funcionalidad de noche y ahorrar cierta energía, aunque no se arriesga a determinar los

mensajes por día medios para garantizar tales duraciones, en este aspecto Libelium toma la delantera.

En cuanto al uso de los sensores, Winext cuenta uno de los mejores sensores para la detección de vehículos, mientras que el sistema de Libelium emplea el radar en primera instancia, de menor precisión y respaldándose con un magnetómetro en segundo plano. Añadir, que el rango de detección de ambos es claramente aceptable, Winext tiene una mayor amplitud de detección desde los 10 centímetros hasta el metro, mientras que Libelium necesita los 20 primeros centímetros. Al tener unas diferencias tan poco sustanciales, este factor que neutralizado ya que los dos pueden servir perfectamente. En este caso se Winext brilla por su simplicidad y por tener un mayor rango de detección útil de vehículos que el de Libelium.

El aparcamiento de plaza Catalunya es un lugar transitado y donde existe una alta demanda de aparcamiento, por ello suele ser normal que cuando un vehículo abandona su plaza, el siguiente en llegar tarde poco o incluso ya esté esperando. Esto requiere ciclos de detección rápidos, afectando claramente al consumo, en este caso Libelium lidera la detección en ciclos que bajan hasta los 16 segundos, 10 segundos de ciclo de detección más 6 segundos para el ciclo de envío de la señal por LoRa. Winext no se queda corto y realiza un ciclo similar con una duración de 20 segundos.

Aunque Libelium pueda tener mayor periodicidad, los dos se encuentran muy por debajo del KPI detección del vehículo ≤ 35 segundos, el cual se basa en el periodo que tarda un coche en salir de su plaza y en que el siguiente entre (realizando pruebas *in situ*).

Respecto a la calibración y configuración Winext destaca por tener un modelo muchísimo más sencillo, aunque menos versátil. Este, permite su instalación directa y fácil en solo dos pasos, primero su instalación para luego realizar su activación dentro de LoRa, configuración y calibración inicial a la vez, haciendo uso del *dongle* y programa propio *in situ*. Con Libelium, este proceso requiere de un mínimo de tres pasos y el acceso directo al sensor para su primera calibración, retardando el despliegue y posterior calibración.

Además, el ayuntamiento no da tanta importancia a la gestión automática de los dispositivos hoy por hoy y acepta tener que desplazarse a la zona de despliegue para realizar calibraciones o cambios, siempre que no sea necesario acceder directamente al dispositivo. Aparte, también valora la facilidad de uso del *dongle* y su programa. Para el caso del dispositivo de Libelium,

no le genera mucha confianza poder reiniciar el dispositivo de manera tan fácil con un imán, pudiendo generar situaciones no deseadas. En este punto Winext se escoge para este ámbito.

Por parte de las comunicaciones, Winext presenta un sistema de mensajes mucho más intuitivos y justificados que el de Libelium. Son directos, concisos y no requieren de computación extra, con un sencillo "Payload Decoder" es más que suficiente. En este sentido puede monitorizar todos sus aspectos más relevantes en cada mensaje de detección de vehículos, cosa que con el dispositivo de Libelium debes esperar a recibir diferentes tipos de mensajes. Por ejemplo, para saber el estado de la batería, con Libelium, se debe esperar al mensaje de "Heartbeat" para saberlo. Winext se adelanta en este aspecto, por su sencillez y efectividad.

Finalmente, con todos los argumentos dados en esta subsección, Winext es el candidato que este proyecto selecciona para su uso en el proyecto del ayuntamiento y, sus datos y características van a ser empleados para el modelo de simulación de la Fase 3. Además, este sensor va a ser adquirido, esto conlleva poder realizar ciertas pruebas físicas para poder contrastar datos usados para el diseño e implementación de la simulación.

4.2.2. Métricas y características asociadas a la simulación

Una vez el candidato ha sido seleccionado, esta subsección pretende recoger los datos de este dispositivo y del escenario que son de utilidad para la realización de la simulación, de la misma forma argumenta las razones por las cuales descartar o se adaptan ciertos datos.

En cuanto al emplazamiento del proyecto hay que destacar distintas causalidades y escenarios siguientes. Primero de todo, los datos del trazado longitudinal que se han recogido se simplifican, la plaza y el lado Oeste representan un terreno plano y sin variabilidad. Mientras que, en el lado Este, empezando por la primera hilera de casas adjuntas a plaza Catalunya tienen la elevación pertinente al pendiente extraído entre este punto y la torre de Alfons XII, luego de asimilar los datos a un triángulo rectángulo.

En segundo lugar, la posición de todos los dispositivos (sensores y gateway) y los edificios generados tienen como referencia absoluta el primer aparcamiento para gente de movilidad reducida situada al Noroeste del emplazamiento. Su posicionamiento se trata en la fase correspondiente, en este caso, la Fase 3.

En tercer lugar, las alturas de los dispositivos sensoriales, al tratarse de instalaciones en superficie, su altura se referencia respecto la posición de la antena con el suelo. Esta se extrae de la altura del dispositivo elegido (Winext) ya que la antena se sitúa en la parte superior. Además, se respeta la posición en la cual se debe instalar el sensor dentro de la plaza de aparcamiento, justo en medio, respecto a la anchura y, a tres quintas partes de la longitud, a ser posible en la dirección que apunte el coche (mayor detección con el bloque motor). Así pues, la separación entre estos no solo se realiza según la posición de los aparcamientos, sino también respecto la posición que serían instalados en ellos.

Al haber escogido el sensor de Winext, los datos del consumo, el voltaje, la periodicidad de detección, la capacidad de la batería y la longitud del "Payload" son empleados. Además, todas las características pertenecientes al módulo de comunicaciones LoRa posibles (SF, Tx, Chip, entre otros). Otros datos como el rango de detección, la calibración, el modelo de la batería y demás no son empleados.

En cuanto a la altura referencial de la gateway se realiza la operación necesaria que sustrae o añade la altura referencial de los puntos geodésicos (torre Alfons XII a 160.20 m) como la altura de la torre (a 18 m) y la del soporte (a 2.50 m), quedando a una altura de 112.30 m, respecto plaza Catalunya (a 68.40 m).

Para finalizar, se remarcan un seguido de aclaraciones, en primer lugar, se tiene en cuenta que los periodos en que se mandan mensajes en cada uno de los escenarios simulados se realizan de forma procedural, empezando por periodos de tiempo muy bajos (el mínimo, 20 segundos), indicando una alta cantidad de tránsito, finalizando con tiempos largos, todo lo contrario (hasta los 15 minutos del límite tarifado), representados en la Tabla 21 de los anexos. La tipología de estos mensajes siempre es la misma, indicando la detección de un cambio en el aparcamiento, emulando entradas o salidas, indistintamente. Además, dentro de estas periodicidades se engloban los mensajes de error, calibración, activación del dispositivo, entre otros.

La duración del funcionamiento del sistema es de 12 horas diarias, por ello la mayoría de las simulaciones utilizan esta duración. Dentro de esta duración se implementan los distintos escenarios, modelos, errores, entre otros.

4.3. Fase 3: Diseño e implementación la simulación

En este punto, el trabajo consiste en definir y desarrollar la implementación de los distintos escenarios de la simulación, teniendo en cuenta el máximo de factores técnicos y ambientales, que afecten a las comunicaciones y el dispositivo seleccionado. Con la intención de mostrar detenidamente estos pasos, esta fase se divide en dos subsecciones. Primero, la Subsección 4.3.1 muestra los algoritmos y características más relevantes para los distintos escenarios, argumentando las pruebas realizadas con tal de justificar los datos que se van a emplear. Mientras que en la Subsección 4.3.2 se detallan las diferentes secciones de los programas donde se han implementado dicho modelaje del escenario.

4.3.1. Modelaje de los escenarios base ns-3

Existen tres escenarios para esta simulación, aunque todos comparten entorno de aplicación y modelaje de algoritmos, la diferencia principal entre ellos es el comportamiento y la funcionalidad de LoRaWAN. A continuación, se empieza por analizar los diferentes modelos de propagación de la señal, su pérdida y los obstáculos.

Así pues, en primer lugar, argumentar las principales razones para la selección del modelo de propagación de la señal para LoRa/LoRaWAN. Para todos los escenarios se implementa el modelo llamado 3GPP, las razones para su elección abarcan motivos técnicos propios de dicho modelo, hasta su implementación y uso dentro de la herramienta de simulación ns-3.

Actualmente existen distintos modelos de propagación con distintos fines, genéricos, específicos como entornos rurales, urbanos, con presencia de obstáculos y demás. Según Stusek et al. (2020) son cinco los modelos que presentan aspectos cruciales para entornos urbanos y dispositivos que emplean comunicaciones LPWAN, estos se tratan del 3GPP Urban, SUI Model, Ericsson Urban, Okumura-Hata Urban, Cost 231. Todos ellos pueden ser usados para calcular propagaciones en distintos protocolos de comunicaciones IoT LPWAN, como NB-IoT, LoRaWAN y Sigfox, ya que todos estos modelos permiten un amplio rango de frecuencias, del entorno a los MHz hasta los GHz, dependiendo del modelo.

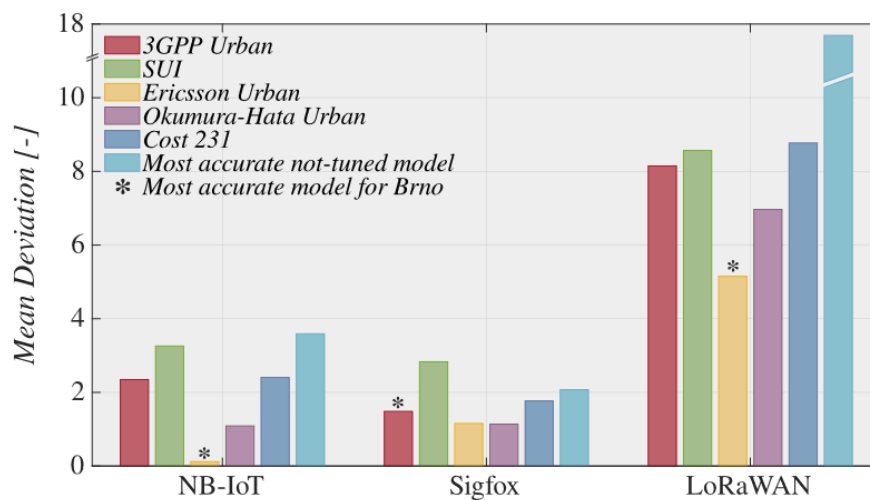
El trabajo realizado por Stusek et al. (2020) realiza una comparativa entre los modelos citados anteriormente en dos ciudades, Brno y Ostrava, ambas en Chequia. Allí se realiza la implantación de los distintos modelos de propagación empleando tres protocolos (NB-IoT, LoRaWAN y Sigfox), siendo estos contrastados con datos proporcionados por señales reales

tomadas en su estudio, finalizando, con una validación cruzada entre los conjuntos de datos de ambas ciudades.

Las conclusiones de este análisis son diversas, pero antes de indicar cuales son los mejores modelos para los escenarios y protocolos empleados, los autores ratifican la importancia y relevancia que tiene el ajuste para todos y cada uno de los modelos empleados. Estos, si se usan empleando fórmulas generalistas generan resultados con valores divergentes a los que reflejan la realidad. Según Stusek et al. (2020), ajustar el modelo al escenario y características de estos implica mejoras en torno al 40% hasta el 800%, dependiendo del modelo y protocolo.

Con todo, los mejores modelos resultantes se especifican según las ciudades y protocolos, luego de que estos fuesen ajustados y se haya realizado la validación cruzada entre Brno y Ostrava, estos pueden ser observados en la Figura 20.

Figura 20. Resultados de las desviaciones generadas por los modelos de propagación de la señal en la ciudad de Brno.



Fuente: Stusek et al. (2020).

Según los resultados obtenidos, el mejor modelo para LoRaWAN es el llamado Ericsson Urban con una desviación media muy inferior respecto al resto de los modelos. Por detrás le sigue el modelo Okumura-Hata, 3GPP y demás, en este sentido la lógica indica que se debe escoger el mejor modelo posible. Aun así, es indispensable revisar la implementación de dichos modelos dentro del programa del ns-3, en la Tabla 8 se especifican los que se encuentran actualmente implementados.

Tabla 8. Modelos de propagación de la señal nombrados por Stusek et al. (2020) e implementados en el ns-3.

Nombre	Uso	Características	Modelos
Cost 231	Urbano	Modelo Hata extendido a frecuencias de 1500 a 2000 MHz	LOS
Okumura-Hata	Generalista, urbano y suburbano	Modelo Hata para freq. de 150 a 1500 MHz. Implementa el Cost231 internamente	LOS
3GPP	Urbano, rural, entornos específicos	Modelo versátil implementado para distintos escenarios específicos. Frecuencias de 500 MHz a 5 GHz	RMA, UMa, UmiStreetCanyon, IndoorOffice y específico, V2V todos con LOS/NLOS/NLOSv incluidos

Fuente: Elaboración propia e información del nsnam, s. f.

Solo tres de los modelos se encuentran implementados en el ns-3, en este caso Ericsson Urban no se encuentra implementado, por ello es necesario analizar el modelo subsiguiente en la lista, el Okumura-Hata. Con este modelo se observan distintas coyunturas de la implementación que dan razones para no utilizar este modelo, la primera, es la implementación del modelo generalista para entornos urbanos sin ni siquiera realizar una comprobación del entorno, como los edificios. Además, lleva incorporado la elección del modelo Cost 231, debido a que este se trata de una extensión del modelo base (Hata), destacado solo por el distinto rango de frecuencias empleadas. Si el modelo Okumura-Hata contempla frecuencias entre los 150 y 1500 MHz, el modelo Cost 231 implementado en el simulador ns-3 solo sirve para comunicaciones que vayan de los 1500 a 2000 MHz, quedando anulado su caso para el uso de LoRaWAN.

En segundo lugar, el modelo Okumura-Hata se implementa de manera sencilla, como se ha comentado anteriormente, esto hace que no cuente con un sistema que visualice si los sensores tienen visión directa o no con respecto a la pasarela, lo que se llama línea de visión (LOS) o sin línea de visión (NLOS). Esto deja en suspensión la elección de este modelo y empezar a revisar el tercero en la lista y determinar si se trata de un mejor modelo.

En cuanto al modelo 3GPP implementado en el ns-3, este presenta una mayor pluralidad de escenarios y características que pueden ser utilizadas. Existen distintas modalidades que parten de la raíz básica del modelo para adaptarse de manera concreta a escenarios rurales (RMA), urbanos (UMa) y situaciones concretas, como una calle estrecha (UmiStreetCanyon) o despliegues en oficinas (IndoorOffice). Sin embargo, existe uno aún más específico, con posibilidad de implementación para despliegues en entornos urbanos y con presencia de vehículos, el llamado V2V (*Vehicle to Vehicle*).

Este modelo es capaz de ajustar los algoritmos 3GPP estándares y tener en cuenta despliegues en situaciones donde los dispositivos pueden ser móviles o estacionarios, en ámbitos urbanos, con detección de línea de visión directa, controlando las rutas de transmisión según los edificios generados (LOS o NLOS) e incluso si este ha sido provocado por vehículos (NLOSv). Aunque este último aún no se ha implementado en el programa ns-3 y se espera que esté disponible para versiones futuras. Además, permite determinar si cada uno de los dispositivos (sensores o pasarelas) se encuentran dentro o fuera de edificios, aplicando las atenuaciones correspondientes.

El modelo 3GPP V2V tiene opciones específicas para terminar de ajustar su modelo, como la frecuencia de uso, el escenario, puramente Urbano o con autopistas adyacentes, la implementación de modelo "Shadowing", permitiendo añadir fluctuaciones debidas a los obstáculos y la generación de pérdidas debidas a edificios, como se manifiesta en el párrafo anterior.

Los algoritmos implementados dentro del ns-3 para el modelo 3GPP V2V se implementan a raíz del trabajo Boban et al. (2016), estos se encuentran representados en las siguientes fórmulas, según el estándar ATIS, s. f., en concreto ATIS.3GPP.37.885.V1530.

Para el modelo de computación LOS se emplea la fórmula establecida en la Tabla 6.2.1-1:

$$PL = 38.77 + 16.70 \log_{10}(d_{3D}) + 18.20 \log_{10}(fc) \quad (1)$$

El modelo de pérdida de propagación para el caso NLOS, de la misma tabla:

$$PL = 36.85 + 30.00 \log_{10}(d_{3D}) + 18.90 \log_{10}(fc) \quad (2)$$

Donde la “ d_{3D} ” representa la distancia Euclídea entre dispositivos (sensoriales y pasarelas), emisor (Tx) y receptor (Rx), en un espacio 3D y en metros. En cuanto a la “ fc ”, esta representa la frecuencia central de la banda por la que se está transmitiendo en GHz. Asimismo, a estos modelos de pérdida de propagación de la señal se les asigna una constante, llamada “Shadow Fading”, según LOS o NLOS, de $\sigma_{SF} = 3$ y $\sigma_{SF} = 4$ respectivamente según la Tabla 6.2.1-1 del documento ATIS nombrado, para el posible empleo del algoritmo “Shadowing”. En esta línea la fórmula implementada para emular dicho algoritmo se representa por el valor del “Shadowing” (SV).

$$SV = R(\Delta d_{2D}) \cdot SV_{prev} \cdot \left(\sqrt{1 - R^2(\Delta d_{2D})} \right) \cdot \sigma_{SF} Z, \quad Z \sim N(0, 1) \quad (3)$$

Donde $R(\Delta d_{2D})$ es el coeficiente de descorrelación entre la distancia de los dispositivos de la iteración previa.

El algoritmo “Shadowing” añade otra pérdida de la señal después de aplicarse el modelo de pérdida debido a la propagación de 3GPP V2V descrito en la documentación del ns-3, en concreto Propagation — Model Library, s. f..

El modelo siempre aplica una pérdida fija según la constante “Shadow Fading” por dispositivo, por este motivo, se le añade una variabilidad según un generador de variables aleatorias llamado transformada de Box-Muller, multiplicando este efecto.

El modelo Box-Muller es un método de generación de pares aleatorios de números independientes, junto a una distribución normal, “estándar”, con esperanza de 0 y varianza unitaria. Su implementación se rige por la forma básica para su cálculo, representado en el par de fórmulas descritas a continuación.

$$Z_0 = R \cos(\Theta) = \sqrt{-2 \ln U_1} \cdot \cos(2\pi U_2) \quad (4)$$

$$Z_1 = R \sin(\Theta) = \sqrt{-2 \ln U_1} \cdot \sin(2\pi U_2) \quad (5)$$

Donde las variables U_1 y U_2 son muestras independientes escogidas de las fuentes de datos normalizados del modelo, Z_0 and Z_1 representan las variables aleatorias.

En aspectos generales, no se ha observado que ninguno de los modelos implementados contemple la posibilidad de encuentros con interferencias derivadas de los rebotes generados por las mismas emisiones de los dispositivos y los obstáculos. Aunque, para este trabajo, al no haber encontrado ningún modelo ya implementado dentro del ns-3, se considera que el modelo “Shadowing” ayuda a intuir mejores resultados y es más que suficiente.

El mismo método de generación aleatoria (Box-Muller) se aplica para el modelo de propagación de la señal principal de 3GPP V2V cuando se quiera añadir una pérdida de señal provocada por el coche. Las razones para añadir dicho algoritmo se basan en dos argumentos complementarios entre sí, la primera es la implementación incompleta del modelo 3GPP V2V, como se ha argumentado anteriormente su desarrollo es práctico pero ciertas funcionalidades aún no se encuentran implementadas. Entre estas, se encuentra el modelo NLOSv (*Non Line of Sight vehicle*), por lo que no se pueden simular escenarios aleatorios donde se añada la pérdida provocada por vehículos.

En segundo lugar, existen trabajos y análisis en la literatura que muestran datos del impacto resultante de dicha atenuación de la señal ocasionado por los vehículos. Por ello, en el trabajo realizado por Turner et al. (2021) se realizan distintos escenarios donde calculan la pérdida de la señal generada por distintos vehículos (coches y autobuses) situados entre un sensor y la pasarela, situados a los laterales de dichos vehículos. Estos resultados son comparados con modelos de propagación de la señal como LOG-Distance o Two-Ray, entre otros, para encontrar el modelo que se adapte mejor a la señal.

En sus resultados se muestran y comparan los distintos errores cuadráticos medios (RSME) generados entre los distintos modelos y los escenarios con respecto a los datos reales obtenidos. Con ellos, en las conclusiones, Turner et al. (2021) corrobora que de todos los modelos implementados el Two-Ray tiene una mayor fidelidad y argumenta que en la literatura existe validación cruzada entre este y otros modelos más nuevos y mejorados como

“Free Space Loss” (Friis) o CORNER para entornos con obstrucción de vehículos y modelos V2V, como es el caso. Tal premisa permite visualizar un primer rango de valores con respecto a la atenuación y añadir otra razón para emplear el modelo V2V de 3GPP en el algoritmo del ns-3.

Aún con los resultados obtenidos de la literatura, estos no pueden ser empleados directamente en la simulación, antes deben ser contrastados según la tecnología y escenario lo más fiel posible. Es por ello por lo que se realizan unas pruebas físicas con el sensor de Winext y la tecnología LoRa/LoraWAN para terminar de ajustar el valor que se emplea en la simulación para la atenuación del coche ficticio.

El escenario para la realización de la prueba de atenuación emplea el sensor Winext instalado en una plaza de aparcamiento, situado en la universidad de Girona, donde se recogen los datos de distintos aparcamientos (entradas y salidas) realizados por tres modelos de coche los cuales se tratan de un Mazda 3, un Toyota Avensis y un Suzuki Jinmy. El sensor se sitúa en una posición donde queda directamente oculto por el bloque motor del coche, considerado el peor escenario por añadir la mayor distorsión y pérdida a la señal. Aun así, para contrastar los datos, se realiza la misma prueba con uno de los coches (el Mazda 3) realizando aparcamientos de forma contraria, el área de detección del sensor se trata del maletero de dicho vehículo.

Las pruebas se realizan capturando los quince primeros eventos de cada uno de los escenarios descritos anteriormente. Los datos obtenidos se representan en la Tabla 9, donde se observan las desviaciones típicas (estándar) según el indicador de fuerza de la señal recibida (RSSI), en dBm, según la entrada y salida del vehículo, de estos se pueden extraer distintas conclusiones.

La primera y más destacada se trata de la importancia que tiene la altura del vehículo, esto añade una gran variabilidad en la pérdida de la señal ya que cada uno de los coches es más alto que el anterior, empezando por el Mazda 3, el Toyota Avensis hasta el Suzuki Jinmy. Otras características como la longitud y la anchura de estos no se han considerado para esta prueba.

En segundo lugar, se puede apreciar la poca variación de los resultados según la entrada o salida del vehículo, esto puede ser originado por un posible comportamiento intrínseco en el sensor escogido, el cual se relaciona en la forma que tiene de detectar cambios en su entorno empleando el sensor magnético. Dicho razonamiento se sustenta en la existencia de los “Interrupts” y la funcionalidad del magnetómetro. Las interrupciones en los

microcontroladores pueden ser empleadas para despertar la unidad central en cuanto el sensor detecta un cambio sustancial en su entorno, como el cambio en el campo magnético, siempre dentro de unos márgenes y respetando los ciclos mínimos de “Deep Sleep”.

Tabla 9. Datos de la desviación estándar según el RSSI extraído de las pruebas realizadas con LoRa y el sensor.

Detección	Mazda 3 (dBm)	Toyota Avensis (dBm)	Suzuki Jimny (dBm)	Mazda 3 opuesto (dBm)
Entrada	12.59	8.79	1.07	11.45
Salida	11.83	7.66	3.36	3.98

Elaboración: Propia.

En tercer lugar, se encuentra la diferencia de los datos según el comportamiento generado en la entrada o salida de un vehículo con respecto a la detección del bloque motor o el maletero. Este hecho tiende a otorgar una mayor importancia en el comportamiento del dispositivo y la relación que existe en tener una mayor sensibilidad cuando se trata del bloque motor. En este sentido se han detectado emisiones “Uplink” inmediatamente cuando el vehículo realiza una entrada con el bloque motor y un mayor tiempo con respecto al maletero.

Además, se han detectado comportamientos que indican que el hecho de aparcar o salir del aparcamiento con mayor rapidez con el coche genera una mayor variabilidad de RSSI entre “Uplinks”, obteniendo menor atenuación a la salida y mayor a la entrada.

Según las conclusiones y datos resultantes extraídos por esta prueba, si se comparan con las del trabajo realizado por Turner et al. (2021) se puede ultimar que en esta ocasión el uso de datos fieles al proyecto a desarrollar son de mayor utilidad. Es por ello por lo que se emplea una media entre los distintos valores de atenuación captados, resultando con un valor igual a 8.2338 dBm, convirtiéndose como el valor estándar a aplicar al método de generación aleatoria de Box-Muller para que este sea añadido al cálculo de la pérdida de la señal del modelo 3GPP V3V, emulando un coche.

Resuelto el diseño de la atenuación del coche, el siguiente punto a abordar es la representación de los edificios en el escenario simulado, hasta donde ha observado el autor

el ns-3 no contempla la inserción de mapas de edificios, aun con esto el programa ns-3 contiene una implementación que genera un mapa aleatorio de edificios según un repertorio de configuraciones. Esta implementación crea una red de edificios que contemplan su posición en los tres ejes de la simulación, su altura, número de pisos, tamaño de las habitaciones y, finalmente, el espacio de las calles colindantes.

La obtención de los datos para los edificios se realiza según distintas referencias catastrales (*Sede Electrónica del Catastro, s. f.*), complementándose a los datos del terreno extraídos en la Fase 1 en la Sección 4.1. Por ello, como se ha mencionado en secciones anteriores, se considera que el terreno en el lado Oeste de plaza Catalunya se genera sin elevación, luego, para la elevación del lado Este se simplifica. Esta elevación del terreno se considera como un triángulo rectángulo, con la elevación de la plaza Catalunya como punto de referencia hasta la última línea de edificios que se contempla una distancia de 380 metros, 580 metros de separación entre el primer sensor instalado y la pasarela situada en la torre Alfons XII menos 200 metros entre la pasarela y la última línea de edificios, a 90 metros de altura, aplicada la correspondiente sustracción de la altura de plaza Catalunya, con ello se calcula una pendiente de 0.2368.

Además, se tiene en cuenta el entorno de la ciudad, con ello es indispensable considerar la línea que marca el río. Así pues, el modelo a implementar consiste en tres zonas, el lado Oeste, sin inclinación y lleno de edificios, la zona central, el despliegue sensorial en plaza Catalunya, sin obstáculos ni inclinación y la zona del lado Oeste donde se contempla la zona urbana correspondiente con la inclinación descrita en el párrafo anterior.

Para finalizar, el tercer escenario de este proyecto debe emplear un modelo de consumo de energía que tenga en cuenta el máximo de características intrínsecas de los sensores y entorno, es por ello por lo que emplean los mismos modelos de propagación y pérdida de la señal que en los dos primeros escenarios, añadiendo el comportamiento del consumo.

En el simulador ns-3, existen dos algoritmos que permiten diseñar e implementar dicho modelo para las comunicaciones LoRa/LoRaWAN, el primero es genérico dentro del entorno ns-3 y permite la creación de las baterías, el segundo es particular de las comunicaciones LoRa/LoRaWAN y permite implementar el modelo de consumo del dispositivo.

El generador de la batería del dispositivo permite parametrizar distintos factores, dentro del tercer escenario, existen dos parámetros indispensables, el voltaje del dispositivo (V_{cc} , Voltios) y la energía de la batería (Joules). Según los datos del sensor, estos son 3.30 V y, aunque no se sabe la carga en Coulomb, se obtienen los 16 000 mA·h.

Para determinar el valor de la energía de las baterías se emplean las ecuaciones de la energía y carga eléctricas. Dada la relación que existe con la energía (E), la cual es igual al producto de la carga eléctrica (Q) en Coulomb y el voltaje (V) al cual se ve sometido el dispositivo.

$$E = Q \cdot V \quad (6)$$

A la vez la carga eléctrica (Q) es el producto de la corriente (I) por tiempo (t).

$$Q = I \cdot t \quad (7)$$

Con dichas ecuaciones el primer paso es realizar la conversión entre la carga eléctrica, en Coulomb, con los mA·h, dando como resultado una relación de 3600 segundos por hora. Una vez con este valor y los 16 000 mA·h se sustituyen en la ecuación de la energía empleando el valor de voltaje de la carga, el cual se trata del voltaje de alimentación de la batería (3.60 V), dando como resultado final 207 360 J.

$$E = 16.00 \text{ A} \cdot \text{h} \cdot 3600.00 \frac{\text{s}}{\text{h}} \cdot 3.60 \text{ V} = 207\,360.00 \text{ J} \quad (8)$$

Esta misma ecuación se realiza para el voltaje mínimo de funcionamiento del dispositivo, reglado a 2.90 V y conlleva un límite en la energía de 167 040 J, en el cual deja de funcionar y se considera inoperativo.

Con los valores necesarios para la implementación del sistema de la batería ahora solo falta diseñar el segundo modelo, el que determina el consumo de los dispositivos LoRa/LoRaWAN. Por ello son imprescindibles toda una serie de requerimientos, existen cuatro requerimientos imprescindibles, algunos de ellos referentes al consumo del dispositivo en distintos modos de funcionamiento como el estado “Standby” y “Deep Sleep”, asimismo se piden los consumos del chip empleado para las comunicaciones durante la transmisión (Tx) y recepción (Rx). Todos los parámetros pueden ser rellenados con la información que se extrae en la primera fase de este proyecto, añadir que la tipología de la batería Li-SOCL2 puede ser empleada en este

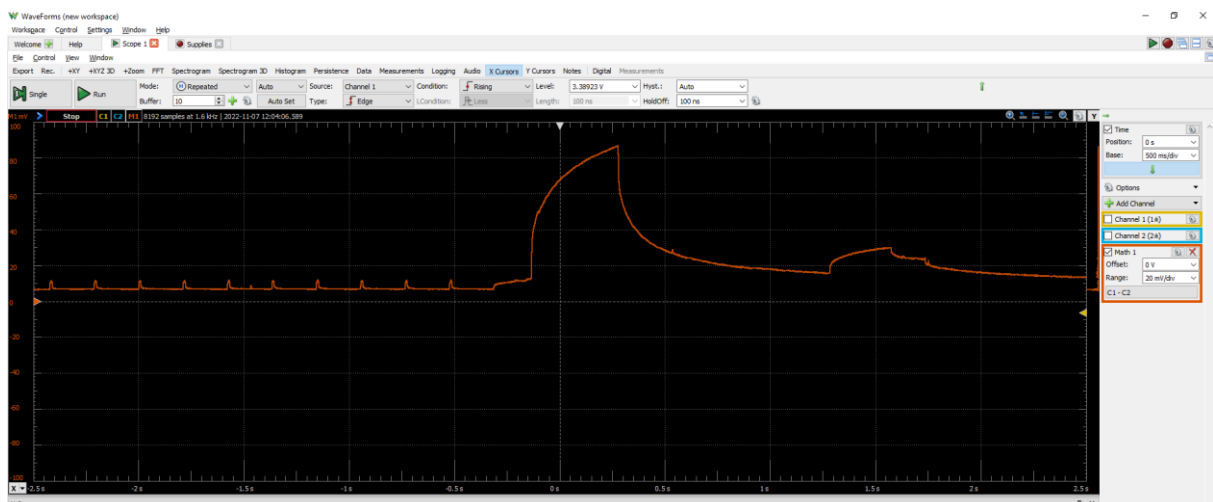
simulador y su consumo sigue el modelo implementado en el programa `li-ion-energysource.h`.

A razón de la disponibilidad de uso del dispositivo, se opta por obtener datos contrastados y más precisos de este, por ello se decide abrir el dispositivo Winext para monitorizar los consumos de los diferentes ciclos. Tal prueba consiste en la realización de un circuito electrónico para calcular la corriente empleando una resistencia de alta precisión y potencia (Shunt) conectada en serie a la alimentación del dispositivo.

Esto permite calcular la corriente empleando la ley de Ohm, computando la caída de tensión de la resistencia a corriente, en la Figura 21 se muestra una captura del comportamiento del sensor al detectar la salida de un obstáculo, con el SF10, "Payload" de 11 bytes y Tx a 14 dBm, siendo el más importante este último, y recibiendo un ACK en la primera ventana (Rx1). Para capturar los datos se emplea el programa, Waveforms de Digilent, junto al dispositivo "Analog Discovery 2".

El dispositivo empleado para capturar los resultados capta un voltaje mínimo proveniente del dispositivo, muy probablemente provocado por el condensador conectado a la entrada del dispositivo. Este hecho impide determinar con exactitud el consumo en "Deep Sleep" por lo que a este resultado no se aplica ningún cambio. Aun con ello, se han podido medir los consumos para el resto de los comportamientos, mostrándolos de forma comparativa en la Tabla 10.

Figura 21. Captura del consumo del dispositivo sensorial, envío más recepción ACK en Rx1.



Fuente: Elaboración propia.

Al realizar la toma de los datos del consumo de esta forma permite entender el comportamiento del sensor, en especial, el ciclo de la detección más el envío de la señal por LoRa. En la captura de la Figura 21 se aprecian todos los eventos del dispositivo que describen el comportamiento descrito, en primer lugar, se encuentran diez picos pequeños, estos representan la toma de datos del magnetómetro. La cuantía y periodicidad en estos picos indica un cambio en el entorno del dispositivo y da paso al procesado y retransmisión de la información, respetando siempre los tiempos de apagado y encendido. El mismo comportamiento se perpetúa en el tiempo, pero con periodicidades más largas durante la fase "Deep Sleep", otorgando una mayor rapidez en la detección.

En la segunda fase se activa el microcontrolador y procesa todas las métricas para ser emitidas por LoRa, representado por el pico previo a la aleta de tiburón. A continuación, se procede con la transmisión de los datos, periodo que depende de la modulación de la señal (SF) o carga ("Payload"), entre otros. Justo después de un segundo, se observa la apertura de la primera ventana de recepción (Rx1), último pico, durante un periodo de tiempo más largo de lo normal debido a la recepción del ACK por parte de la gateway.

En cuanto a la aleta de tiburón registrada durante la transmisión (Tx), se interpreta que es resultante de un condensador electrolítico conectado a la alimentación del dispositivo ya que se trata de una capacitancia muy elevada, de 27 F a 3.80 V. A falta de los esquemáticos se decide no quitar dicho condensador y continuar con el muestreo de los datos.

Para el parámetro Tx no se define como el definitivo ya que el resultado con la mejor cobertura y menos error de transmisión de paquetes extraído en las dos primeras fases es el empleado para el tercer escenario.

Con los valores actualizados, se contemplan variaciones importantes indicando una diferencia sustancial entre el comportamiento especificado en la documentación y la obtenida. Así pues, se emplean los datos obtenidos por el experimento siempre que no se indique lo contrario.

Como último apunte, los periodos de tiempo que se contemplan para las ventanas de recepción de los datos (Rx1 y Rx2) se establecen en el mínimo permitido, tratándose de periodos de un segundo entre eventos.

Tabla 10. Comparativa de los consumos para la implementación del modelo de energía de las comunicaciones.

Nombre	Teórico (mA)	Práctico (mA)
Corriente del dispositivo o "Standby Current"	120.000	15.00
Corriente durante el estado "Deep Sleep"	0.015	--
Corriente del chip para la emisión de la señal (Tx)	15.000	90.00
Corriente del chip para la recepción de la señal (Rx)	11.200	25.00 sin ACK 32.00 con ACK

Fuente: Elaboración propia.

4.3.2. Implementación para los distintos escenarios

En primer lugar es imprescindible hacer mención del módulo por el cual es posible realizar dicha simulación gracias al llamado *LoRaWAN ns-3 module (2017/2023)* desarrollado por Magrin et al. (2017) y Capuzzo et al. (2018) en sus trabajos correspondientes. Toda implementación se realiza con base a los programas realizados por estos autores y se adaptan a los escenarios pertinentes, dicho esto, la implementación siguiendo el mismo orden que se ha empleado en la subsección anterior.

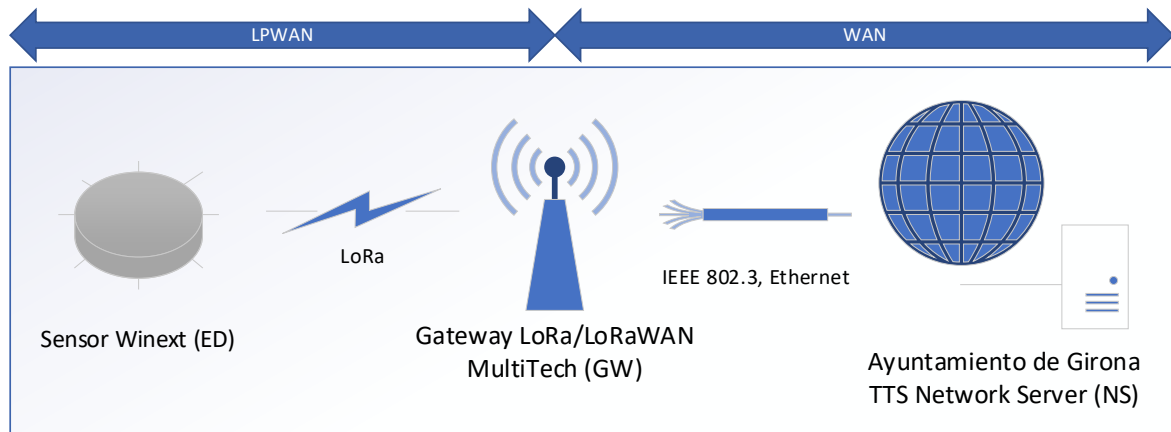
Es por ello, por lo que se parte del programa llamado `complete-network-example.cc` y se adapta para la casuística necesaria para este proyecto, aun así, se mantiene la topología tipo árbol que se implementa, esta se representa la Figura 22.

Todos los escenarios crean 35 dispositivos que, por orden enumerativo, representan los 33 sensores, la pasarela y el servidor (*Network Server*). Todos estos dispositivos incorporan las características y emulan el comportamiento de dicho elemento por el cual ha sido asignado.

En el caso particular de las comunicaciones LoRa/LoRaWAN y los dispositivos que emplean estas comunicaciones (ED y GW) deben tener en consideración en todo momento ciertas

métricas, algunas propias del chip de comunicaciones, como la sensibilidad y otras de legislativas, como el límite del DC (*Duty Cycle*), establecido al 1%, estos quedan representados en la Tabla 19 del anexo B.

Figura 22. *Arquitectura de gestión y conectividad de borde planteada para la simulación.*



Fuente: Elaboración propia.

Estas regulaciones, creación de dispositivos y demás se estipulan al inicio del programa en la forma que se observa en el siguiente extracto de código. Todos ellos son de fácil interpretación, aun así, el “packet_size” se debe mencionar que se debe tener en cuenta solo la carga de datos que envía el sensor, no se debe tener en consideración el “Header”, por defecto, este emplea el tamaño estándar y se añade una vez se inicia la simulación.

Listado 1. *Código de configuración de la simulación.*

```
// Network settingsheader
int nDevices = 33;           // Num. ED
int nGateways = 1;          // Num. GW
double radius = 15000;      // CObertura teórica de la gateway
double simulationTime = 43200; // 43200 = 12 h // 86400 segundos = 1 día

int packet_size = 11; // Sólo Payload, sin Header de LoRaWAN

// Modelo del canal
bool realisticChannelModel = true;

int appPeriodSeconds = 20; // Periodicidad del ED --> 20 / 180 / 360 / 540 /
720 / 900
```

A partir de este punto empieza la realización de la implementación del modelo de pérdida, por ello se crea el modelo del canal con las atribuciones de la propagación 3GPP V2V. En este indica la frecuencia de transmisión, la cual al tratarse de la banda ISM de EU, se estipula en

868 MHz, se activa la implementación “Shadowing” junto al algoritmo de pérdidas debido a los edificios.

Listado 2. Código de elaboración de la configuración para el modelo de propagación de la señal.

```

ObjectFactory propagationLossModelFactory =
    ObjectFactory("ns3::ThreeGppV2vUrbanPropagationLossModel");
Ptr<ThreeGppV2vUrbanPropagationLossModel> m_lossModel =
    propagationLossModelFactory.Create<ThreeGppV2vUrbanPropagationLossModel>();

Ptr<ChannelConditionModel> losCondModel =
CreateObject<BuildingsChannelConditionModel>();
m_lossModel->SetAttribute("ShadowingEnabled", BooleanValue(true));
m_lossModel->SetAttribute("Frequency", DoubleValue(868.0e6));
m_lossModel->SetAttribute("BuildingPenetrationLossesEnabled",
BooleanValue(true));
m_lossModel->SetAttribute("ChannelConditionModel",
PointerValue(losCondModel));

```

Como se estipula en el diseño del modelo 3GPP a este se le añade la aleatoriedad de atenuación provocada por el vehículo. A continuación, se muestra el código de implementación que afecta a la función “DoCalcRxPower” de la librería three-gpp-propagation-loss-model.cc, con la sección anulada hasta su empleo.

Listado 3. Extracto de código de la pérdida de la potencia de la señal.

```

rxPow = rxPow - GetLoss(cond, distance2d, distance3d, heights.first,
heights.second);
// - abs(m_normRandomVariable->GetValue()) * 8.2338);

```

Igual de importante es la configuración del comportamiento y parámetros de LoRaWAN para el servidor (Network Server), la gateway y el Sensor. Por ello en el siguiente código se observa esta configuración, la cual varía según escenario. El parámetro “MaxTransmissions” se emplean solo tres, ya que el dispositivo Winext solo emite dos mensajes extras si el primer ciclo ACK falla por lo que el servidor y la gateway deben tener esto en cuenta.

Listado 4. Código de configuración del ED dentro de la simulación.

```

// Modo ED para que el NS actue en consideración
bool adrEnabled = true;
std::string adrType = "ns3::AdrComponent";
Config::SetDefault("ns3::EndDeviceLorawanMac::DRControl",
BooleanValue(true));

```

```

Config::SetDefault("ns3::EndDeviceLorawanMac::DataRate",
UIntegerValue(5));
Config::SetDefault("ns3::EndDeviceLorawanMac::MType",
                    EnumValue(LorawanMacHeader::CONFIRMED_DATA_UP));
// El ED de Winext solo permite 2 intentos extra para el ACK, 3 porque se
// empieza por el primero
Config::SetDefault("ns3::EndDeviceLorawanMac::MaxTransmissions",
IntegerValue(3));

```

En este mismo sentido, este paquete de simulación LoRa/LoRaWAN solo contempla el uso de los tres canales frecuenciales básicos de LoRa. La gateway contempla ocho canales y es por este motivo por lo que se debe implementar una modificación de la `lorawan-mac-helper.cc` y añadir los cinco canales restantes que permite la Gateway. En la función “ApplyCommonEUConfigurations” se implementan las propiedades de estos canales, recogidos en la Tabla 20, con el siguiente código se amplía el rango de los canales y especifica su “Duty Cycle” y Tx máximos.

Listado 5. *Creación de los canales para LoRa/LoRaWAN.*

```

LogicalLoraChannelHelper channelHelper;
// Añadido RC
channelHelper.AddSubBand(867, 868.6, 0.01, 14);

```

Seguidamente se diseñan las tres zonas referentes al escenario (Oeste, centro y Este), por ello es imprescindible modificar el programa estándar de la librería “buildings”, en concreto el programa llamado `building-allocator.cc`. A continuación, se muestra la función “GridBuildingAllocator” con los datos modificados para que respete las dimensiones correspondientes.

Listado 6. *Código de creación del escenario.*

```

BuildingContainer
GridBuildingAllocator::Create(uint32_t n) const
{
    NS_LOG_FUNCTION(this);
    PushAttributes();
    BuildingContainer bc;
    uint32_t limit = n + m_current;
    uint8_t m_plaza = 0; // Añadido RC
    float_t boundaries = 0;
    for (; m_current < limit; ++m_current)
    {
        // Añadido RC deja un margen proporcional al Delta_X empleado
        // Primero se añade esto para sumar el m_current del position-
        allocator
    }
}

```



```

Vector lowerLeft = m_lowerLeftPositionAllocator->GetNext();
Vector upperRight = m_upperRightPositionAllocator->GetNext();

if (lowerLeft.x < -30)
{
    // Vector lowerLeft = m_lowerLeftPositionAllocator->GetNext();
    // Vector upperRight = m_upperRightPositionAllocator->GetNext();
    Box box(lowerLeft.x, upperRight.x, lowerLeft.y, upperRight.y, 0,
m_height);
    NS_LOG_LOGIC("new building : " << box);
    BoxValue boxValue(box);
    m_buildingFactory.Set("Boundaries", boxValue);
    Ptr<Building> b = m_buildingFactory.Create<Building>();
    bc.Add(b);
}
else if (!m_plaza)
{
    ++m_plaza;
    Box box(-30, 50, -200, 200, -1, 0);
    NS_LOG_LOGIC("new building : " << box);
    BoxValue boxValue(box);
    m_buildingFactory.Set("Boundaries", boxValue);
    Ptr<Building> b = m_buildingFactory.Create<Building>();
    bc.Add(b);
}
else if (lowerLeft.x >= 50 && lowerLeft.x < 380)
{
    boundaries = lowerLeft.x * 0.2368;
    Box box(lowerLeft.x,
            upperRight.x,
            lowerLeft.y,
            upperRight.y,
            boundaries,
            m_height + boundaries);
    NS_LOG_LOGIC("new building : " << box);
    BoxValue boxValue(box);
    m_buildingFactory.Set("Boundaries", boxValue);
    Ptr<Building> b = m_buildingFactory.Create<Building>();
    bc.Add(b);
}
}
return bc;
}

```

Luego de realizar la modificación para la matriz de los edificios se aplican los detalles de dichos edificios que se crean en el programa principal de la simulación, en el siguiente código se pueden observar de cuales se tratan. Sus valores representan la longitud del edificio (Length) y sus habitaciones interiores (Delta) en sus consecuentes ejes, su altura, todos ellos en metros,

luego la cantidad de edificios en las dos direcciones de la matriz, finalizando con el punto referencial donde se quiere empezar a crear los edificios.

Listado 7. *Distribución de los edificios dentro de la simulación.*

```

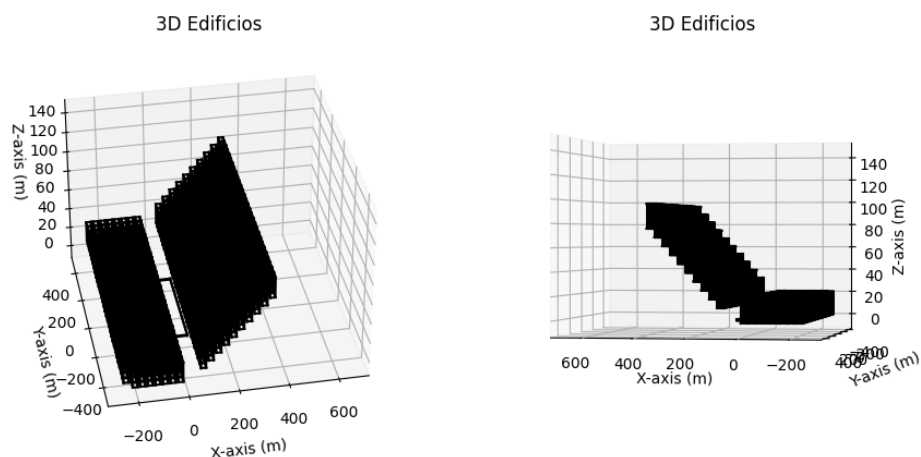
/*****
 * Handle buildings *
 *****/
double xLength = 20;
double deltaX = 10;
double yLength = 30;
double deltaY = 5;
double zHeight = 20;
int gridWidth = 30;
int gridHeight = 20;
double offX = -250; // Oeste-Este
double offY = -500; // Norte-Sur

```

Para visualizar que el modelo de la ciudad queda representado según el diseño estipulado se crea un pequeño *script* de Python para recrear la visualización de dicho escenario gracias a los datos recogidos durante la creación de la simulación. Dicha representación se puede visualizar en la Figura 23.

Finalmente, el modelo de consumo para el tercer escenario se aplican los valores descritos en su diseño en el programa principal, en concreto en el código que se puede observar a continuación.

Figura 23. *Visualización de los edificios creados.*



Fuente: Elaboración propia.

Por último, con referencia al comportamiento de los dispositivos sensoriales respecto a las ventanas de recepción, estos se encuentran en la librería llamada `class-a-end-device-lorawan-mac.cc`, específicamente en la función “`ClassAEndDeviceLorawanMac`”, e indican el tiempo transcurrido después de la transmisión “Tx”.

Listado 8. *Configuración de los consumos en el tercer escenario.*

```
// Configuración del modelo de energía
// Batería
basicSourceHelper.Set("BasicEnergySourceInitialEnergyJ", DoubleValue(207360)); // Energy in J
basicSourceHelper.Set("BasicEnergySupplyVoltageV", DoubleValue(3.3));

// Energía para el ED
radioEnergyHelper.Set("StandbyCurrentA", DoubleValue(0.006));
radioEnergyHelper.Set("TxCurrentA", DoubleValue(0.068));
radioEnergyHelper.Set("SleepCurrentA", DoubleValue(0.000015));
radioEnergyHelper.Set("RxCurrentA", DoubleValue(0.01075));

// Modelo de consumo fijo, desactivar si existe ADR activado
radioEnergyHelper.SetTxCurrentModel("ns3::ConstantLoraTxCurrentModel",
                                     "TxCurrent", DoubleValue(0.068));
```

Todos los escenarios contemplados en las simulaciones llevan estos parámetros, afectando a los protocolos de ACK, ADR y modelo de consumo implementado en el tercer escenario.

Listado 9. *Ventanas de tiempo para la recepción de mensajes.*

```
// LoraWAN default
m_receiveDelay1(Seconds(1)),
// LoraWAN default
m_receiveDelay2(Seconds(2)),
m_rx1DrOffset(0)
```

4.4. Fase 4: Análisis de los datos

En esta fase se termina el trabajo realizado hasta la fecha analizando los resultados encontrados en las simulaciones de la Fase 3, estos se reparten por escenarios por orden enumerativo.

En primer lugar, se necesita crear y emplear un *script* de Python para poder digerir la información, en la Subsección 4.4.1 se desarrolla este paso. A continuación, la Subsección 4.4.2 se argumentan el producto de la simulación para el primer escenario y se considera el mejor modelo, se tienen en cuenta las características intrínsecas del terreno y los sensores sin

la activación del ACK o ADR. En la Subsección 4.4.3 analiza el modelo para el mismo escenario anterior, pero activando las capacidades ADR u ACK. La Subsección 4.4.4 compara y presenta el mejor modelo de los escenarios anteriores y lo analiza para evaluar su aplicación en el tercer escenario. Finalizando con la Subsección 4.4.5, se muestran los efectos del modelo sobre el consumo del módulo de comunicaciones LoRa y por consiguiente la vida útil del sensor.

Para el análisis de esta fase es importante determinar la jerarquía de prioridades para establecer límites y poder discernir aquellos casos semejantes. Es por ello por lo que se establece como primera prioridad la fiabilidad, aquellos escenarios donde el modelo resulte con una tasa de errores (*Packet Error Rate* o PER) menor se antepone. En segundo lugar, la Cobertura, los escenarios que presenten mayores resultados en la señal recibida (Rx y SNR) son percibidos como prioritarios. Finalmente, el Consumo, en medida de lo posible se debe tener presente la posibilidad de aumentar la vida útil del dispositivo sin afectar gravemente el orden jerárquico.

Con el marco de subordinación establecido, los análisis mostrados en las siguientes subsecciones encarnan el producto final como resultado de diversas iteraciones por los distintos modelos y conjuntos de datos extraídos de las simulaciones.

Para finalizar, es necesario argumentar que todos los datos que se exponen en el análisis del primer escenario y de los subsiguientes para el periodo de 20 segundos, relacionados con el SF9 hasta el SF12 representan comportamientos que evocan a la saturación de los canales de las comunicaciones LoRa para los ED. Este comportamiento es debido a que los ciclos de retransmisión de eventos son mayores al tiempo necesario que debe transcurrir entre eventos para un mismo canal, creando un balance negativo que, de manera progresiva, va llegando al DC máximo permitido, activando el retardo de transmisión por canal ocupado y repercutiendo a la cantidad total de eventos posibles.

Para que el funcionamiento sea el correcto, el periodo empleado en las simulaciones tiene que ser igual o superior a los establecidos en la Tabla 19 de los anexos, específicamente en la columna llamada "1% (s)" se establecen los tiempos que deben transcurrir entre eventos. No obstante, se decide emplear estos datos con la finalidad de poder emplear las tendencias de dichos SF, DC, número de eventos y otros para alertar de que este factor se debe tener en consideración y, en caso de que se vayan a emplear, ya que pueden llevar a una saturación

repentina de todos los canales si el volumen de tráfico es suficientemente elevado y sostenido en el tiempo.

4.4.1. Creación del digester de datos de Python

El análisis de los distintos escenarios comporta recoger toda una lista de factores, métricas y características por cada transmisión, además de los que pertenecen a la configuración del emplazamiento, sensor, pasarela y otros. En este sentido, el simulador genera archivos que recogen los datos de transmisión y eventos que suceden dentro de la simulación. Dicho documento resultante es una aglomeración de mensajes (Logs) que contienen la información resultante de la simulación, muy útiles para la comprensión humana pero que se encuentran en un formato desordenado y no apto para realizar tareas de análisis.

Python permite implementar técnicas y algoritmos de búsqueda de información (*"Data scraping"*), por ello se definen palabras clave dentro del archivo generado que permiten localizar los datos, estas se exponen en el siguiente extracto de código. Cabe mencionar que el código desarrollado permite extraer la información con independencia del escenario generado, ya que este es capaz de discernir según los datos que ha encontrado dentro del archivo ingerido.

Listado 10. *Definición de las palabras clave para la búsqueda de información del script de Python.*

```
# Definir las palabras claves de búsqueda.  
keywords = [  
    "PacketToSend:", "Propagation:", "dstNode = ", "Received packet:",  
    "ACK packet",  
]
```

Este programa se separa en tres fases, la ingestión del archivo, la selección de los datos y la generación del archivo final. Así pues, el primer paso se trata del más duradero ya que necesita inspeccionar la totalidad del archivo, para quedarse con la línea de datos donde aparezca la palabra clave del listado.

En una segunda fase, se realiza otra búsqueda de información, pero esta vez se realiza solo con la lista generada por el primer algoritmo siendo esta más precisa, buscando la posición exacta de todos los datos. También se emplean condiciones para distinguir que algoritmos seguir según la palabra clave a la que pertenece esa línea de texto, de esta manera se van guardando los datos en los *arrays* correspondientes (ED, Tx, Rx, Time, entre otros).

Algunos de los datos no se encuentran en el archivo, sino que se generan según se analizan los datos como es el caso del SNR y su posición geoespacial, su Longitud y Latitud. Para el SNR se emplea la ecuación como:

$$\text{SNR} = \text{Rx} + 174 \text{ dBm} - 10 \cdot \log_{10}(\text{BW}) - \text{NF} \quad (9)$$

Donde el Rx, es la potencia de la señal recibida en dBm, la BW representa el ancho de banda en Hertz, siendo 125 000 Hz y el NF representando la figura del ruido (“Noise Figure”), 6 según el SX1276. En el siguiente extracto de código se observa como el resultado de la ecuación se añade a una lista con relación al SNR de la transmisión captada por el programa.

Listado 11. *Algoritmo de computación del SNR.*

```
Signal_t_Noise_R.append(float(Rx[-1] + 174 - 10 * math.log10(125000) - 6)
```

Finalmente, gracias a la librería “Pandas” de Python se emplea la función “DataFrame” para generar un archivo Excel que almacene los datos de todos los Array descritos en la Tabla 22 de los anexos, su código queda representado de la siguiente forma.

Listado 12. *Etiquetas de estructuración de los datos.*

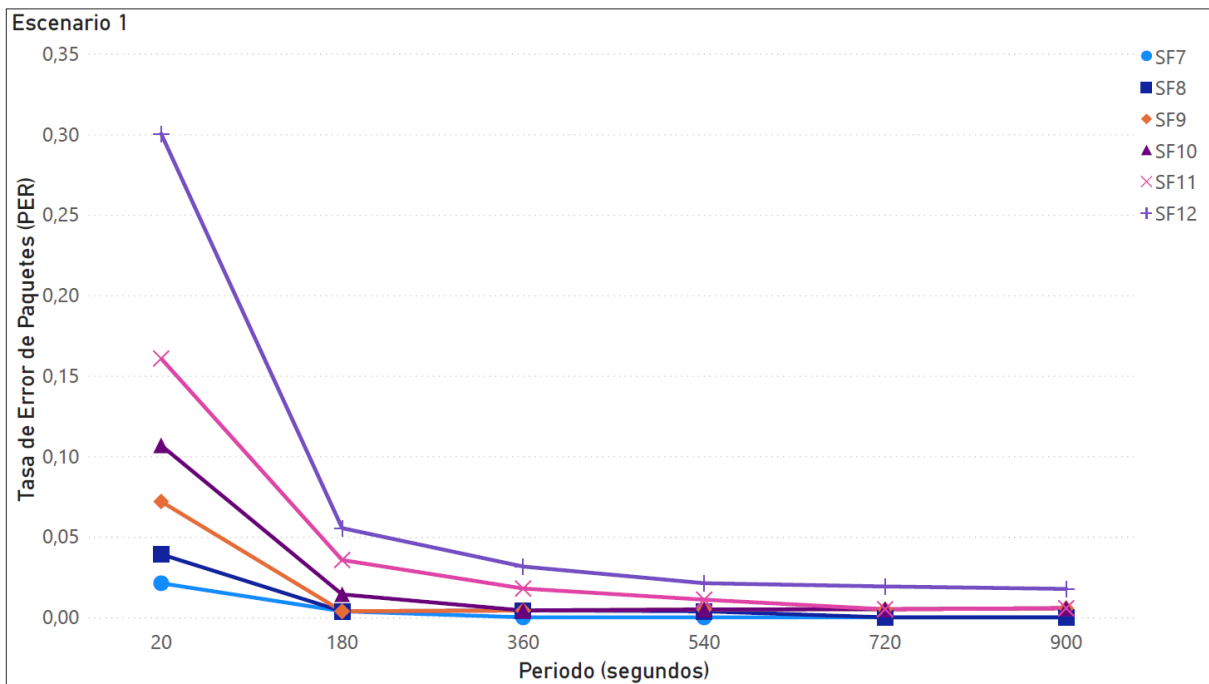
```
df = pd.DataFrame(
    data_final,
    columns=[
        "Time (second)", "ED", "SF", "Tx(dBm)", "Rx(dBm)", "SNR(dBm)", "Dst(m)",
        "Delay(ns)", "ED_Packet", "Received", "Time_Received (second)",
        "ACK_Received", "Msg_Before_ACK", "Sub-Banda (MHz)", "Latitude",
        "Longitude"],
)
```

Con esta funcionalidad se recogen todos los datos según las diferentes especificaciones, allanando el uso de los diferentes conjuntos de datos.

4.4.2. Fiabilidad del primer escenario

Los resultados referentes a la fiabilidad de las comunicaciones indican una menor tasa de error de paquetes enviados (PER o *Packet Error Rate*) por ED para los SF pequeños, del entorno del 7 al 9, y en ambos modelos, con y sin vehículo. Con la potencia de transmisión máxima permitida (Tx = 14 dBm), variando los periodos de envío y SF, en la Figura 24 y Figura 25 se observan fluctuaciones importantes del PER para los periodos más pequeños que se magnifican para aquellos SF más altos, seguido de una estabilización a medida que los periodos van en aumento.

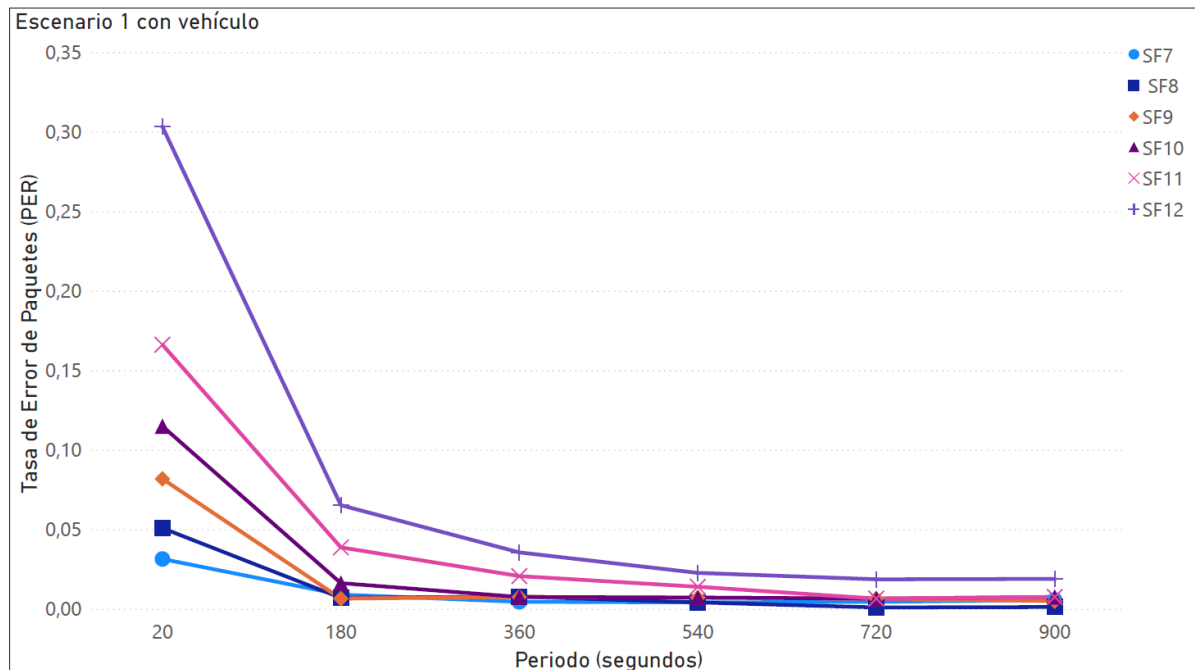
Figura 24. Gráfico PER-Periodo para el primer escenario.



Fuente: Elaboración propia.

Según los modelos presentados, los más fiables a lo largo de todos los periodos evaluados son los que emplean SF bajos y medios, del rango 7 al 10.

Figura 25. Gráfico PER-Periodo para el primer escenario con vehículo.



Fuente: Elaboración propia.

Para el caso comparativo entre el mismo escenario con y sin vehículo, existen pequeñas desviaciones entre los valores PER y periodos, acentuadas para los periodos pequeños. En cuanto al modelo que contempla el vehículo, la fiabilidad mejora en SF altos y se reduce para los bajos.

Referente a las causas de las grandes desviaciones del PER entre periodos, estas pueden ser causadas por distintos tipos de orígenes. En primer lugar, del SF9 al SF12, tal y como se indica al inicio de esta fase, donde se argumenta la casuística de la saturación de los canales por razón del cumplimiento del "Duty Cycle", esto provoca la menor generación de eventos y genera mayores tiempos de transmisiones ("Time on Air").

Además, existe el "Shadowing" y las distorsiones, que también juegan un papel en esta pérdida de mensajes. Aún con todo ello, este proyecto no discierne la causalidad de estos errores y se concentra en la evaluación total de la fiabilidad.

Es por ello por lo que, si se evalúan con más detalle, en todos ellos se aprecia claramente una mayor tasa de error en el modelo que emplea el vehículo. En este caso se mantiene la disminución a medida que los periodos aumentan en duración, eso sí, los comportamientos difieren según SF y presencia del vehículo.

Los SF9 y SF10 se estabilizan con los periodos grandes, en ambos casos, y sufren fluctuaciones pequeñas (≤ 0.01), en cambio el comportamiento para los SF7 y SF8 sí varían. Mientras que en el escenario sin vehículo se interpreta que el modelo con más fiabilidad es el que emplea el SF7, su comportamiento se inestabiliza con la detección del vehículo. Con la información recogida y el análisis del PER establecido, todo indica que la fiabilidad pasa por el SF7 hasta el SF12, por orden enumerativo.

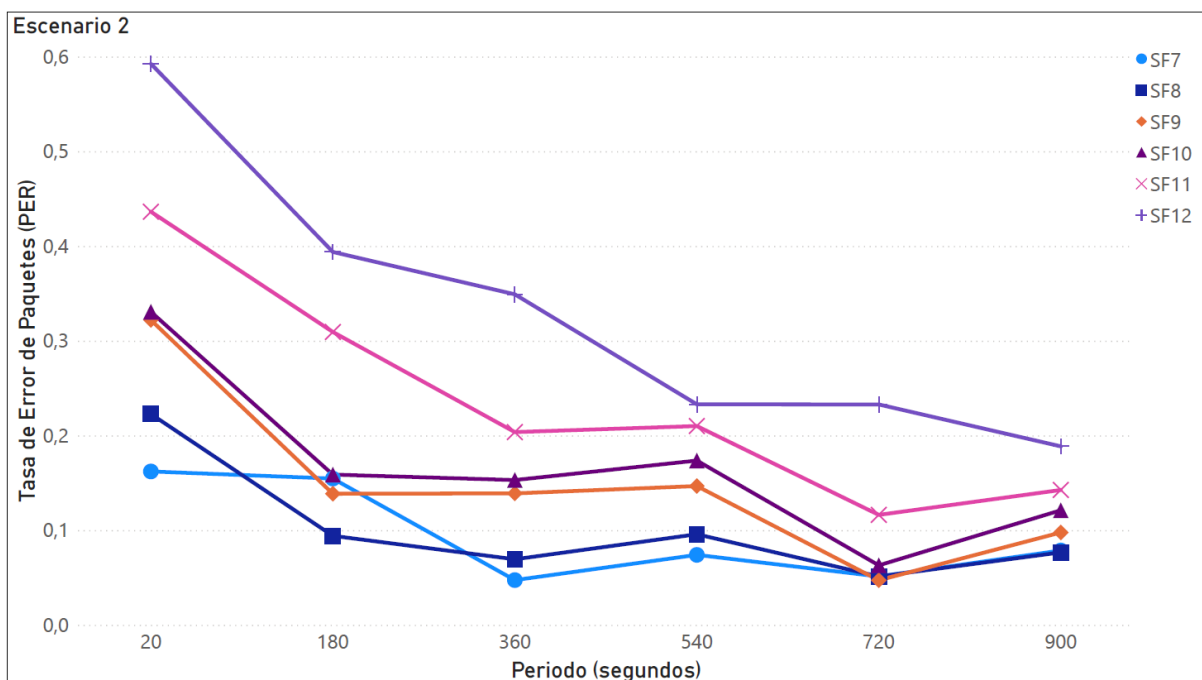
4.4.3. Fiabilidad del segundo escenario

Este escenario, en términos generales, el PER empeora para todos los modelos ya sea solo con el ACK, como para el modelo que contempla el ACK y ADR. Observando los resultados de la Figura 26 y Figura 27 en seguida se aprecian mayores desviaciones en los datos a medida que los periodos aumentan, a la vez que el PER. En este sentido, aún se conserva una tendencia a la disminución del PER según los periodos van aumentando, pero esta vez en mucha menor medida que en el primer escenario.

Otro aspecto importante, asociado a estos resultados, son la cantidad de mensajes que se envían, principalmente en el periodo más pequeño (20 segundos), y sus repercusiones en el DC. En efecto, se supera el límite del 1% de DC asumible para la GW siempre que se usan los SF y periodos más bajos. Este hecho puede ser observado en los gráficos de los anexos, de la Figura 41 hasta la Figura 46, donde se compara el DC según el canal respecto al 1% marcado por la línea de color naranja.

En los gráficos de los anexos descritos se comparan el DC que se emplea en cada escenario, y modelo de uso, primero en los gráficos (a) y (b) para el primer escenario y el segundo en los (c) y (d) para el segundo escenario, estas figuras ya dan una primera pista respecto a este límite en el DC. En los gráficos referentes al primer escenario, todos los que superan el 1% del DC y su "Time on Air" es inferior al mínimo establecido, se ven agravados para el segundo escenario por el uso del ACK.

Figura 26. Gráfico PER-Periodo para el segundo escenario.



Fuente: Elaboración propia.

El comportamiento es totalmente lógico, puesto que el uso de la calculadora interna del DC que emplea el simulador, al quedar saturados los ocho canales que puede emplear cualquier dispositivo, ese genera un retardo en la respuesta y, en poco tiempo, se genera una reacción en cadena que empeora aún más la situación. Los ED no reciben el ACK a tiempo y reenvían el

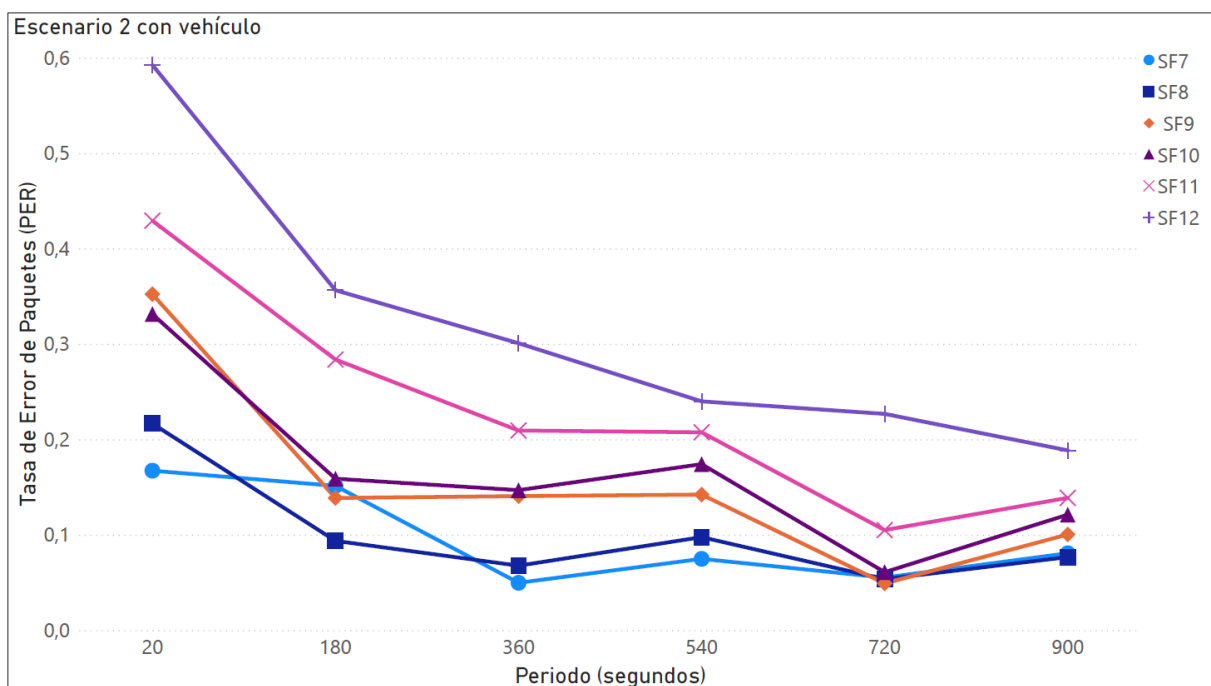
mensaje (hasta tres veces), generando una cola aún mayor de respuestas y uso de los canales. Es por ello por lo que los resultados que se aprecian llegan a triplicar el margen del DC, en algunos casos.

Los datos del DC indican la necesidad de más GW para cumplir con las expectativas del uso para el ACK, en ningún caso se puede asociar una cantidad exacta de estas. Este proyecto contempla el mejor escenario empleando una sola GW, por esta razón no se analiza más profundamente este tema.

En este escenario, la tasa de errores en los modelos con sólo el ACK activado, tiene una mayor variabilidad durante todos los periodos, salvo en algunos casos específicos como los SF11 y SF12, los cuales, en ambos casos con y sin vehículo representan un comportamiento parecido.

Para determinar con mayor contundencia cuál de los SF mantiene un mejor PER, se calculan las medianas en el conjunto de las simulaciones para este escenario, en la Figura 28 y, las desviaciones típicas quedan representadas en la Figura 29, según el SF empleado.

Figura 27. Gráfico PER-Periodo para el segundo escenario con coche.



Fuente: Elaboración propia.

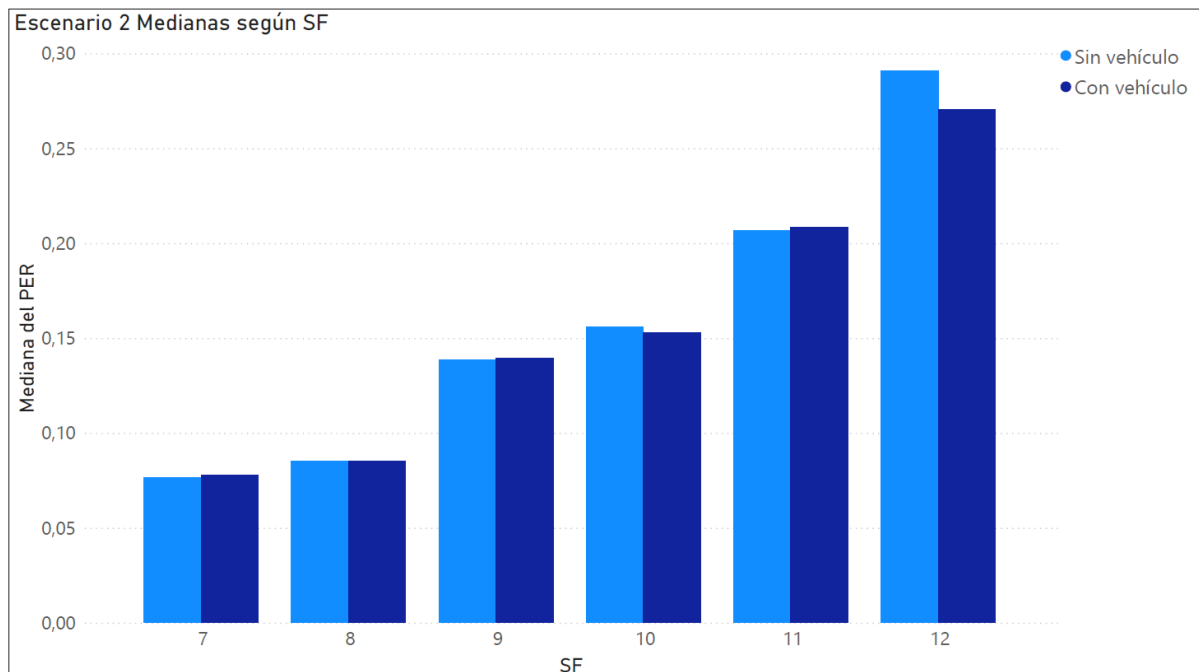
Los datos muestran claramente que el mejor resultado vuelve a estar enumerado de forma progresiva desde los SF más bajo hasta el mayor. Sin embargo, con estos datos se aprecia con mayor detalle la poca diferencia en la mediana del PER y los modelos con y sin vehículo. Este

comportamiento se repite para los datos de las desviaciones a excepción del SF9, a espera del análisis de la cobertura en subsecciones posteriores no se decide analizar con más detalle este aspecto, el análisis de la fidelidad para este modelo se considera cumplido.

En cuanto al empleo de la modalidad ADR y ACK juntos, modelado para empezar con el SF 12 y la Tx a 14 dBm, estas representan una fiabilidad parecida a los SF bajos, aunque empeora respecto estos. Si bien en la Figura 30 se pueden observar los resultados de todos los periodos según disponibilidad del vehículo o no, los valores del PER son mayores que los mejores modelos empleados hasta la fecha.

El empleo del ADR para las aplicaciones de aparcamientos no es recomendable y en este proyecto así queda constatado. Según los resultados adquiridos, las fluctuaciones de los factores de dispersión y potencia de transmisión de los anexos representan la visualización del agravio con respecto a la vida útil del dispositivo y del uso extra de los canales de comunicación de manera fútil.

Figura 28. Mediana del PER según SF en el segundo escenario.



Fuente: Elaboración propia.

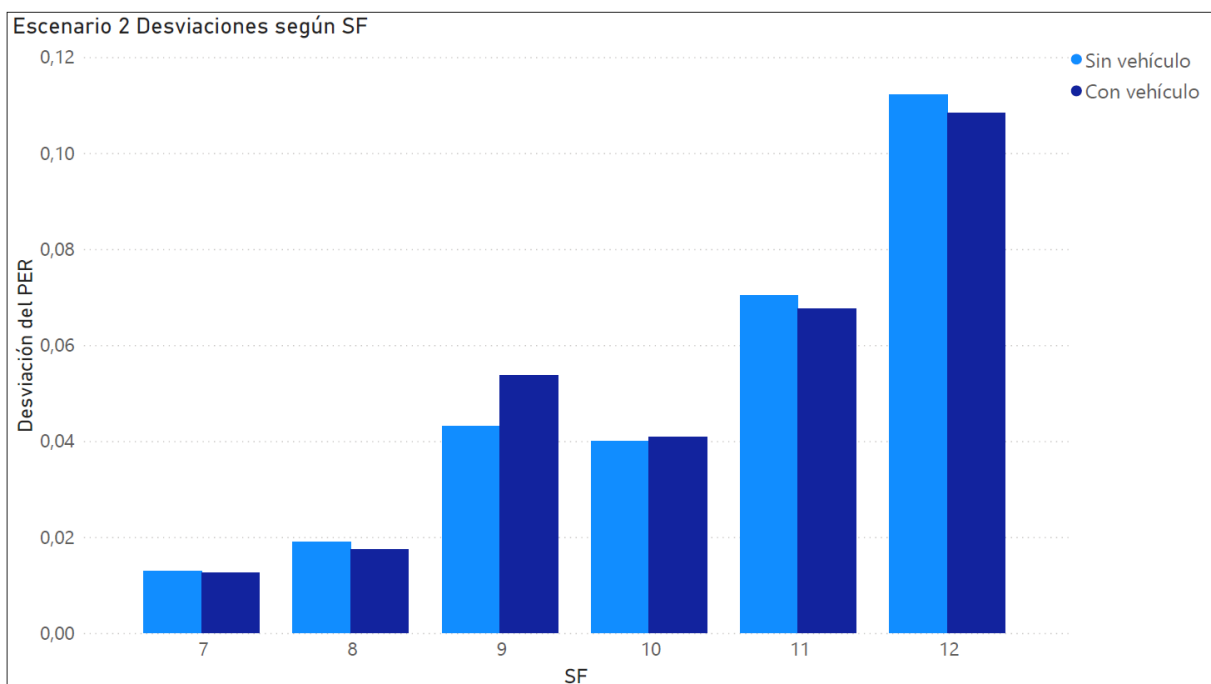
En el escenario empleado el SF no representa un problema para la funcionalidad del ADR ya que, llegado el momento, este lo modifica para la totalidad de los ED y no lo vuelve a modificar para el resto de la duración de la simulación. Dicho comportamiento se puede apreciar en los

anexos, según la Figura 53 hasta la Figura 58, las cuales indican que el comportamiento del SF para cada periodo simulado es indistinto (mayoritariamente) de la presencia del vehículo.

Por otro lado, se encuentra la potencia de transmisión (Tx), la cual representa el verdadero problema por el cual el ADR se ensaña en tratar de resolver a lo largo de la simulación. Los datos recabados y mostrados en la Figura 47 hasta la Figura 52 en los anexos muestran las comparaciones en los cambios del Tx por el escenario sin vehículo (a) y con vehículo (b) demostrando el comportamiento caótico.

Los gráficos de dichas figuras muestran que el ADR intenta reducir al máximo el valor de Tx al inicio, al igual que el SF, pero esta vez las variaciones en la potencia de la señal recibida (Rx) fluctúan demasiado.

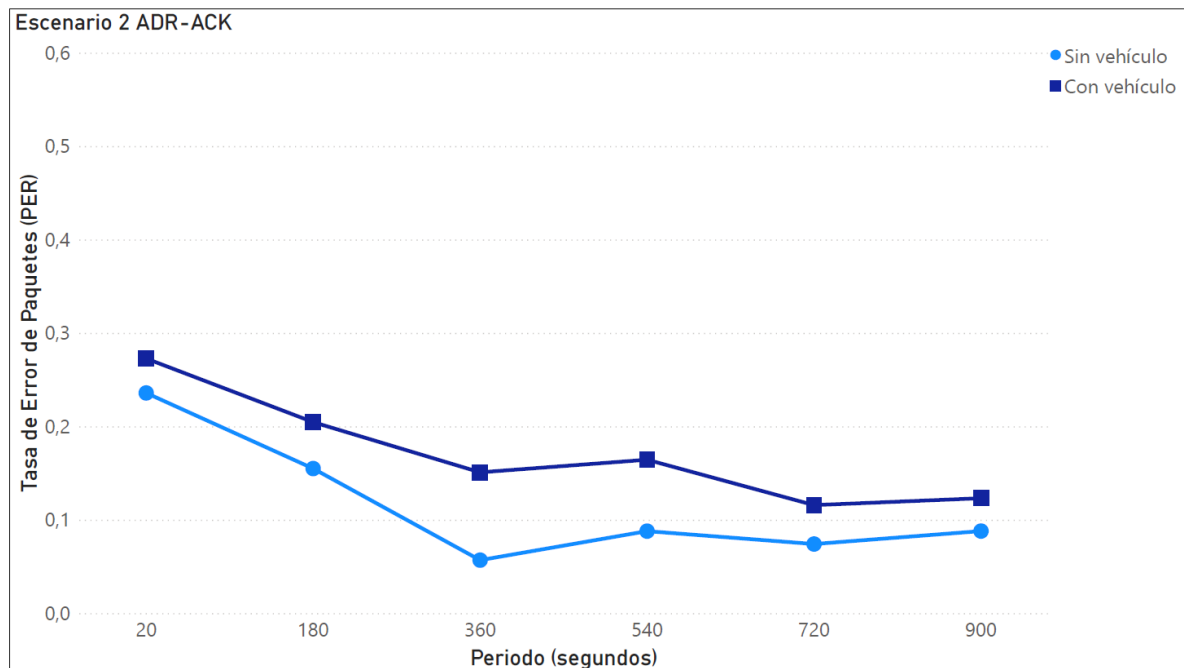
Figura 29. Desviación del PER según SF en el segundo escenario.



Fuente: Elaboración propia.

El sistema ADR intenta compensar este comportamiento para el resto de la simulación, pero dos hechos se lo impiden.

En primer lugar, el ADR guarda los últimos 20 mensajes de cada ED y es luego cuando decide actuar, en segundo lugar, a este hecho se le añade la gran aleatoriedad debido a la fluctuación de los distintos vehículos, provocando el comportamiento que se observa en las figuras de los anexos descritos con anterioridad.

Figura 30. Gráfico PER-Periodo para el segundo escenario con ADR y ACK.

Fuente: Elaboración propia.

4.4.4. Análisis previo al escenario del consumo

Una vez analizados los dos escenarios principales según el PER, en esta subsección se analiza el impacto en la cobertura de los dispositivos según la potencia recibida o Rx y la relación señal y ruido o SNR. Estas métricas solo se ven afectadas por el poder de transmisión (Tx) y las atenuaciones provocadas por los elementos del entorno tales como el modelo de propagación, los obstáculos y otros, por esta razón a continuación se presentan los mapas con dichas métricas y se analizan los datos relacionados. En primer lugar, se encuentra el mapa de cobertura sin emplear el vehículo, la Figura 31, en esta se visualiza el promedio de las potencias de las señales recibidas según ED.

Los valores que se representan en todos los mapas y figuras de esta subsección son extraídos del primer escenario para el SF7 ya que la cantidad de datos generados es el mayor. Estos datos oscilan entre los (-66.55 dBm, -113.29 dBm) en el modelo sin vehículo que, a excepción de un par de datos atípicos, todos los ED siguen una tendencia levemente al alza, este oscila en el entorno a los -100 dBm con un promedio de -105.03 dBm, representado en el gráfico (a) de la Figura 33, e indican una leve mejora en la señal para aquellos ED más cercanos a la GW, en este caso la columna de dispositivos situados al Este.

Los valores atípicos corresponden a los ED 9 y 10, este hecho se vincula al escenario simulado y la acción del modelo de propagación, los dos ED mantienen una línea de visión directa con la GW, por ese motivo el modelo de propagación le otorga la atenuación correspondiente a LOS, y NLOS para el resto de los ED.

Por otra parte, cuando se aplica el modelo con el uso del vehículo, la atenuación provocada por este se proyecta en mapa en la Figura 32. Todos los ED sufren una atenuación en el Rx, aumentando el rango de valores, ahora situado entre (-75.52 dBm, -119.89 dBm) e invirtiendo la tendencia, representada en la Figura 33, gráfico (b), la cual ahora denota una leve inclinación al empeoramiento para la columna de ED del Este.

Figura 31. Mapa de la posición de los dispositivos sensoriales en plaza Catalunya según el Rx sin vehículo.



Fuente: Elaboración propia.

Estos mismos análisis pueden ser atribuidos para el caso del SNR, mapas y gráficos que se encuentran en los anexos de este proyecto, de la Figura 59 hasta la Figura 61. En estas figuras se encuentra la misma disposición de los mapas y tendencias de los datos SNR por ED.

Los resultados para la cobertura indican ciertos márgenes que pueden ser explotados y analizados con tal de mejorar la vida útil del dispositivo. Considerados todos los datos hasta

este punto, siguiendo la jerarquía de prioridades establecidas (Fiabilidad, Cobertura y Consumo), se procede a analizar si hay margen en la modificación del Tx.

Antes de realizar tal análisis es necesario argumentar que el modelo con el cual se realizan tales pruebas es aquel que ha presentado mayores beneficios. Este se trata del modelo del primer escenario con SF7, la fiabilidad y cobertura resultantes lo avalan.

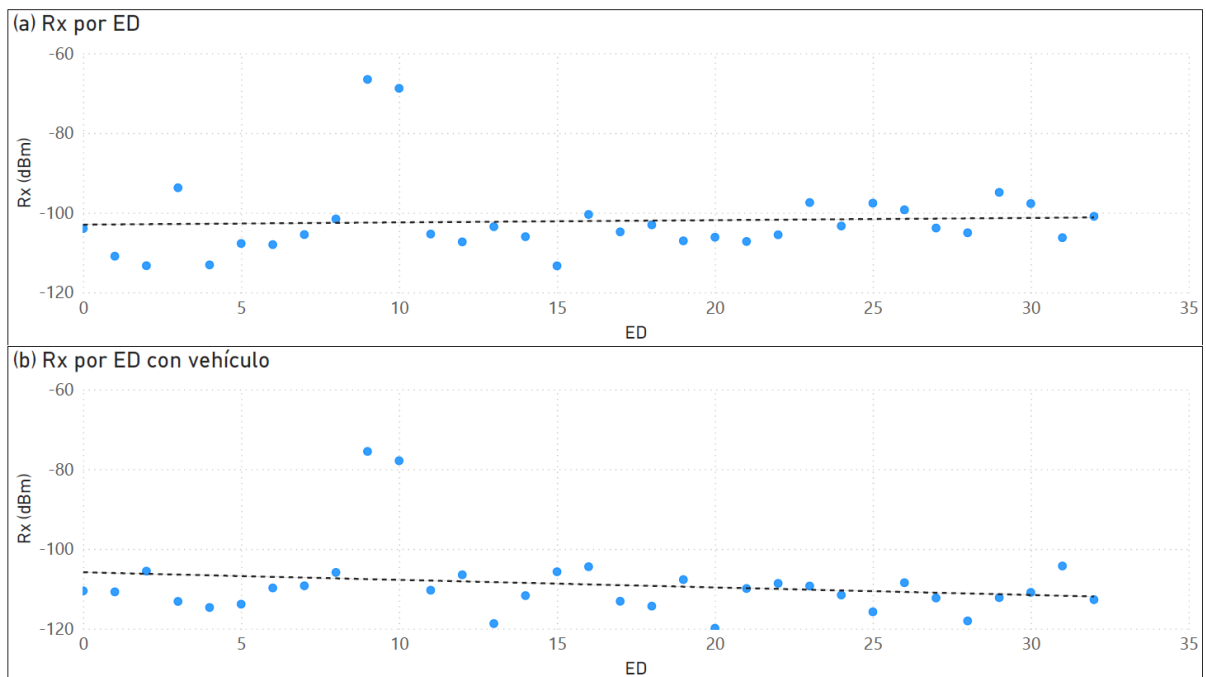
Figura 32. Mapa de la posición de los dispositivos sensoriales en plaza Catalunya según el Rx con vehículo.



Fuente: Elaboración propia.

A parte, por falta de datos reales que avalen unos periodos del comportamiento habitual del tráfico para aquella zona y, para no tratar de generar datos y comportamientos demasiado fuera de lo normal, el periodo que se va a emplear resulta de 100 segundos, siendo esta una media entre los dos periodos más pequeños.

En consecuencia, a continuación, se analizan los datos generados por la variación del Tx, en el rango de 2 – 14 dBm con saltos pares de valores. En primer lugar, se encuentra una clara relación entre el hecho de emplear el vehículo como objeto de atenuación o no, según los datos obtenidos y trazados en la Figura 34 y Figura 35.

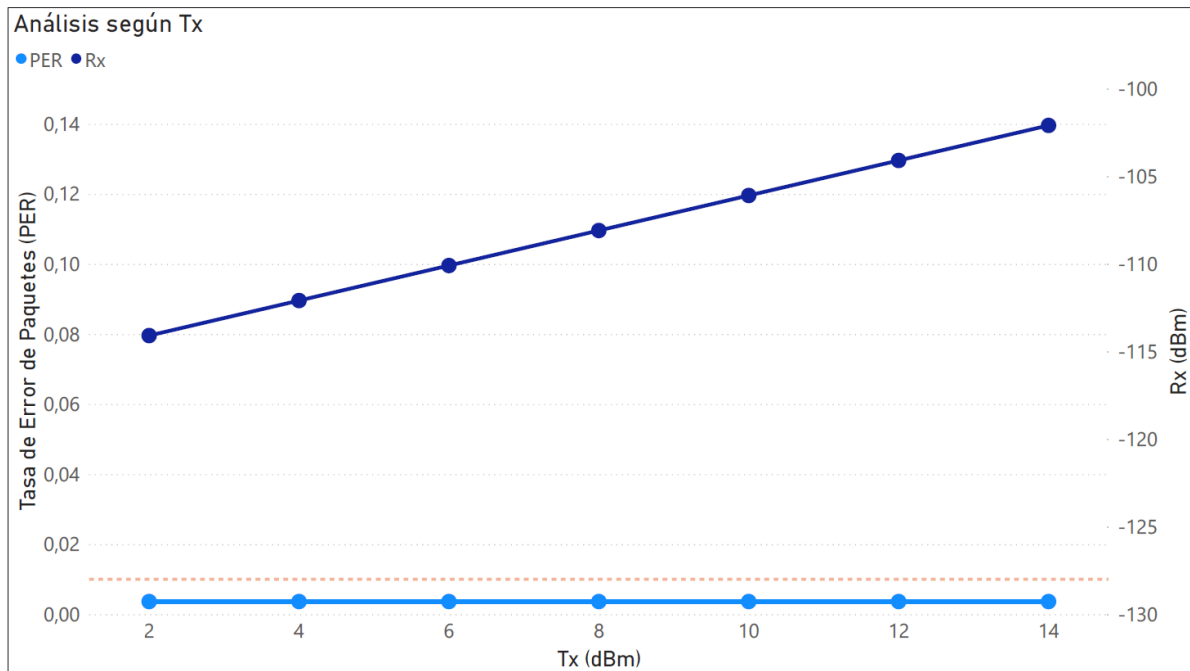
Figura 33. Visualización de los promedios del Rx según ED.

Fuente: Elaboración propia.

En dichas figuras se emplean los promedios obtenidos del PER y el Rx según el rango de Tx empleado, además se añade una traza de color naranja para determinar el máximo PER asumible según KPI (1%). Según estos gráficos el PER sufre un considerable aumento debido a la atenuación del coche, comportamiento que no sucede sin este. Este hecho se atañe al poco margen que existe entre la sensibilidad de la GW a -130 dBm y el SF7 empleado, quedando contrastado por la relación que se encuentra en el PER al cruzar este límite, cuantos más mensajes lo sobrepasan mayor es la tasa de error. A la vez deja claro que la simulación funciona correctamente ya que el algoritmo de cobertura reacciona como es debido.

Todos los gráficos que se emplean para contrastar dicho comportamiento se encuentran en los anexos de este documento, en concreto abarcan desde la Figura 62 hasta la Figura 65, donde engloban las mediciones del Rx según el ED y el Tx para cada caso. También se puede observar una línea naranja que delimita el límite de sensibilidad de la GW para el SF7, situado a -130 dBm. Con estos resultados no es de extrañar que para maximizar los resultados de alguno de los tres puntos clave (Fiabilidad, Cobertura y Consumo) esto implique empeorar alguno de ellos. Según el KPI del sistema que afecta directamente a la fiabilidad del sistema, este no puede sobrepasar el 1% de tasa de error.

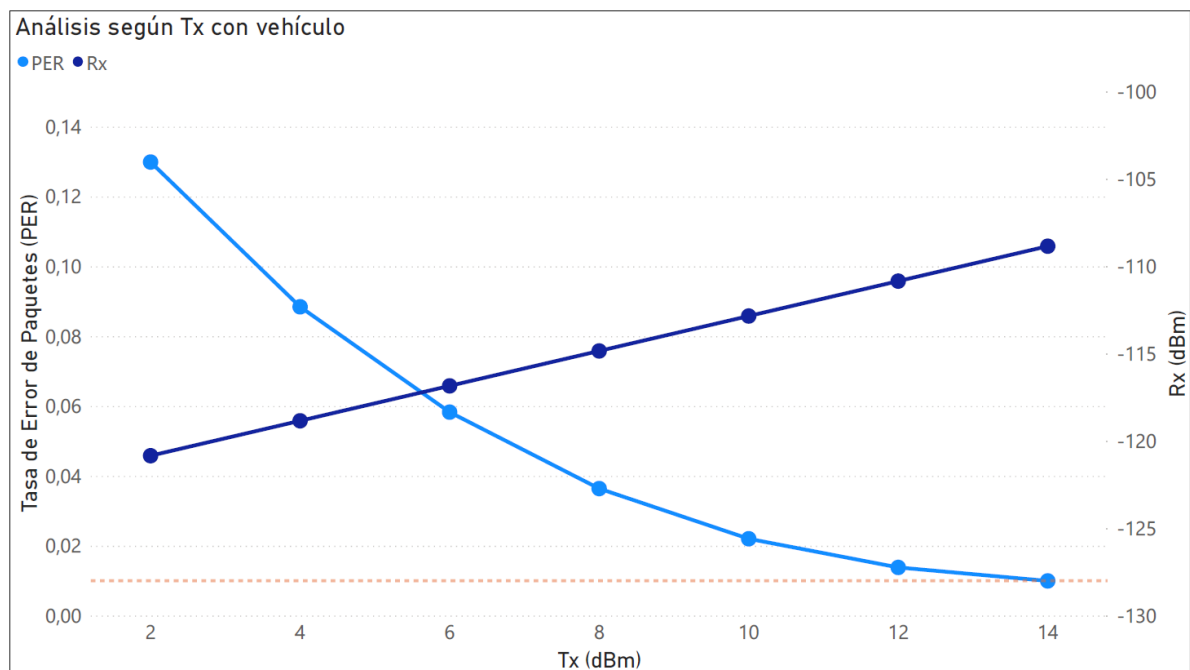
Figura 34. Datos cruzados del Tx según PER y Rx sin vehículo según configuración analizada.



Fuente: Elaboración propia.

Por este motivo, el poco margen de sensibilidad que ofrece el SF7 respecto la GW, relacionados con los datos obtenidos del uso del vehículo y la gran capacidad de la batería, el último escenario contempla el Tx a 14 dBm.

Figura 35. Datos cruzados del Tx según PER y Rx con vehículo según configuración analizada.



Fuente: Elaboración propia.

4.4.5. Consumo del tercer escenario para el dispositivo

El primer lugar se debe tener en cuenta los márgenes de funcionamiento del sensor, los cuales se establecen empezando a 3.60 V con una energía de 207 360 J hasta los 2.90 V, a 167 040 J de energía. Estos márgenes son los datos que se toman para determinar la duración total del dispositivo.

Según los resultados de la simulación empleada con periodos de 100 segundos, SF7 y los consumos de la Tabla 10, la vida útil del dispositivo representa un tiempo de 293.35 periodos de 12 horas hasta que el dispositivo llega al límite inferior. Este periodo es claramente inferior al par de años preestablecidos por el KPI, además va a tratarse de una duración inferior a la descrita debido a distintas causalidades como el uso que les vaya a dar, la cantidad de calibraciones, ambiente y demás factores que no se contemplan en este proyecto. Por este motivo se recogen los datos para el resto de los periodos superiores y se genera la Tabla 11.

De la tabla descrita con anterioridad se genera el gráfico de la Figura 36, puesto que de esta manera se da un rango de duraciones máximas a las cuales el dispositivo puede estar sometido.

Tabla 11. Periodos de duración de la vida útil del dispositivo sensorial.

Periodo (s)	Días (12 horas)	Años
100	293.35	0.80
180	359.58	0.99
360	717.78	1.97
540	1074.69	2.94
720	1434.28	3.93
900	1793.86	4.91

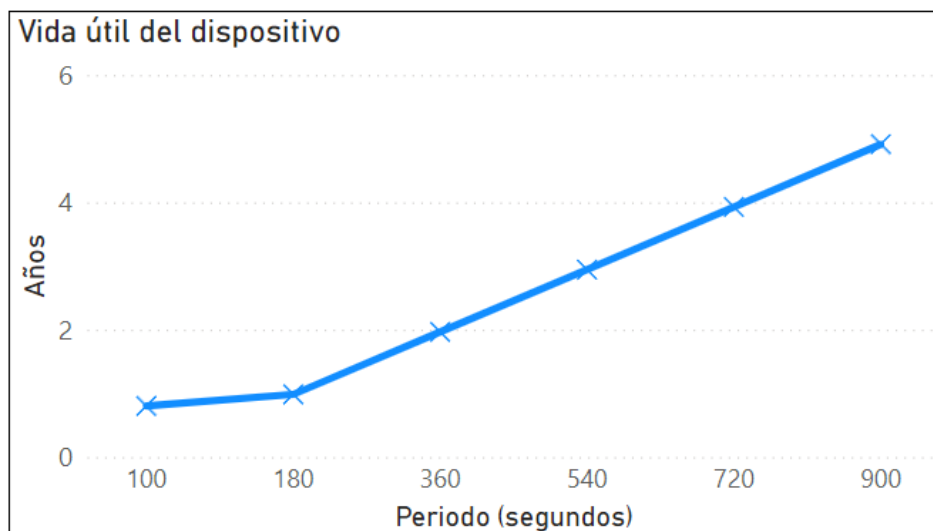
Fuente: Elaboración propia.

En ella se puede observar la curva de generada de la vida útil del dispositivo y delimitando el cuadro de duraciones que se pueden tener en cuenta en sistemas de mantenimientos preventivos o de predicción, entre otros.

En este marco de valores se deja establecido que el par de años mínimo de funcionamiento para este dispositivo se encuentran en los periodos de 360 y 540 segundos a partir de este punto solo puede mejorar, siempre que se emplee un uso corresponsable del dispositivo y no se modifiquen sus características intrínsecas.

Mientras Winext garantiza una duración de hasta 4 años con 20 mensajes diarios con el SF12, con las implementaciones realizadas se observa que se pueden llegar a mayores periodos de duración de vida útil superando este dato con un SF inferior y una periodicidad mucho menor, los cuales son aceptables para este proyecto.

Figura 36. *Vida útil del dispositivo sensorial respecto los periodos entre eventos.*



Fuente: Elaboración propia.

5. Conclusiones y trabajos futuros

Este proyecto finaliza con la resolución de este capítulo, el cual queda dividido en dos secciones. En la primera, la Sección 5.1 describe los hechos argumentales de los objetivos marcados por este trabajo, enumerando todos aquellos aspectos, ya se hayan cumplido o no. En la segunda, la Sección 5.2 detalla elementos importantes que deberían explorarse en un futuro próximo si lo que se pretende es mejorar este trabajo.

5.1. Conclusiones

En la Introducción de este trabajo alega la necesidad de la implantación de soluciones IoT en el paradigma de las Smart Cities. Algunos de los aspectos más importantes como es la gestión del tráfico y la problemática asociada que generan como la contaminación y los retardos son de vital importancia en las ciudades. Tras el Capítulo de 2 Contexto y estado del arte se atañe la falta de proyectos en sistemas de aparcamiento en que proporcionan lo siguiente:

- Evaluaciones para sensores que se pretenden llegar a implementar.
- Realización de escenarios simulados con el máximo de detalles reales (escenario, sensores, protocolo de comunicaciones) que proporcionen información relevante al proyecto antes de que este se lleve a cabo.

Dada esta situación, este trabajo expone un procedimiento de cuatro fases expuestas en la Sección 3.2 Metodología del trabajo en la que se pretende completar con la culminación del objetivo principal de la Sección 3.1 Objetivos definido como:

- Implementar y evaluar de los resultados de la simulación, realizada con el ns-3, en la que se ha tomado en consideración el máximo de aspectos posibles y similares al entorno urbano de Girona junto al mejor dispositivo sensorial seleccionado.

Con tal de obtener este objetivo se plantea granular este en objetivos específicos, todos ellos necesarios para alcanzar dicho logro. En este sentido, los resultados distribuidos en todo el documento exponen el alcance de los siguientes hitos:

- **Buscar la información relevante a través de fuentes públicas.** Todas las características de los sensores, escenario y terreno, así como del protocolo de comunicaciones (LoRa/LoRaWAN) y listado de sensores es pública y de uso libre. El uso de distintas

fuentes como el *Institut Cartogràfic i Geològic de Catalunya*, s. f., el *Instituto Geográfico Nacional*, s. f. o *OpenStreetMap*, s. f. son de especial relevancia para la obtención de los detalles del terreno. Así como toda aquella documentación disponible de los distintos sensores, las comunicaciones LoRa, el simulador ns-3 y Power BI que se pueden encontrar en la Fase 1: Definición del caso de uso y la Fase 2: Selección de la información del Capítulo 4.

- **Seleccionar el sensor Winext.** Este sensor ha sido evaluado, teniendo en cuenta toda la información recabada y contrastada posible. A través de distintos factores como el consumo, el sensor empleado, las características físicas y demás encontrados en el argumentario de la Sección 4.2 Fase 2: Selección de la información.
- **Correcto diseño e implementación de la simulación empleando el ns-3.** Según se han llevado a cabo las distintas fases, se ha seleccionado el mejor modelo de propagación de la señal para las comunicaciones LWPAN posible dentro del ns-3 (3Gpp V2V, de la Sección 4.3), añadiendo factores de adversidad como el algoritmo “Shadowing”. Además, partiendo del módulo completo de *LoRaWAN ns-3 module (2017/2023)* se han podido generar todos los elementos indispensables (ED, GW, NS) para la recreación de una comunicación LoRa/LoRaWAN con todas sus características (SF, Tx, ADR, ACK, entre otros). Finalizar con la creación de un entorno urbano gracias al módulo de creación de edificaciones para mejorar el algoritmo de propagación de la señal, el cual se puede encontrar en la Sección 4.3, en concreto en la Subsección 4.3.2.
- **Generar el *script* de Python para la ingesta de los datos.** Se ha creado un programa específico para la búsqueda de información para todos los archivos de historial de los resultados generados para cada escenario y condición empleada en ellos. Este se encuentra descrito en el Subsección 4.4.1 de la Sección 4.4. De esta manera se convierten todos los datos en tipología estructurada y así poder ser usadas por la herramienta de visualización de Power BI.
- **Realizar los distintos dashboard para la visualización de los datos.** A lo largo de todo el trabajo se han desarrollado gráficas de distintas casuísticas como gráficos de barras, de líneas, de puntos, entre otras. Este hecho parte de la premisa de facilitar el traspaso de información y agilizar el proceso de percepción del conocimiento generado.
- **Generar la información, conocimiento y valor a través de los resultados de las simulaciones.** Los argumentos originarios del análisis final de las simulaciones, desde

la Subsección 4.4.1 hasta la Subsección 4.4.5, permiten establecer un marco en el cual es posible alcanzar totalmente todos los OKR y KPI propuestos, a excepción de la duración de la batería, la cual se alcanza parcialmente, ya que depende estrictamente del uso al cual se vea sometido el sensor. Además, se establece la configuración óptima de las comunicaciones del dispositivo Winext en el SF7 y la potencia de transmisión Tx en 14 dBm, juntamente con un rango de vida útil de entre menos de un año hasta casi los cinco. Esto ha sido posible gracias a la constante consideración de estos objetivos e indicadores, ya sea para la evaluación del sensor como en la realización de la simulación y el estudio realizado en la Sección 4.4.

Finalmente, siguiendo con el criterio de cumplimiento de todos los puntos descritos anteriormente, se logra que el proyecto realizado por este autor cumpla con su objetivo principal.

5.2. Líneas de trabajo futuras

Este trabajo contempla multitud de aspectos y temas que tratan de cumplir con los objetivos marcados. Aun así, existen caminos que se sería beneficioso explorar para añadir rotundidad y alejar focos originarios de incertidumbres, a continuación, se enumeran las más importantes y destacadas.

- **Mejorar el sistema de simulación sensorial LoRa en el ns-3 para los edificios:** Existen multitud de efectos distorsionadores en la propagación de la señal en las comunicaciones sin cables, rebotes, climatología, disconformidad en el terreno y obstáculos, entre otros. En este aspecto, en la simulación empleada se argumenta la posibilidad de la futura afectación de dichos factores en los resultados de la simulación, por ello existen variaciones. Por ello, resulta interesante trabajar en este aspecto para añadir estos comportamientos y factores para ir reduciendo el margen o variación de las simulaciones. De esta forma se abre un abanico de posibilidades de implementación de estas simulaciones en entornos urbanos.
- **Trabajar para encontrar la causalidad del PER (*Packet Error Rate*):** Existen multitud de factores para los cuales este factor puede aumentar, este proyecto no los contempla, abriendo la puerta a la generación de trabajos más profundos y exhaustivos en esta línea. Disminuir el PER hallando estas razones puede ayudar a

detectar y generar nuevas oportunidades de mejoras para el sistema desempañado en este trabajo y parecidos.

- **Modelar el contraste cruzado con valores reales:** En este trabajo no se contempla una comparación con una prueba de concepto o parecido. Aunque las pruebas realizadas con el dispositivo seleccionado vayan en esta dirección, estas suponen solamente un inicio e indican el camino para realizar una prueba de mayor envergadura, la cual cuanta mayor similitud con el escenario, mayor será la validación del modelo simulado.
- **Introducir una mejora en la distribución de generación de eventos de telemetría.** En este sentido, se debería explorar la posibilidad de implementación de tiempos aleatorios para la generación de eventos empleando sistemas más complejos que las iteraciones periódicas. Existen métodos de distribución, como “Weibull” u otros, que podrían aportar un enfoque distinto al escenario de las simulaciones realizadas en este trabajo. Esto podría añadir un mayor factor de aleatoriedad para los estacionamientos.

Referencias bibliográficas

- Abduljabbar, R. L., Liyanage, S., & Dia, H. (2021). The role of micro-mobility in shaping sustainable cities: A systematic literature review. *Transportation Research Part D: Transport and Environment*, 92, 102734. <https://doi.org/10.1016/j.trd.2021.102734>
- Abu-Rayash, A., & Dincer, I. (2020). Analysis of mobility trends during the COVID-19 coronavirus pandemic: Exploring the impacts on global aviation and travel in selected cities. *Energy Research & Social Science*, 68, 101693. <https://doi.org/10.1016/j.erss.2020.101693>
- Ajuntament de Girona*. (2022). Ajuntament de Girona. <https://web.girona.cat>
- Alavi, A. H., Jiao, P., Buttlar, W. G., & Lajnef, N. (2018). Internet of Things-enabled smart cities: State-of-the-art and future trends. *Measurement*, 129, 589-606. <https://doi.org/10.1016/j.measurement.2018.07.067>
- Allam, Z., & Dhunny, Z. A. (2019). On big data, artificial intelligence and smart cities. *Cities*, 89, 80-91. <https://doi.org/10.1016/j.cities.2019.01.032>
- Al-Turjman, F., & Malekloo, A. (2019). Smart parking in IoT-enabled cities: A survey. *Sustainable Cities and Society*, 49, 101608. <https://doi.org/10.1016/j.scs.2019.101608>
- Amiri, W. A., Baza, M., Banawan, K., Mahmoud, M., Alasmay, W., & Akkaya, K. (2019). Privacy-Preserving Smart Parking System Using Blockchain and Private Information Retrieval.

2019 International Conference on Smart Applications, Communications and Networking (SmartNets), 1-6.

<https://doi.org/10.1109/SmartNets48225.2019.9069783>

Anenberg, S., Miller, J., Henze, D., & Minjares, R. (2019). A global snapshot of the air pollution-related health impacts of transportation sector emissions in 2010 and 2015. *International Council on Clean Transportation: Washington, DC, USA*.

ATIS. (s. f.). Recuperado 31 de mayo de 2023, de <https://www.atis.org/>

Awan, F. M., Saleem, Y., Minerva, R., & Crespi, N. (2020). A comparative analysis of machine/deep learning models for parking space availability prediction. *Sensors*, 20(1), Article 1. <https://doi.org/10.3390/s20010322>

Bhattacharya, S., Somayaji, S. R. K., Gadekallu, T. R., Alazab, M., & Maddikunta, P. K. R. (2022). A review on deep learning for future smart cities. *Internet Technology Letters*, 5(1), e187. <https://doi.org/10.1002/itl2.187>

Boban, M., Gong, X., & Xu, W. (2016). Modeling the Evolution of Line-of-Sight Blockage for V2V Channels. *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, 1-7. <https://doi.org/10.1109/VTCFall.2016.7881090>

Bock, F., Di Martino, S., & Origlia, A. (2019). Smart parking: Using a crowd of taxis to sense on-street parking space availability. *IEEE Transactions on Intelligent Transportation Systems*, 21(2), 496-508.

BOSCH. (2023). *BOSCH Parking Lot Sensor*. Robert Bosch GmbH.
<https://www.bosch.com/products/connected-mobility/parking-lot-sensor/downloads/>

Capuzzo, M., Magrin, D., & Zanella, A. (2018). *Confirmed traffic in LoRaWAN: Pitfalls and countermeasures*. 1-7.

Celaya-Echarri, M., Froiz-Míguez, I., Azpilicueta, L., Fraga-Lamas, P., Lopez-Iturri, P., Falcone, F., & Fernández-Caramés, T. M. (2020). Building Decentralized Fog Computing-Based Smart Parking Systems: From Deterministic Propagation Modeling to Practical Deployment. *IEEE Access*, 8, 117666-117688.
<https://doi.org/10.1109/ACCESS.2020.3004745>

Chang, J., Nimer Kadry, S., & Krishnamoorthy, S. (2020). Review and synthesis of Big Data analytics and computing for smart sustainable cities. *IET Intelligent Transport Systems*, 14(11), 1363-1370. <https://doi.org/10.1049/iet-its.2020.0006>

Cookson, G., & Pishue, B. (2017). Inrix global traffic scorecard—appendices. *INRIX research*.

de Paula Ferreira, W., Armellini, F., & De Santa-Eulalia, L. A. (2020). Simulation in industry 4.0:

A state-of-the-art review. *Computers & Industrial Engineering*, *149*, 106868.

<https://doi.org/10.1016/j.cie.2020.106868>

Del Esposte, A. D. M., Santana, E. F., Kanashiro, L., Costa, F. M., Braghetto, K. R., Lago, N., &

Kon, F. (2019). Design and evaluation of a scalable smart city software platform with

large-scale simulations. *Future Generation Computer Systems*, *93*, 427-441.

<https://doi.org/10.1016/j.future.2018.10.026>

Diaz Ogás, M. G., Fabregat, R., & Aciar, S. (2020). Survey of smart parking systems. *Applied*

Sciences, *10*(11), Article 11. <https://doi.org/10.3390/app10113872>

Duygan, M., Fischer, M., Pärli, R., & Ingold, K. (2022). Where do Smart Cities grow? The spatial

and socio-economic configurations of smart city development. *Sustainable Cities and*

Society, *77*, 103578. <https://doi.org/10.1016/j.scs.2021.103578>

Eiza, M. H., Cao, Y., & Xu, L. (2020). *Toward Sustainable And Economic Smart Mobility: Shaping*

The Future Of Smart Cities. World Scientific.

Elnashar, A. (2019). *IoT evolution towards a super-connected world* (arXiv:1907.02589). arXiv.

<https://doi.org/10.48550/arXiv.1907.02589>

Environmental Implementation Review: Commission helps Member States to better apply EU

environment rules. (s. f.). [Text]. European Commission - European Commission.

Recuperado 21 de marzo de 2023, de

https://ec.europa.eu/commission/presscorner/detail/en/IP_19_1934

Europa Publications (Ed.). (2020). *The Europa Directory of International Organizations 2020* (22.^a ed.). Routledge. <https://doi.org/10.4324/9781003138587>

Grewal, M. S., Andrews, A. P., & Bartone, C. G. (2020). Kalman Filtering. En *Global Navigation Satellite Systems, Inertial Navigation, and Integration* (pp. 355-417). <https://doi.org/10.1002/9781119547860.ch10>

Idris, S., Karunathilake, T., & Förster, A. (2022). Survey and Comparative Study of LoRa-Enabled Simulators for Internet of Things and Wireless Sensor Networks. *Sensors*, 22(15), Article 15. <https://doi.org/10.3390/s22155546>

Institut Cartogràfic i Geològic de Catalunya. (s. f.). Recuperado 22 de mayo de 2023, de <http://www.icgc.cat/>

Instituto Geográfico Nacional. (s. f.). Recuperado 22 de mayo de 2023, de <https://www.ign.es/web/ign/portal/inicio>

IoT ecosystem: 4 key elements. (s. f.). Recuperado 26 de junio de 2023, de <https://www.avsystem.com/blog/iot-ecosystem/>

IoTens: Soluciones IoT, IA y Big Data. (2023). IOTSENS. <https://www.iotsens.com/>

Jemmali, M., Melhim, L. K. B., Alharbi, M. T., Bajahzar, A., & Omri, M. N. (2022). Smart-parking management algorithms in smart city. *Scientific Reports*, 12(1), Article 1. <https://doi.org/10.1038/s41598-022-10076-4>

Khan, A., Aslam, S., Aurangzeb, K., Alhusein, M., & Javaid, N. (2022). Multiscale modeling in smart cities: A survey on applications, current trends, and challenges. *Sustainable Cities and Society*, 78, 103517. <https://doi.org/10.1016/j.scs.2021.103517>

Lebrusán, I., & Toutouh, J. (2021). Car restriction policies for better urban health: A low emission zone in Madrid, Spain. *Air Quality, Atmosphere & Health*, 14(3), 333-342. <https://doi.org/10.1007/s11869-020-00938-z>

Li, D., Ma, J., Cheng, T., van Genderen, J. L., & Shao, Z. (2019). Challenges and opportunities for the development of MEGACITIES. *International Journal of Digital Earth*, 12(12), 1382-1395. <https://doi.org/10.1080/17538947.2018.1512662>

Libelium. (2023). *Libelium*. Libelium ES. <https://www.libelium.com/es/productos-iot/smart-parking/>

LoRaWAN ns-3 module. (2023). [C++]. SIGNET Lab - DEI, University of Padova. <https://github.com/signetlabdei/lorawan> (Obra original publicada en 2017)

Magrin, D., Centenaro, M., & Vangelista, L. (2017). Performance evaluation of LoRa networks in a smart city scenario. *2017 IEEE International Conference on Communications (ICC)*, 1-7. <https://doi.org/10.1109/ICC.2017.7996384>

Malhotra, P., Singh, Y., Anand, P., Bangotra, D. K., Singh, P. K., & Hong, W.-C. (2021). Internet of Things: Evolution, Concerns and Security Challenges. *Sensors*, 21(5), Article 5. <https://doi.org/10.3390/s21051809>

Malik, A. W., Rahman, A. U., Qayyum, T., & Ravana, S. D. (2020). Leveraging Fog Computing for Sustainable Smart Farming Using Distributed Simulation. *IEEE Internet of Things Journal*, 7(4), 3300-3309. <https://doi.org/10.1109/JIOT.2020.2967405>

Miller, M. R., & Newby, D. E. (2020). Air pollution and cardiovascular disease: Car sick. *Cardiovascular Research*, 116(2), 279-294. <https://doi.org/10.1093/cvr/cvz228>

MultiTech Conduit IP67 LoRa Gateway | Outdoor LoRa Deployment. (s. f.). Recuperado 24 de mayo de 2023, de <https://www.multitech.com/brands/multiconnect-conduit-ip67>

Nagowah, S. D., Sta, H. B., & Gobin-Rahimbux, B. A. (2019). An Ontology for an IoT-enabled Smart Parking in a University Campus. *2019 IEEE International Smart Cities Conference (ISC2)*, 474-479. <https://doi.org/10.1109/ISC246665.2019.9071751>

Nagpal, A., & Gabrani, G. (2019). Python for Data Analytics, Scientific and Technical Applications. *2019 Amity International Conference on Artificial Intelligence (AICAI)*, 140-145. <https://doi.org/10.1109/AICAI.2019.8701341>

Network, T. T. (s. f.). *The Things Network*. The Things Network. Recuperado 15 de mayo de 2023, de <https://www.thethingsnetwork.org/>

nsnam. (s. f.). *Ns-3*. Ns-3. Recuperado 31 de mayo de 2023, de <https://www.nsnam.org/>

OpenStreetMap. (s. f.). OpenStreetMap. Recuperado 30 de junio de 2023, de <https://www.openstreetmap.org/>

Orejon-Sanchez, R. D., Crespo-Garcia, D., Andres-Diaz, J. R., & Gago-Calderon, A. (2022). Smart cities' development in Spain: A comparison of technical and social indicators with reference to European cities. *Sustainable Cities and Society*, 81, 103828. <https://doi.org/10.1016/j.scs.2022.103828>

Pasolini, G., Buratti, C., Feltrin, L., Zabini, F., De Castro, C., Verdone, R., & Andrisano, O. (2018). Smart City Pilot Projects Using LoRa and IEEE802.15.4 Technologies. *Sensors*, 18(4), Article 4. <https://doi.org/10.3390/s18041118>

Pelton, J. N. (2019). Smart Cities, Megacities and Meta-Cities. En J. N. Pelton (Ed.), *Preparing for the Next Cyber Revolution: How Our World Will Be Radically Transformed—Again!*

Propuesta de estudio del sistema de aparcamiento tarifado utilizando un modelo de simulación sensorial (pp. 67-77). Springer International Publishing. https://doi.org/10.1007/978-3-030-02137-5_4

Piracha, A., & Chaudhary, M. T. (2022). Urban Air Pollution, Urban Heat Island and Human Health: A Review of the Literature. *Sustainability*, 14(15), Article 15. <https://doi.org/10.3390/su14159234>

Propagation—Model Library. (s. f.). Recuperado 26 de junio de 2023, de <https://www.nsnam.org/docs/models/html/propagation.html>

Resolució 173 de 2015 [Ordenances d'edificació del municipi de Girona]. Por la cual se establecen los parámetros de edificación dentro del municipio de Girona., (2015).

Richardson, J., Sallam, R., Schlegel, K., Kronz, A., & Sun, J. (2020). Magic quadrant for analytics and business intelligence platforms. *Gartner ID G00386610*.

Rousseau, S., & Deschacht, N. (2020). Public Awareness of Nature and the Environment During the COVID-19 Crisis. *Environmental and Resource Economics*, 76(4), 1149-1159. <https://doi.org/10.1007/s10640-020-00445-w>

Ruane, P., Walsh, P., & Cosgrove, J. (2023). Using Simulation Optimization to Improve the Performance of an Automated Manufacturing Line. *Procedia Computer Science*, 217, 630-639. <https://doi.org/10.1016/j.procs.2022.12.259>

Saravanan, K., Julie, E. G., & Robinson, Y. H. (2019). Smart Cities & IoT: Evolution of Applications, Architectures & Technologies, Present Scenarios & Future Dream. En V. E. Balas, V. K. Solanki, R. Kumar, & M. Khari (Eds.), *Internet of Things and Big Data Analytics for Smart Generation* (pp. 135-151). Springer International Publishing. https://doi.org/10.1007/978-3-030-04203-5_7

Savaglio, C., & Fortino, G. (2021). A Simulation-driven Methodology for IoT Data Mining Based on Edge Computing. *ACM Transactions on Internet Technology (TOIT)*, 21(2). <https://doi.org/10.1145/3402444>

Sede Electrónica del Catastro—Inicio. (s. f.). Recuperado 3 de junio de 2023, de <https://www.sedecatastro.gob.es/>

Semtech LoRa Technology Overview | Semtech. (2023). <https://www.semtech.com/lora>

Stusek, M., Moltchanov, D., Masek, P., Mikhaylov, K., Zeman, O., Roubicek, M., Koucheryavy, Y., & Hosek, J. (2020). Accuracy Assessment and Cross-Validation of LPWAN Propagation Models in Urban Scenarios. *IEEE Access*, 8, 154625-154636. <https://doi.org/10.1109/ACCESS.2020.3016042>

Sugiyama, T., Chandrabose, M., Homer, A. R., Sugiyama, M., Dunstan, D. W., & Owen, N. (2020). Car use and cardiovascular disease risk: Systematic review and implications for transport research. *Journal of Transport & Health*, 19, 100930. <https://doi.org/10.1016/j.jth.2020.100930>

Thombre, A., & Agarwal, A. (2021). A paradigm shift in urban mobility: Policy insights from travel before and after COVID-19 to seize the opportunity. *Transport Policy*, *110*, 335-353. <https://doi.org/10.1016/j.tranpol.2021.06.010>

Turner, J., Shahrman, A., Harun, A., Hashim, M., Razlan, Z., Ndzi, D., Ismail, R., Murad, S., Isa, M., & Mohyar, S. (2021). Signal Propagation Modelling for Vehicle-To-Infrastructure Communication Under the Influence of Metal Obstruction. *Int. J. Nanoelectron. Mater*, *14*, 127-139.

UNE 178201:2016 Ciudades inteligentes. Definición, atributos y... (s. f.). Recuperado 21 de marzo de 2023, de <https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0056504>

Winext: Un proveedor de soluciones integrales para el Internet de las cosas. (2023). https://www-winext-cn.translate.google/?_x_tr_sch=http&_x_tr_sl=zh-CN&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=sc

Worldwide Interoperability for SEmantics IoT | Wise-IoT Project | Fact Sheet | H2020 | CORDIS | European Commission. (2022). <https://cordis.europa.eu/project/id/723156>

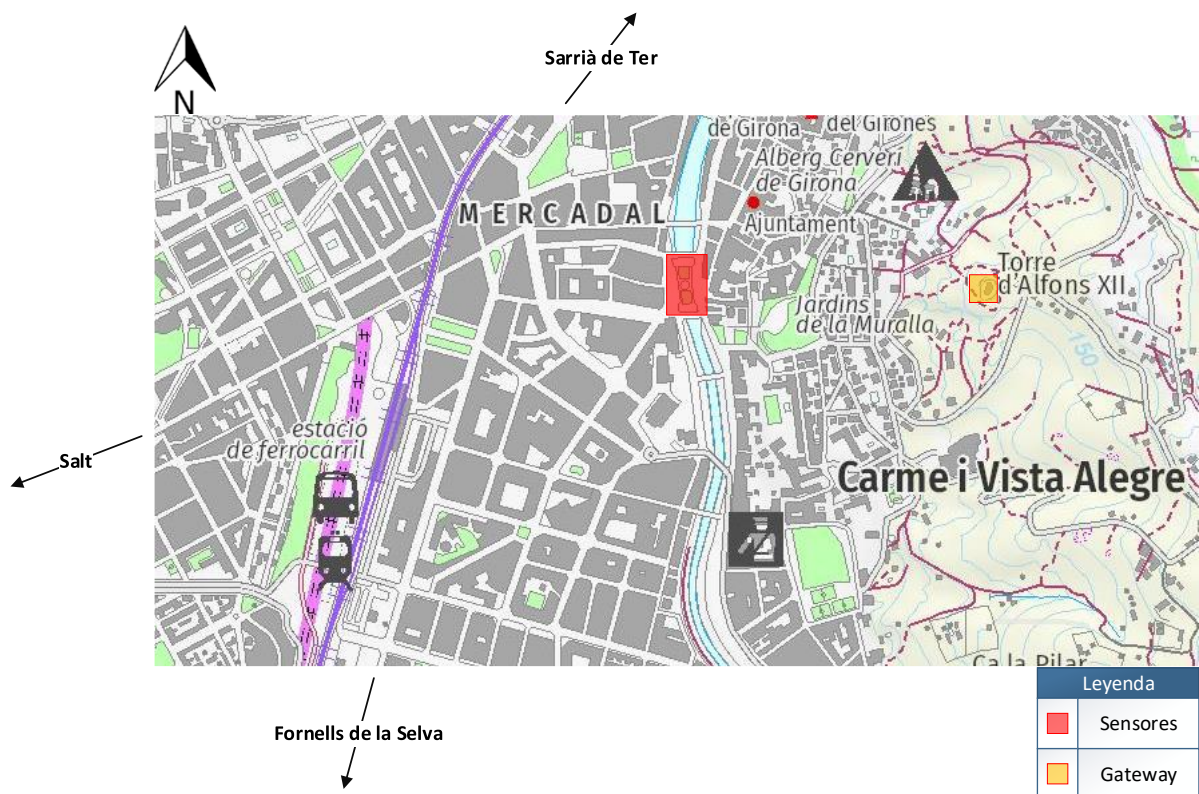
Zhao, P., & Hu, H. (2019). Geographical patterns of traffic congestion in growing megacities: Big data analytics from Beijing. *Cities*, *92*, 164-174. <https://doi.org/10.1016/j.cities.2019.03.022>

Anexo A. Planos

En este anexo se encuentran algunos de los planos empleados para este proyecto. A continuación, se muestran los planos de Situación y Emplazamiento, en este segundo se separa individualmente para los sensores y la pasarela.

A.1. Situación

Figura 37. Plano de situación para el proyecto de estacionamiento tarifado de Plaza Catalunya.



Fuente: Elaboración propia junto a *Institut Cartogràfic i Geològic de Catalunya*, s. f.

Anexo B. Datos de implementación

En el siguiente anexo se encuentran algunos datos en formato de tablas usados en las distintas fases del proyecto.

B.1. Ubicaciones de los dispositivos

Tabla 12. *Ubicación de los dispositivos en Plaza Catalunya.*

Dispositivos	Num.º	Latitud	Longitud
Sensores	0	41,981988	2,823137
	1	41,981842	2,823145
	2	41,981885	2,823154
	3	41,981838	2,823163
	4	41,981810	2,823169
	5	41,981776	2,823178
	6	41,981747	2,823182
	7	41,981716	2,823187
	8	41,981623	2,823207
	9	41,981593	2,823211
	10	41,981563	2,823218
	11	41,981531	2,823220
	12	41,981497	2,823232
	13	41,981469	2,823232
	14	41,981440	2,823236
	15	41,982012	2,823561
	16	41,981979	2,823557
	17	41,981947	2,823557
	18	41,981916	2,823553
	19	41,981884	2,823549
	20	41,981852	2,823548
	21	41,981816	2,823549
	22	41,981789	2,823548
	23	41,981758	2,823554
	24	41,981727	2,823560
	25	41,981619	2,823580
	26	41,981590	2,823585
	27	41,981557	2,823589
	28	41,981525	2,823596
	29	41,981490	2,823606
	30	41,981460	2,823607
	31	41,981431	2,823615
32	41,981398	2,823619	
Gateway	33	41,981683	2,830186

Fuente: Elaboración propia.

B.2. Métricas y características de los dispositivos

Tabla 13. Características de los sensores candidatos propuestos por el Ayuntamiento.

Subministrador	Modelo	IP	Instalación	FOTA	Calibración	Modelo Bat.	Intercambiable	Sensor	Chip Semtech(LoRa)
IOTSens	AN-101D Winext	67	Superficie	No	APP/Dongle (Bluetooth)	Li-SOCI2	Sí	Magnetómetro	SX1276
Bosch	Parking Lot Sensor	67	Superficie	Sí	MAN. (Reset Magnético)	Li-SOCI2	No	Radar y Magnético	--
Libelium	Model Radar/Mag Dual	68	Superficie	Sí	AUTO (Reset Magnético)	Li-SOCI2	Sí	Radar y Magnético	--

Fuente: Winext s. f., IoTsens, s. f., Parking Lot Sensor, s. f., Libelium, s. f.

Tabla 14. Métricas de los sensores candidatos propuestos por el Ayuntamiento.

Modelo	Alimentación (Vcc)	Payload (bytes)	SF	Rango Detección (m)	T. Detección (s)	Capacidad (mA·h)	Auto. Min. (años)	Mensajes/Día
Winext	3,3	11	7–12	0,10 – 1	20	16.000	4	20
Bosch	3,3	1	7–12	0,10 – 1	40-60	1.200	5	28
Libelium	3,3	11	7–12	0,20 – 1	16	10.400	4	--

Fuente: Winext s. f., IoTsens, s. f., Parking Lot Sensor, s. f., Libelium, s. f.

Tabla 15. Características de la Gateway propuesta por el Ayuntamiento.

Fabricante	Modelo	IP	Instalación	Comunicaciones	Alimentación	Chip Semtech(LoRa)	Canales	Antena (dBi)	Max. EIRP (dBm)
MultiTech	MTCDTIP-266A-868	67	Outdoors	LoRa/PoE	PoE (RJ45)	SX1301	8	3	14 - 27

Fuente: MultiTech Conduit IP67 LoRa Gateway | Outdoor LoRa Deployment, s. f..

Tabla 16. *Payload dispositivo Winext.*

Byte/Bit	7	6	5	4	3	2	1	0
0	Tipo de sensor							
1	Frame				Reservado			Up./Downlink
2	Eje X de calibración							
3								
4	Eje Y de calibración							
5								
6	Eje Z de calibración							
7								
8	Temperatura							
9								
10	Detección V.	Voltaje Batería						

Fuente: *Winext*, s. f.

Tabla 17. *Payload dispositivo BOSCH.*

Byte/Bit	7	6	5	4	3	2	1	0
0	Reservado							Detección V.

Fuente: *BOSCH*, s. f.

Tabla 18. *Payload dispositivo Libelium.*

Byte/Bit	7	6	5	4	3	2	1	0
0	Header 1							
1	Header 2							
2	Error del sensor							
3								
4	Temperatura							
5								
6	Timestamp (horas y minutos)							
7								
8	Distancia							
9	Amplitud							
10	Numero de reflexiones							

Fuente: *Libelium*, s. f.

B.3. Datos de las simulaciones

En este anexo se encuentran distintas tablas con datos empleados tanto para la simulación como los resultantes de estos.

Tabla 19. *Sensibilidades y Duty Cycle de los dispositivos según Payload de Winext.*

SF	GW (dBm)	ED (dBm)	1% (msg/h)	1% (s)	T. on Air (ms)
7	-130	-124	583	6,2	61,7
8	-132,5	-127	318	11,3	113,2
9	-135	-130	174	20,6	205,8
10	-137,5	-133	97	37,1	370,7
11	-140	-135	43	82,3	828,3
12	-142,5	-137	24	148,3	1.428,80

Fuente: Elaboración propia.

Tabla 20. *Canales de comunicación LoRaWAN.*

Sub-Bands (MHz)	Channel
867,1	Chnl_1
867,3	Chnl_2
867,5	Chnl_3
867,7	Chnl_4
867,9	Chnl_5
868,1	Chnl_6
868,3	Chnl_7
868,5	Chnl_8

Fuente: Elaboración propia.

Tabla 21. *Periodos de tiempo empleados para las simulaciones.*

Periodos (s)
20
180
360
540
720
900

Fuente: Elaboración propia.

Tabla 22. Set de datos (Dataset) generado por el digestor de datos de Python.

Nombre	Unidades	Tipo	Descripción
Time	s	Real	Tiempo de emisión del mensaje
ED	0	Entero	Número del dispositivo
SF	0	Entero	Factor de dispersión del dispositivo
Tx	dBm	Real	Potencia de la señal enviada
Rx	dBm	Real	Potencia de la señal recibida
SNR	dB	Real	Relación señal/ruido del mensaje
Dst	M	Real	Distancia del ED respecto la GW
Delay	ns	Real	Retraso en la siguiente transmisión
ED_Packet	0	Texto	Identificador del paquete enviado
Received	0	Entero	Indicador, la GW recibió el mensaje
Time_Received	s	Real	Tiempo en que la GW recibió el mensaje
ACK_Received	0	Entero	Indicador, ED recibió el ACK de la GW
Msg_Before_ACK	0	Entero	Número de repeticiones del ciclo ACK
Sub-Band	MHz	Real	Sub-banda por la cual se envía el mensaje
Latitude	º	Real	Latitud del ED
Longitude	º	Real	Longitud del ED

Fuente: Elaboración propia.

Tabla 23. Valores PER-Periodo del primer escenario.

Periodos	SF7_1	SF8_1	SF9_1	SF10_1	SF11_1	SF12_1	SF7_1_C	SF8_1_C	SF9_1_C	SF10_1_C	SF11_1_C	SF12_1_C
20	0,0212	0,0392	0,072	0,1068	0,1608	0,3004	0,03147	0,05087	0,08186	0,114946	0,16619	0,303236
180	0,0038	0,0038	0,0038	0,0143	0,0356	0,0554	0,00909	0,00707	0,00657	0,016288	0,038763	0,065278
360	0	0,0043	0,0043	0,0043	0,0179	0,0316	0,00455	0,00783	0,00758	0,007576	0,020707	0,035606
540	0	0,0038	0,0049	0,0049	0,011	0,0212	0,00417	0,00417	0,0072	0,007197	0,014015	0,022727
720	0	0	0,0051	0,0051	0,0051	0,0192	0,00455	0,00101	0,00657	0,006566	0,006566	0,018687
900	0	0	0,0057	0,0057	0,0057	0,0177	0,00568	0,00126	0,00505	0,007576	0,007576	0,018939

Fuente: Elaboración propia.

Tabla 24. Valores PER-Periodo del segundo escenario.

Periodos	SF7_2	SF8_2	SF9_2	SF10_2	SF11_2	SF12_2	SF7_2_C	SF8_2_C	SF9_2_C	SF10_2_C	SF11_2_C	SF12_2_C	ADR_ACK	ADR_ACK_C
20	0,1621	0,2228	0,322	0,3307	0,4364	0,5926	0,16724	0,21666	0,352304	0,331403	0,429339	0,592604	0,23590046	0,27276994
180	0,1546	0,0942	0,1386	0,1588	0,3094	0,39388	0,15133	0,09368	0,138651	0,158827	0,283791	0,356436	0,15497879	0,20467909
360	0,0474	0,0694	0,139	0,1531	0,2037	0,34914	0,04972	0,06765	0,140593	0,146843	0,20936	0,300876	0,05689981	0,1509272
540	0,0741	0,0958	0,1467	0,1736	0,2101	0,23303	0,07484	0,09759	0,142202	0,174109	0,207488	0,239933	0,08796562	0,16442688
720	0,0513	0,0513	0,0473	0,063	0,1163	0,23288	0,05523	0,05353	0,049057	0,060878	0,104987	0,226801	0,07416463	0,11581811
900	0,0787	0,0767	0,0977	0,1212	0,1426	0,18865	0,08069	0,07668	0,100404	0,121148	0,138771	0,188542	0,08803729	0,12320633

Fuente: Elaboración propia.

Tabla 25. Promedios del Rx y SNR de los ED con y sin vehículo.

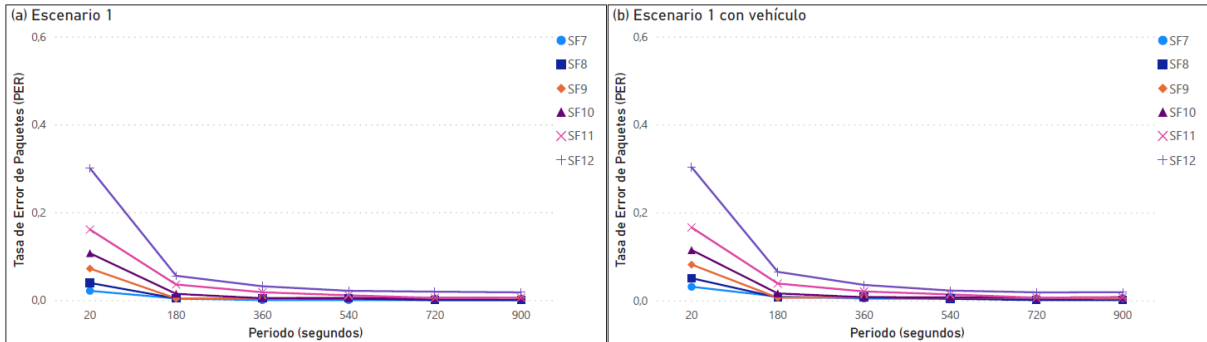
ED	SF7_Rx	SF7_SNR	SF7_Rx_C	SF7_SNR_C	Latitude	Longitude
0	-103,963443	13,0674567	-110,509567	6,52133274	41,981988	2,823137
1	-110,904832	6,12606746	-110,760166	6,27073367	41,981942	2,823145
2	-113,297477	3,73342334	-105,562401	11,4684993	41,981885	2,823154
3	-93,7351882	23,2957116	-113,16195	3,86894941	41,981838	2,823163
4	-113,086113	3,94478691	-114,68597	2,3449295	41,98181	2,823169
5	-107,715183	9,31571654	-113,849352	3,18154756	41,981776	2,823178
6	-108,002979	9,0279207	-109,779265	7,25163506	41,981747	2,823182
7	-105,484694	11,5462059	-109,239197	7,79170306	41,981716	2,823187
8	-101,569215	15,461685	-105,887291	11,1436093	41,981623	2,823207
9	-66,5499219	50,480978	-75,5198588	41,5110411	41,981593	2,823211
10	-68,7806493	48,2502506	-77,8616478	39,1692521	41,981563	2,823218
11	-105,348374	11,6825263	-110,353185	6,6777145	41,981531	2,82322
12	-107,316163	9,71473691	-106,48218	10,5487198	41,981497	2,823232
13	-103,499896	13,531004	-118,724279	-1,69337883	41,981469	2,823232
14	-106,00747	11,0234295	-111,699149	5,3317508	41,98144	2,823236
15	-113,319824	3,7110758	-105,714085	11,3168145	41,982012	2,823561
16	-100,434611	16,5962891	-104,454586	12,576314	41,981979	2,823557
17	-104,801992	12,2289082	-113,12032	3,91057996	41,981947	2,823557
18	-103,047396	13,9835036	-114,34272	2,6881795	41,981916	2,823553
19	-107,056869	9,97403089	-107,692898	9,33800144	41,981884	2,823549
20	-106,141146	10,8897536	-119,8968	-2,86589967	41,981852	2,823548
21	-107,208241	9,82265867	-109,924719	7,10618135	41,981816	2,823549
22	-105,534268	11,4966318	-108,636298	8,394602	41,981789	2,823548
23	-97,4443622	19,5865376	-109,293562	7,73733783	41,981758	2,823554
24	-103,325719	13,7051804	-111,557685	5,47321468	41,981727	2,82356
25	-97,5944631	19,4364368	-115,778833	1,252067	41,981619	2,82358
26	-99,2553922	17,7755076	-108,465637	8,5652633	41,98159	2,823585
27	-103,838091	13,1928091	-112,30321	4,72768982	41,981557	2,823589
28	-105,029263	12,0016369	-118,062134	-1,03123393	41,981525	2,823596
29	-94,8861352	22,1447646	-112,214624	4,8162758	41,98149	2,823606
30	-97,6920937	19,3388062	-110,901873	6,12902672	41,98146	2,823607
31	-106,259977	10,770923	-104,273784	12,7571157	41,981431	2,823615
32	-100,905188	16,125712	-112,726244	4,30465543	41,981398	2,823619

Fuente: Elaboración propia.

Anexo C. Visualizaciones y gráficos

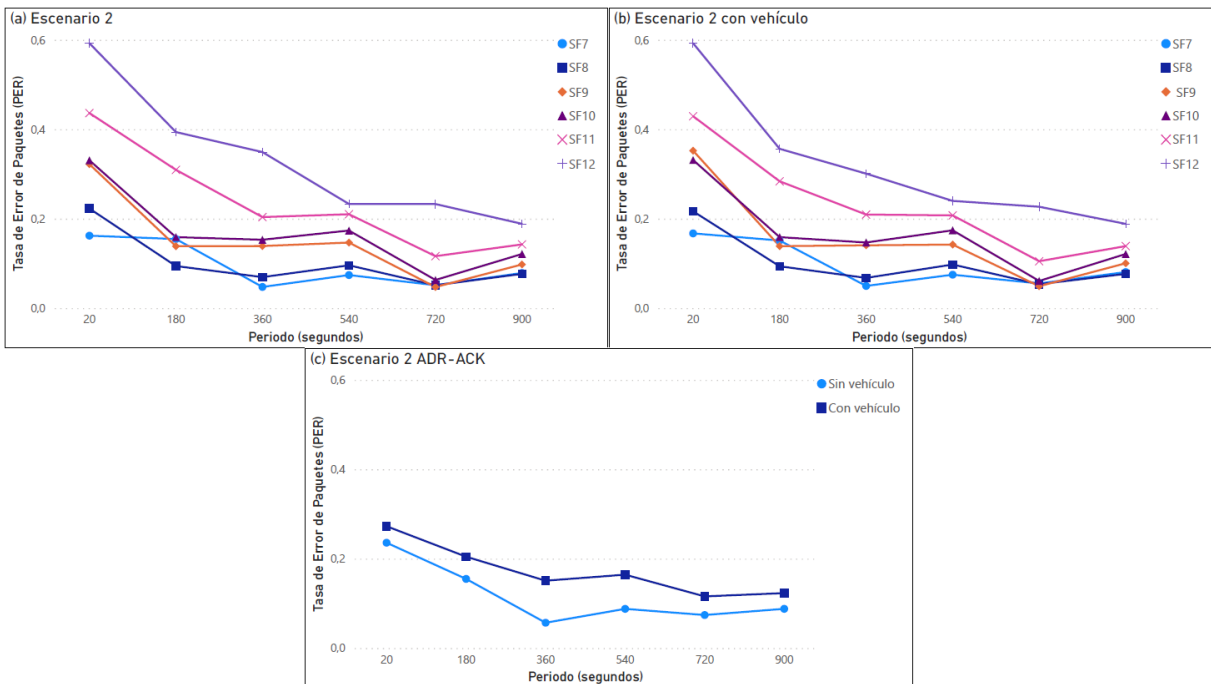
En este anexo se pueden visualizar al completo los distintos gráficos e infografías que se han desarrollado para los análisis de los distintos escenarios de este proyecto.

Figura 38. Conjunto de gráficos comparativos PER según Periodo para el primer escenario.



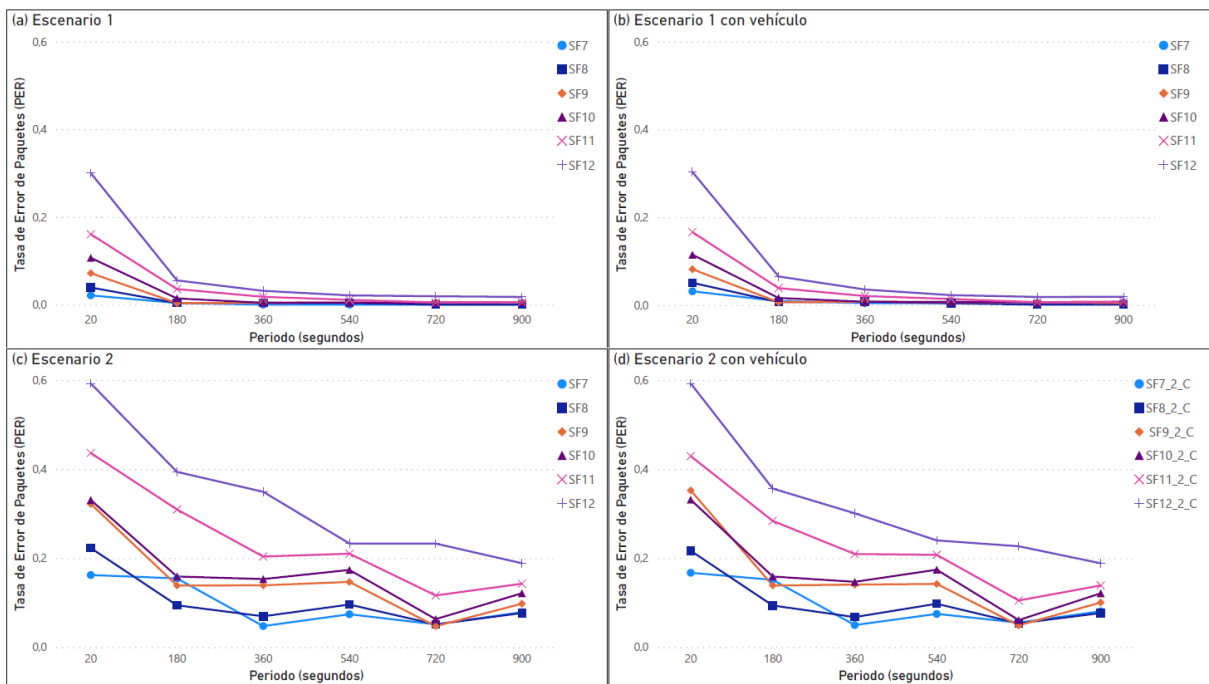
Fuente: Elaboración propia.

Figura 39. Conjunto de gráficos comparativos PER según Periodo para el segundo escenario.



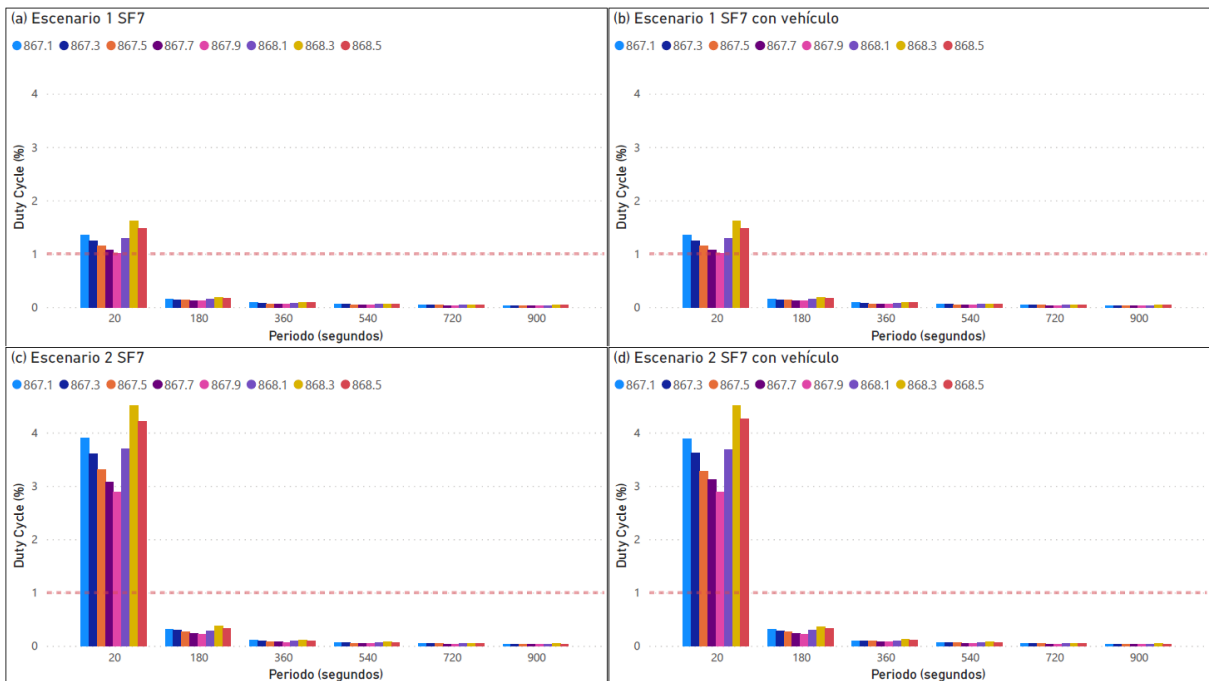
Fuente: Elaboración propia.

Figura 40. Comparativos del PER según Periodo para el primer y segundo escenario.



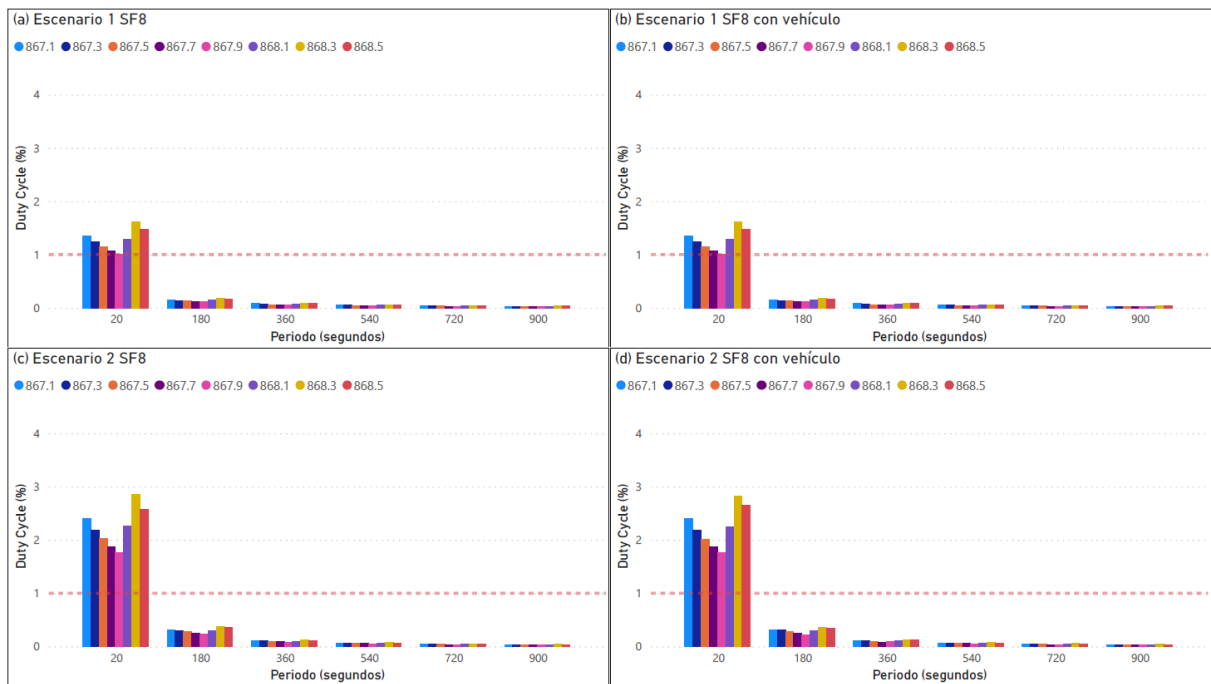
Fuente: Elaboración propia.

Figura 41. Comparativo del DC según Periodo y SF7 para el primer y segundo escenario.



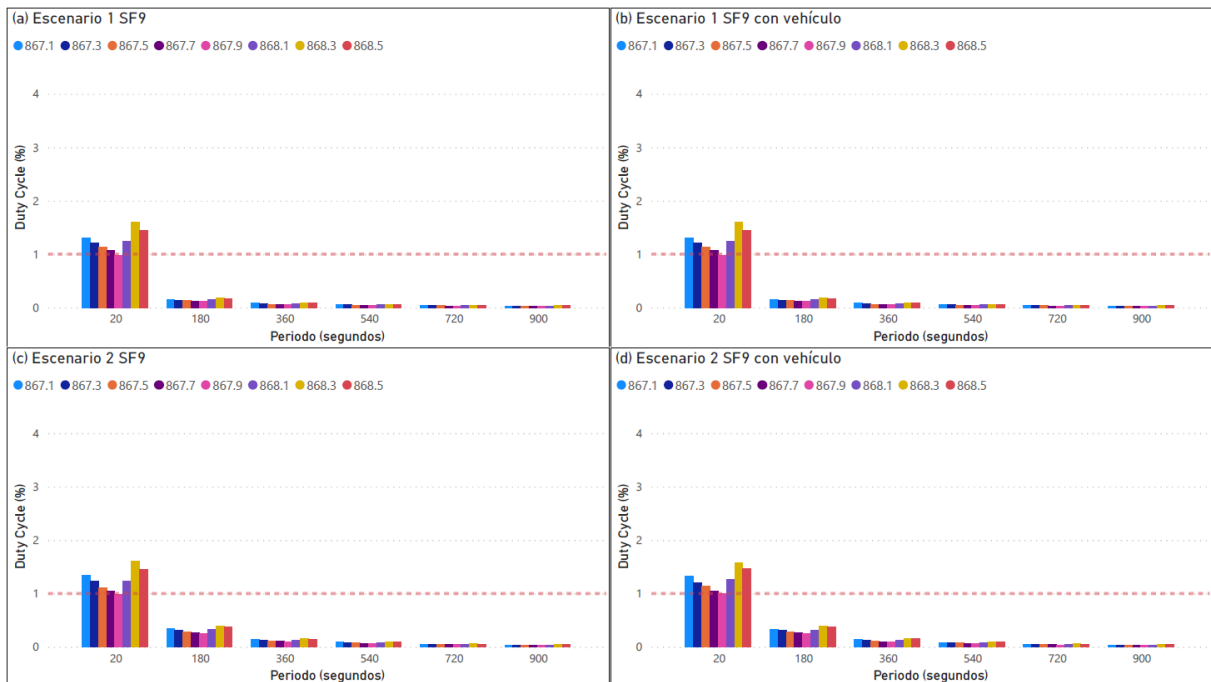
Fuente: Elaboración propia.

Figura 42. Comparativo del DC según Periodo y SF8 para el primer y segundo escenario.



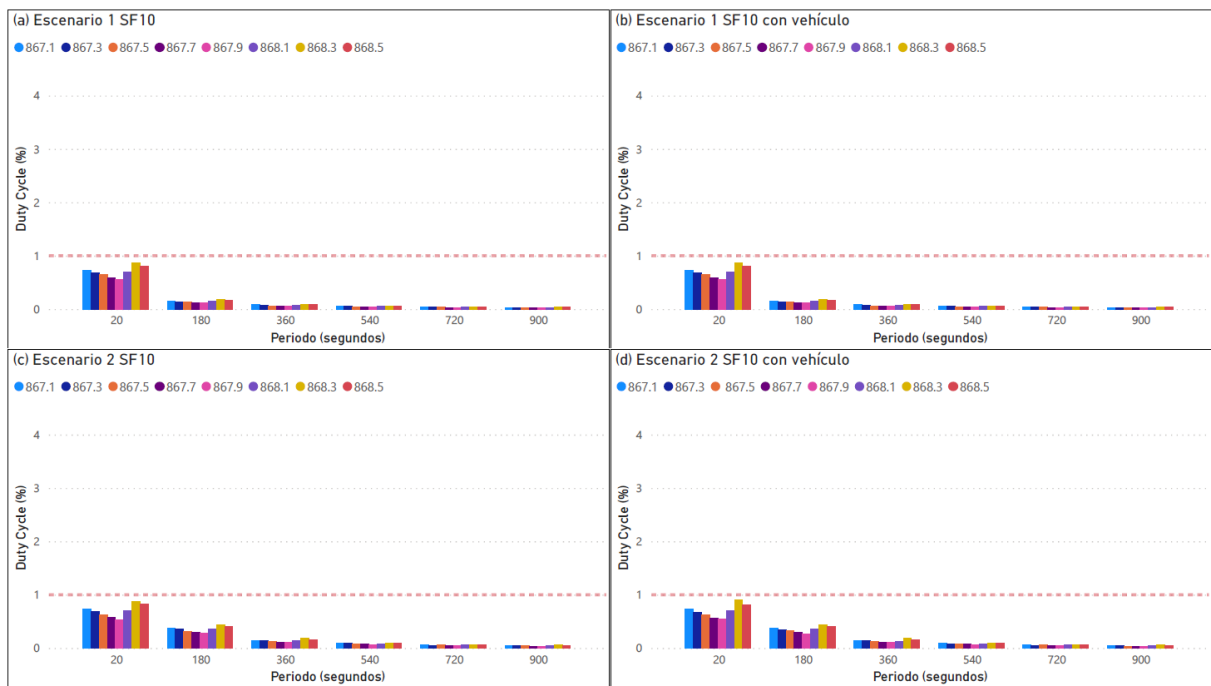
Fuente: Elaboración propia.

Figura 43. Comparativo del DC según Periodo y SF9 para el primer y segundo escenario.



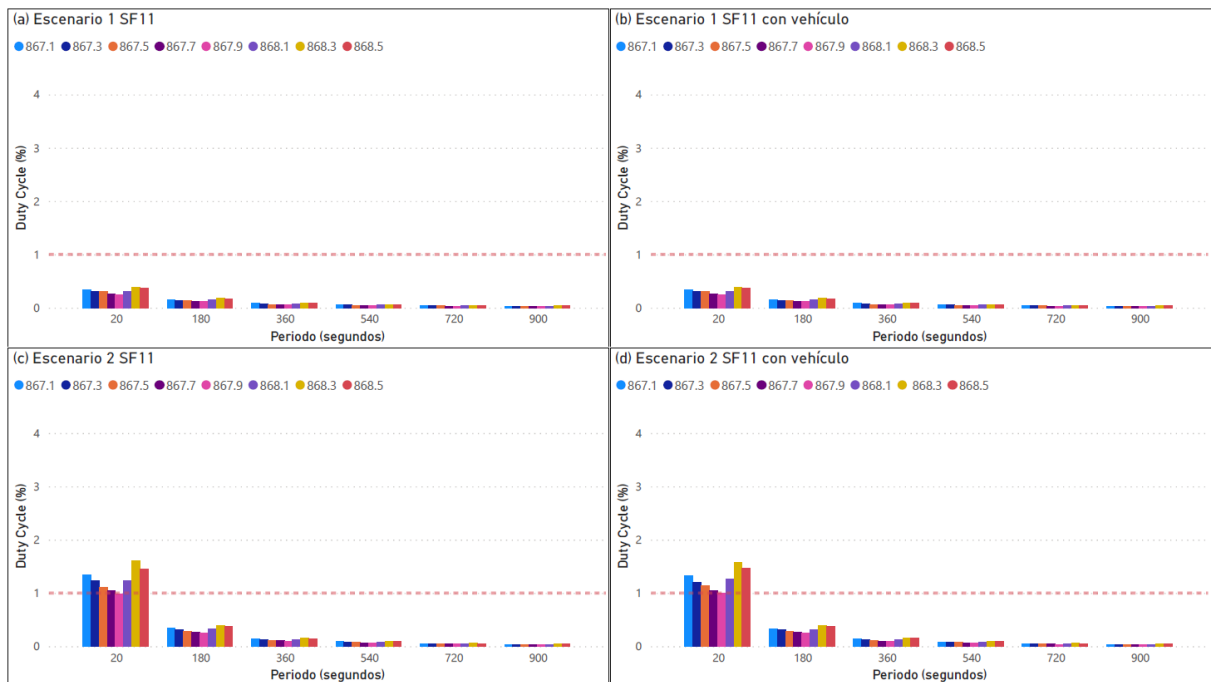
Fuente: Elaboración propia.

Figura 44. Comparativo del DC según Periodo y SF10 para el primer y segundo escenario.



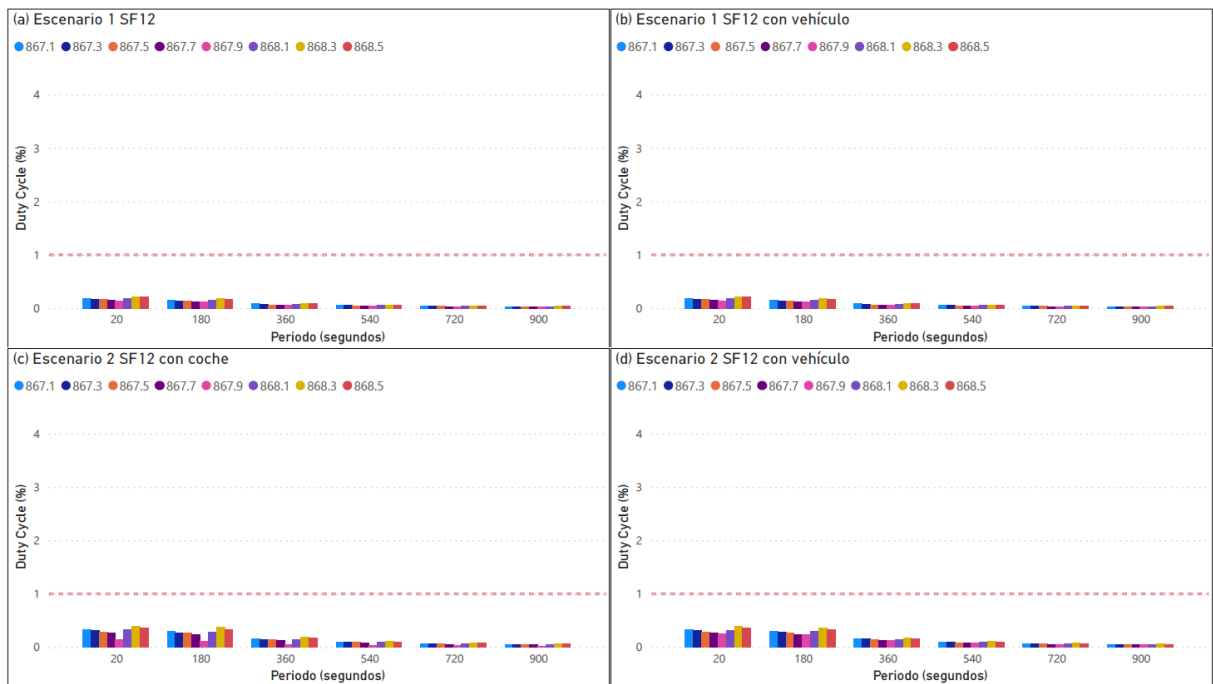
Fuente: Elaboración propia.

Figura 45. Comparativo del DC según Periodo y SF11 para el primer y segundo escenario.



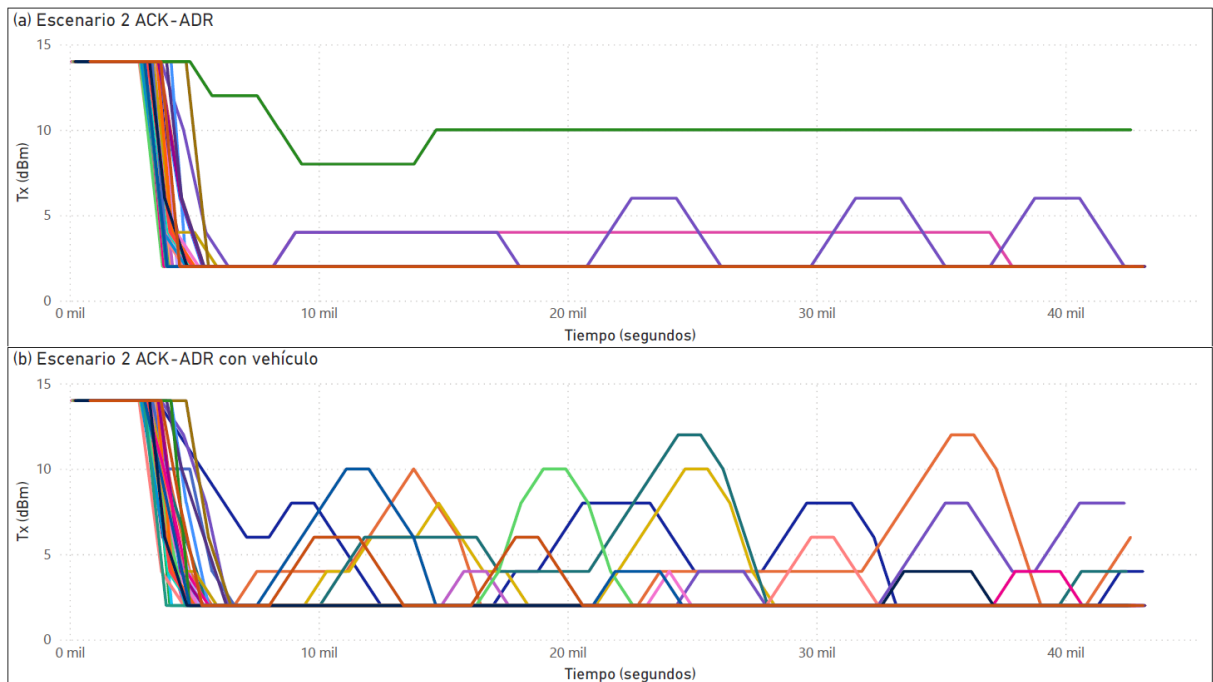
Fuente: Elaboración propia.

Figura 46. Comparativo del DC según Periodo y SF12 para el primer y segundo escenario.



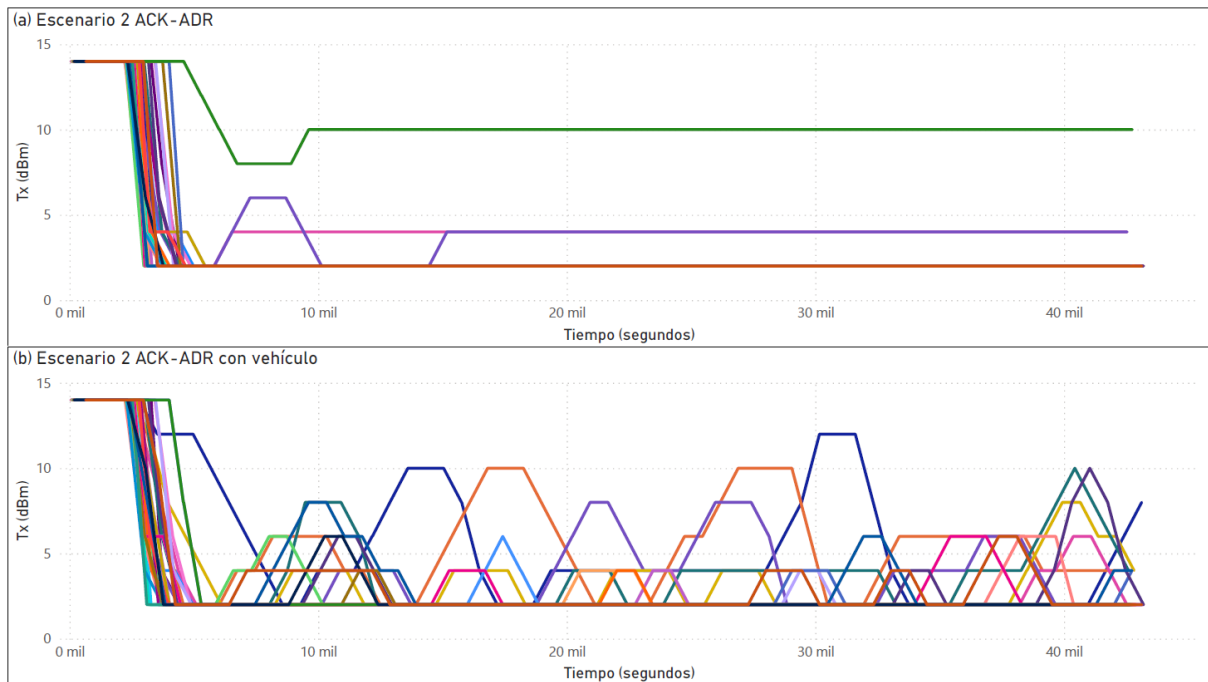
Fuente: Elaboración propia.

Figura 47. Variación del Tx según funcionalidad ADR con periodo de 900 segundos.



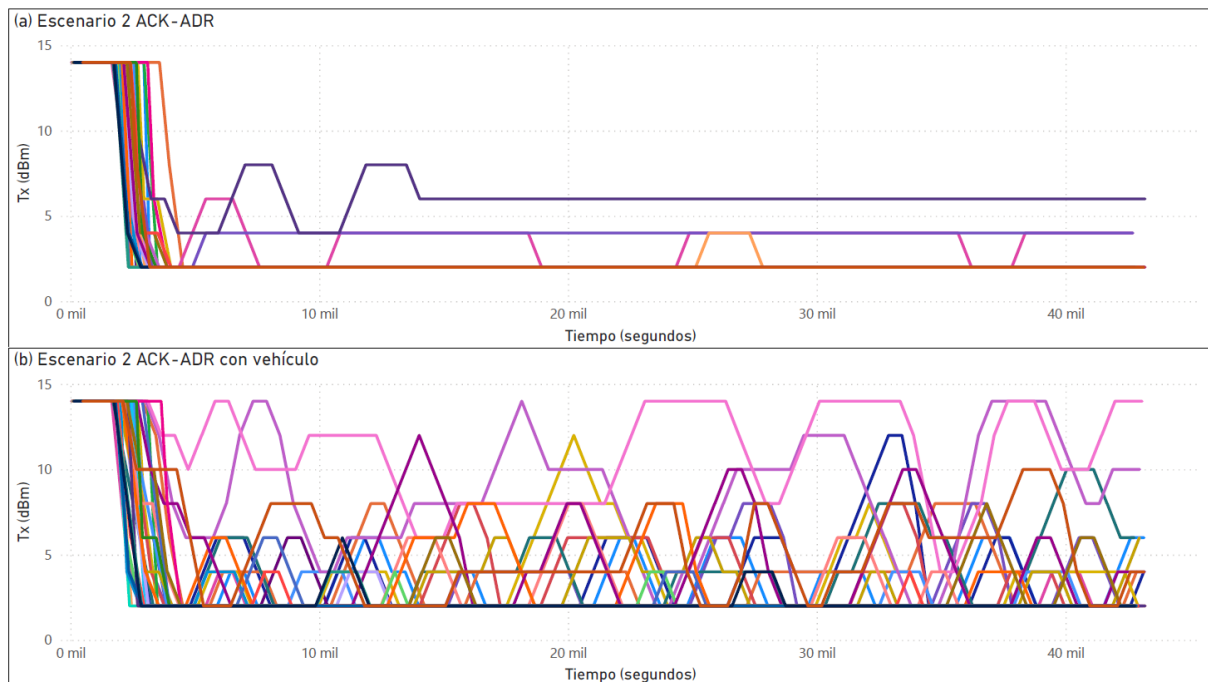
Fuente: Elaboración propia.

Figura 48. Variación del Tx según funcionalidad ADR con periodo de 720 segundos.



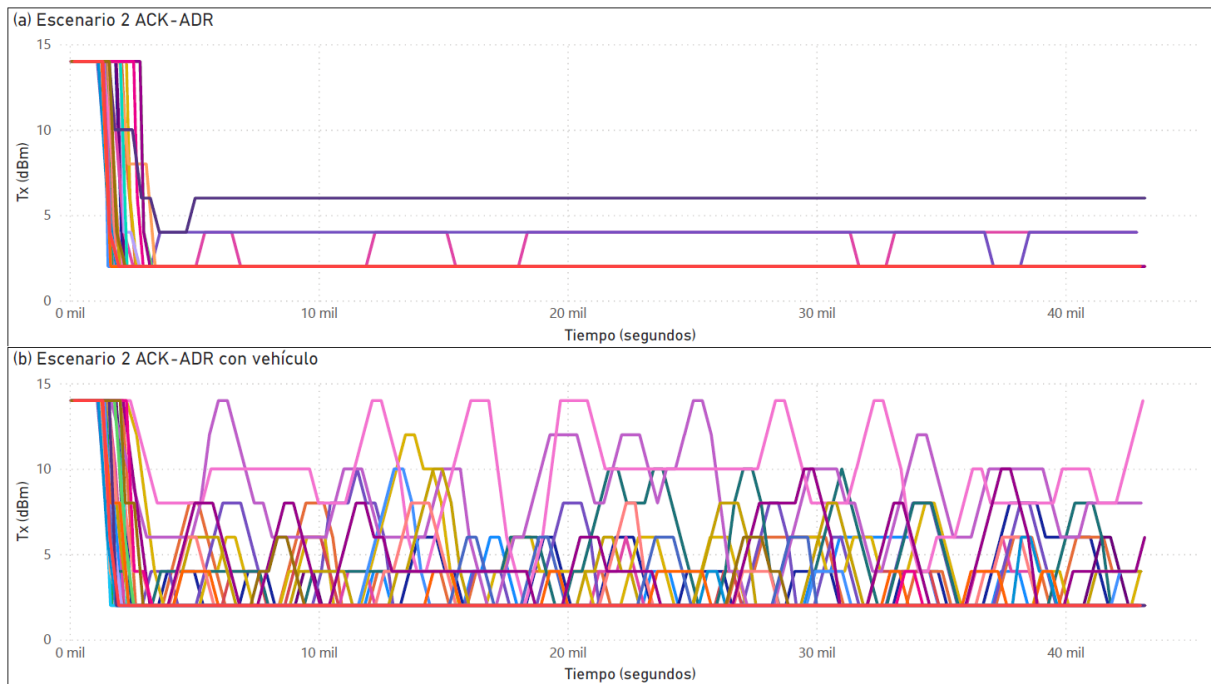
Fuente: Elaboración propia.

Figura 49. Variación del Tx según funcionalidad ADR con periodo de 540 segundos.



Fuente: Elaboración propia.

Figura 50. Variación del Tx según funcionalidad ADR con periodo de 360 segundos.



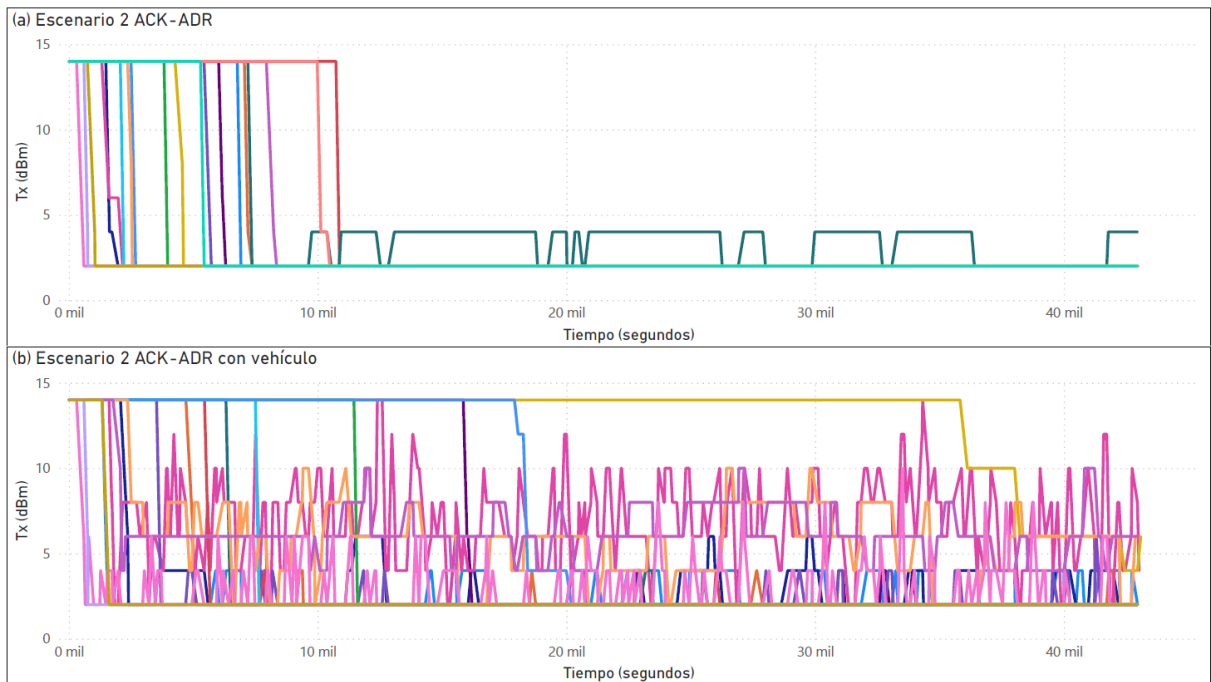
Fuente: Elaboración propia.

Figura 51. Variación del Tx según funcionalidad ADR con periodo de 180 segundos.



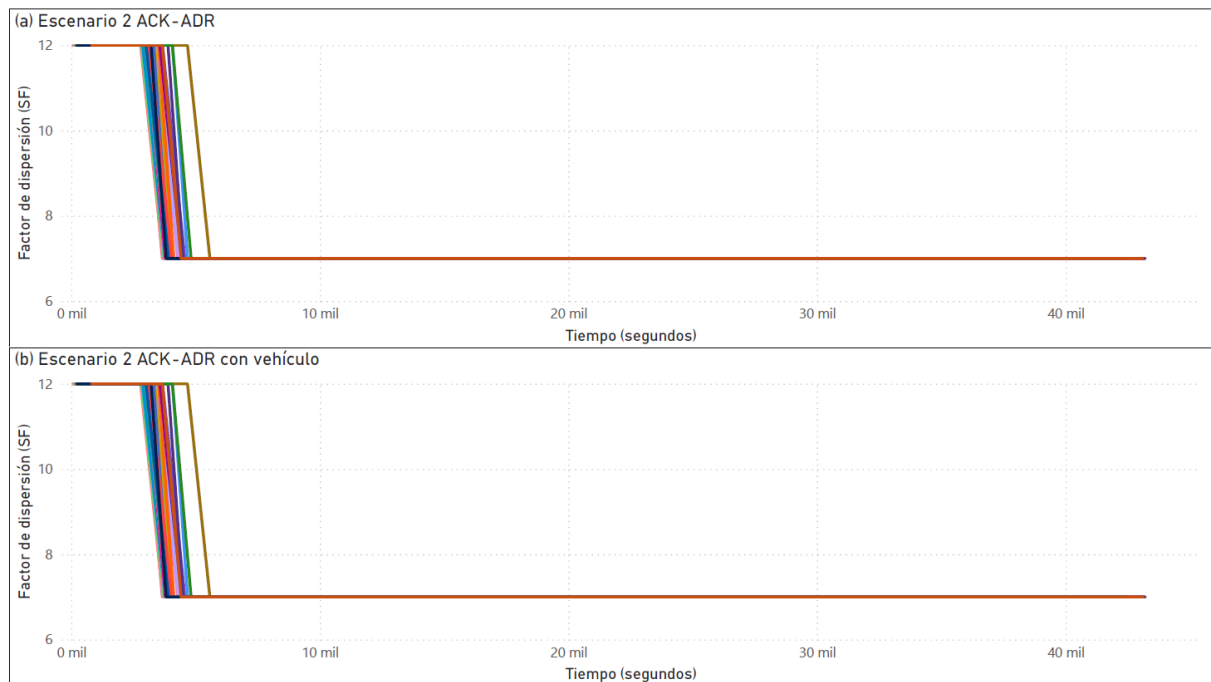
Fuente: Elaboración propia.

Figura 52. Variación del Tx según funcionalidad ADR con periodo de 20 segundos.



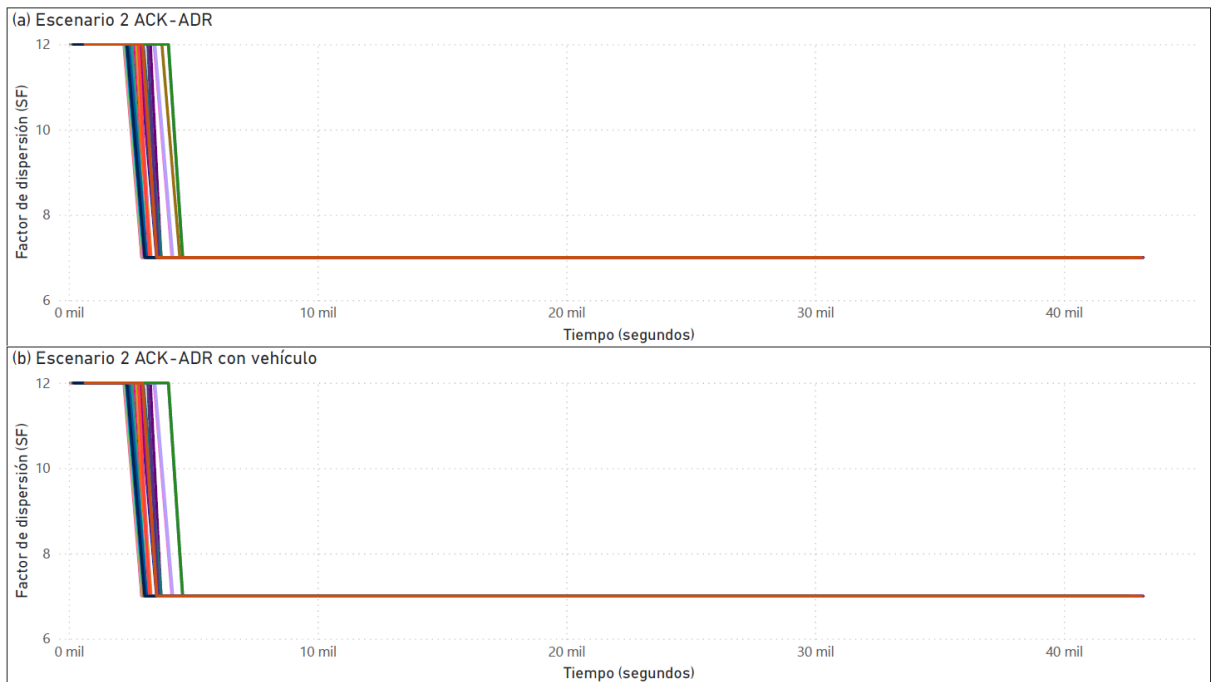
Fuente: Elaboración propia.

Figura 53. Variación del SF según funcionalidad ADR con periodo de 900 segundos.



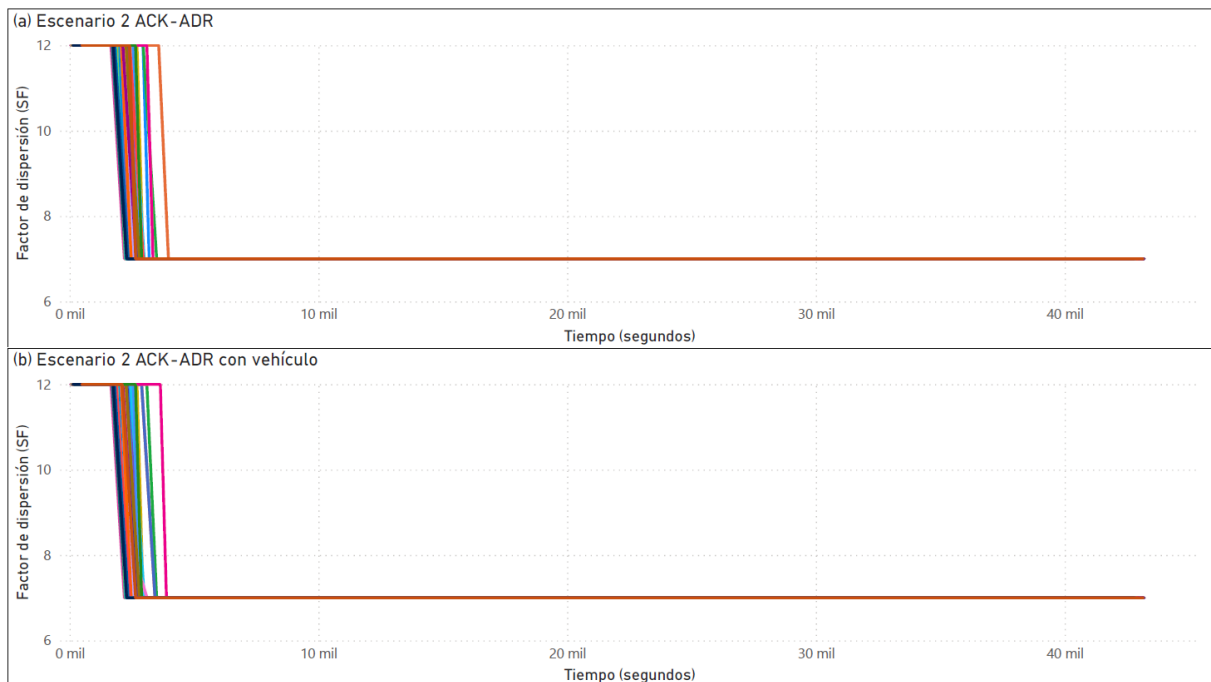
Fuente: Elaboración propia.

Figura 54. Variación del SF según funcionalidad ADR con periodo de 720 segundos.



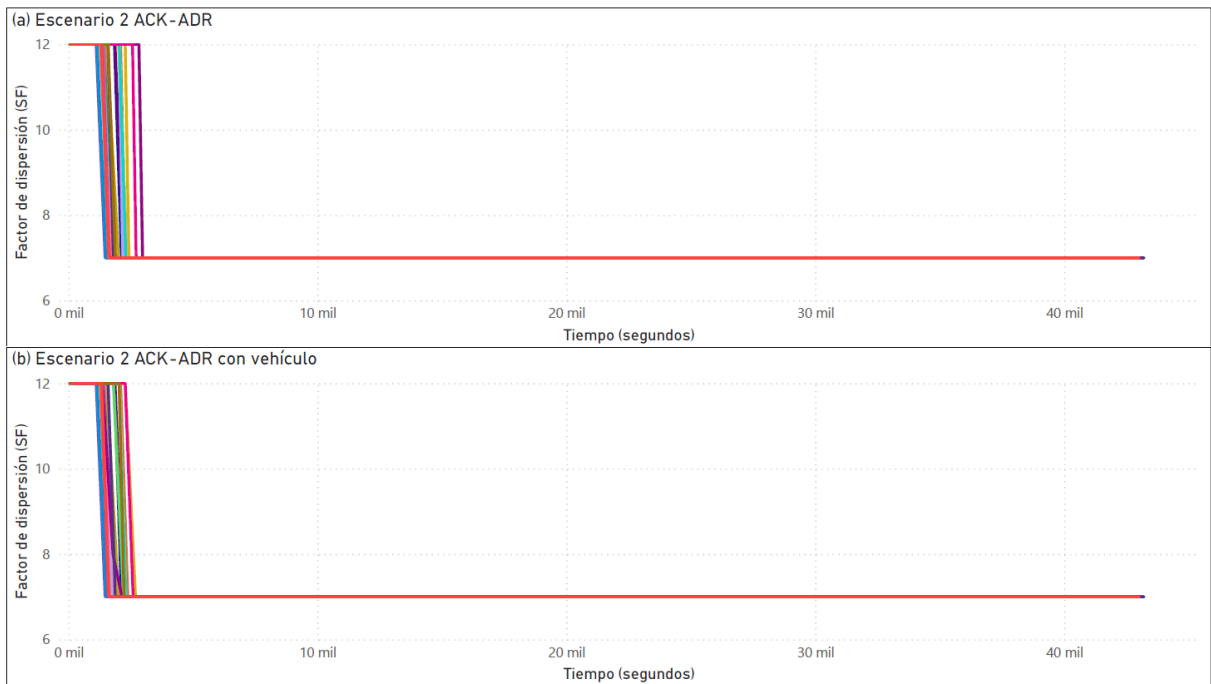
Fuente: Elaboración propia.

Figura 55. Variación del SF según funcionalidad ADR con periodo de 540 segundos.



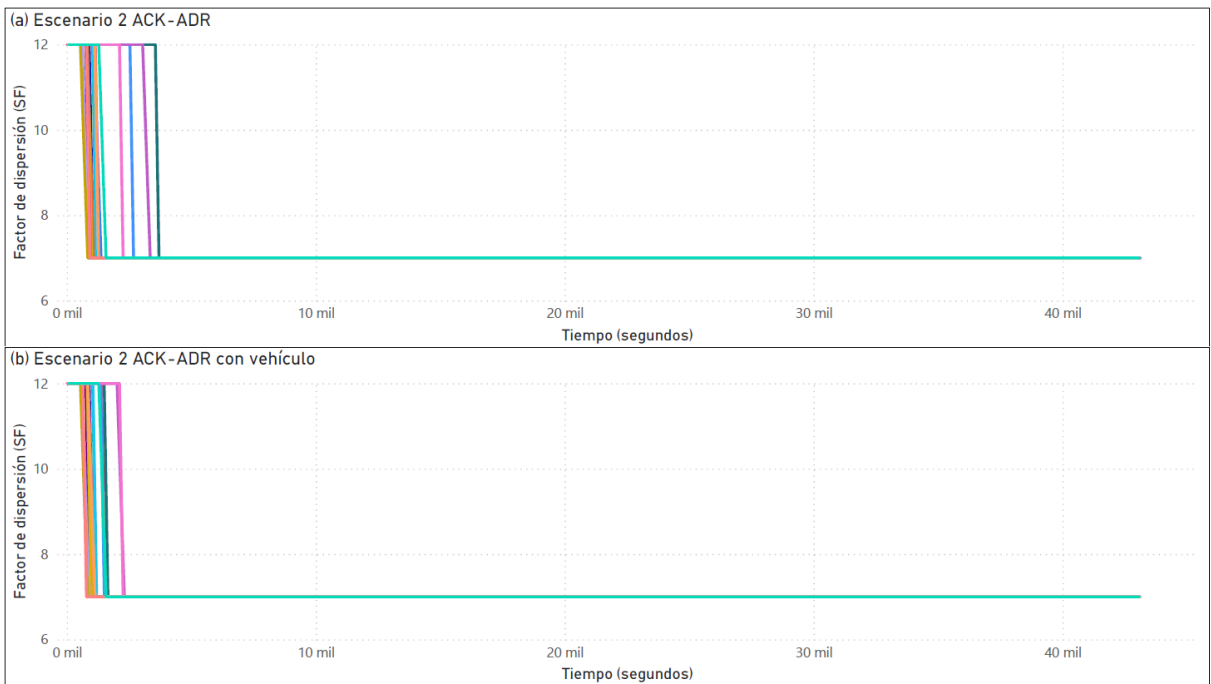
Fuente: Elaboración propia.

Figura 56. Variación del SF según funcionalidad ADR con periodo de 360 segundos.



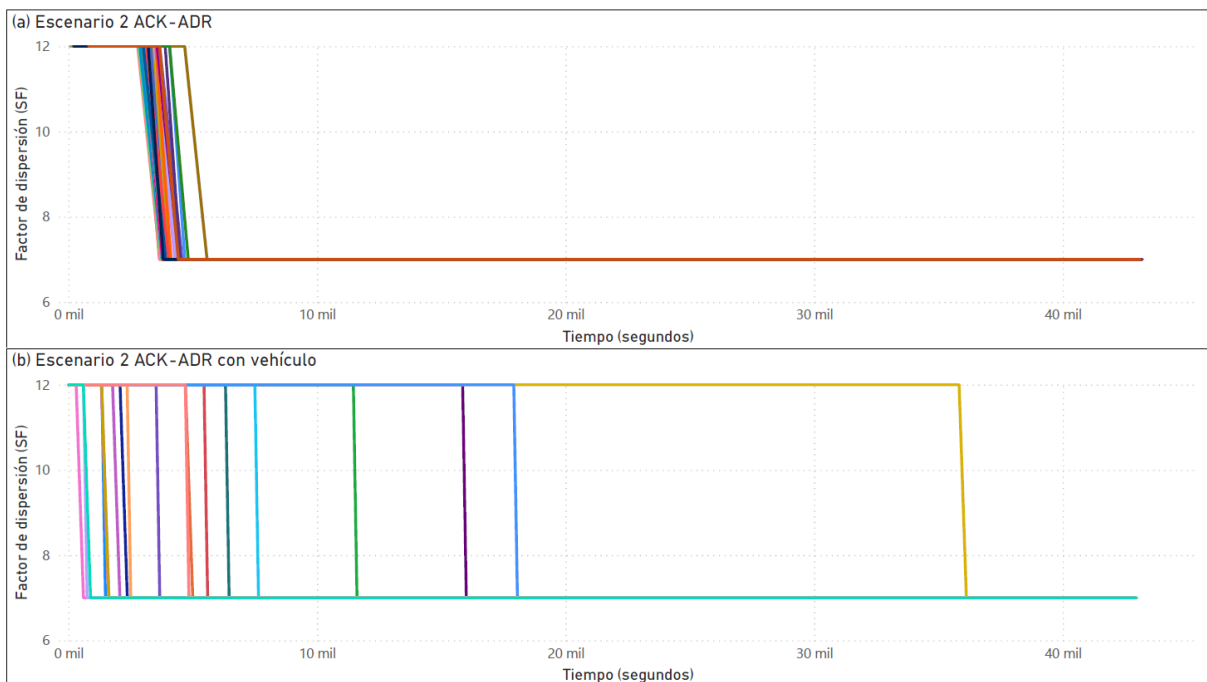
Fuente: Elaboración propia.

Figura 57. Variación del SF según funcionalidad ADR con periodo de 180 segundos.



Fuente: Elaboración propia.

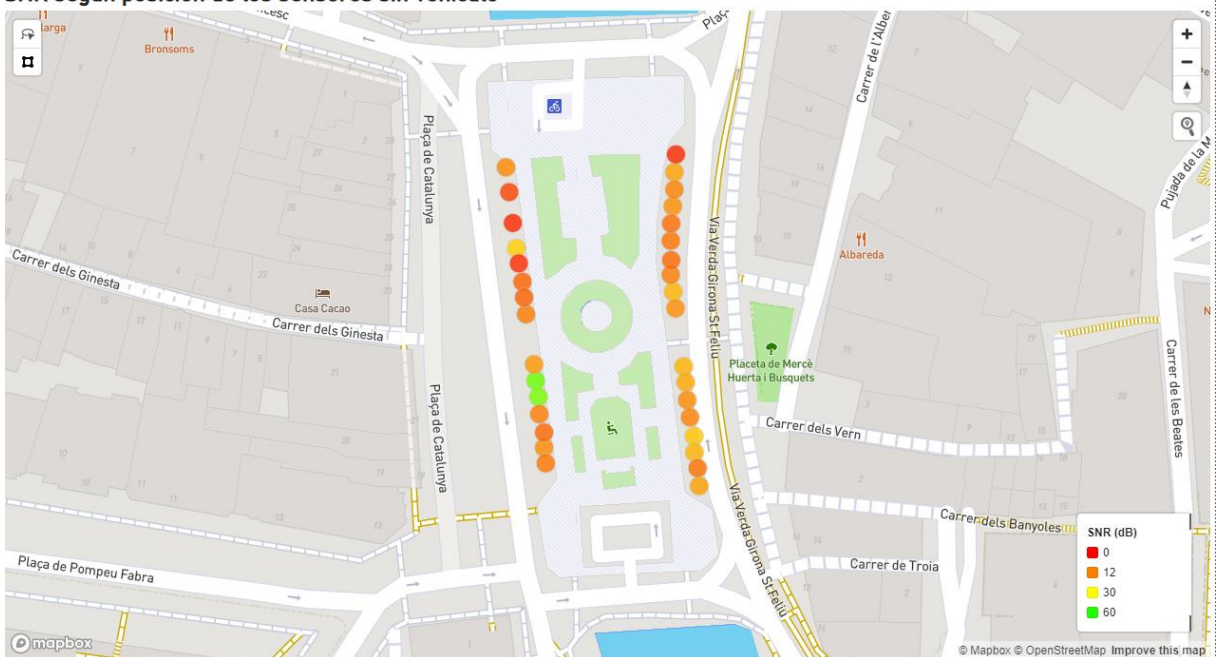
Figura 58. Variación del SF según funcionalidad ADR con periodo de 20 segundos.



Fuente: Elaboración propia.

Figura 59. Mapa del SNR de los sensores sin vehículo en Plaza Catalunya.

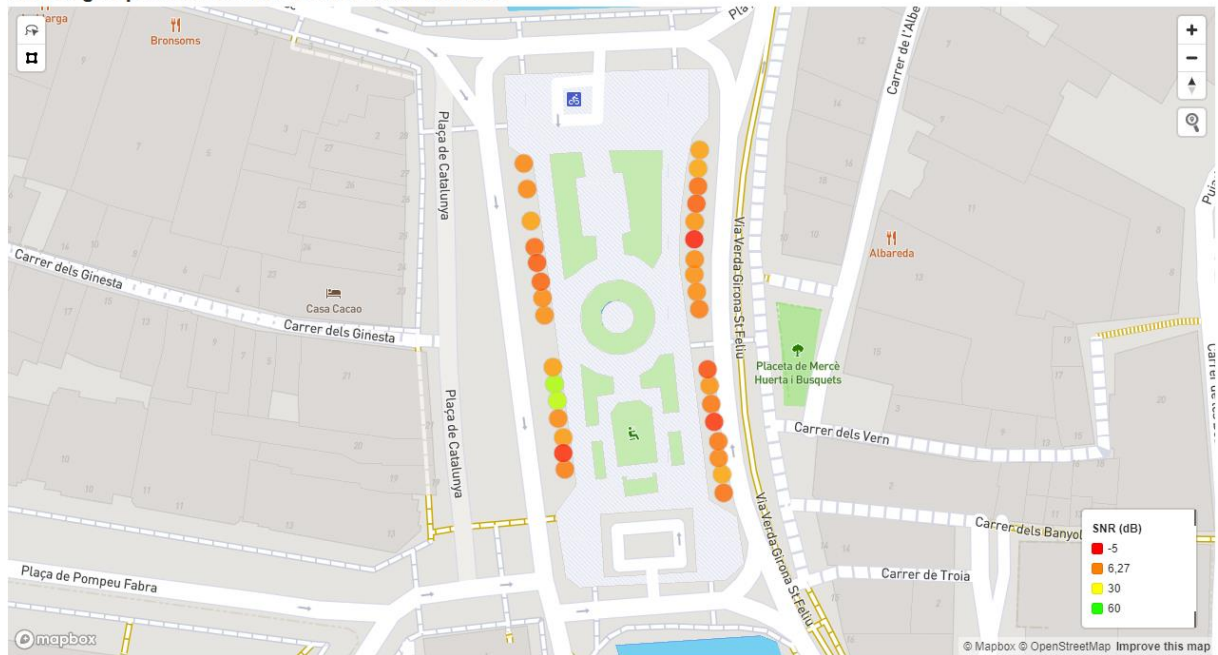
SNR según posición de los sensores sin vehículo



Fuente: Elaboración propia.

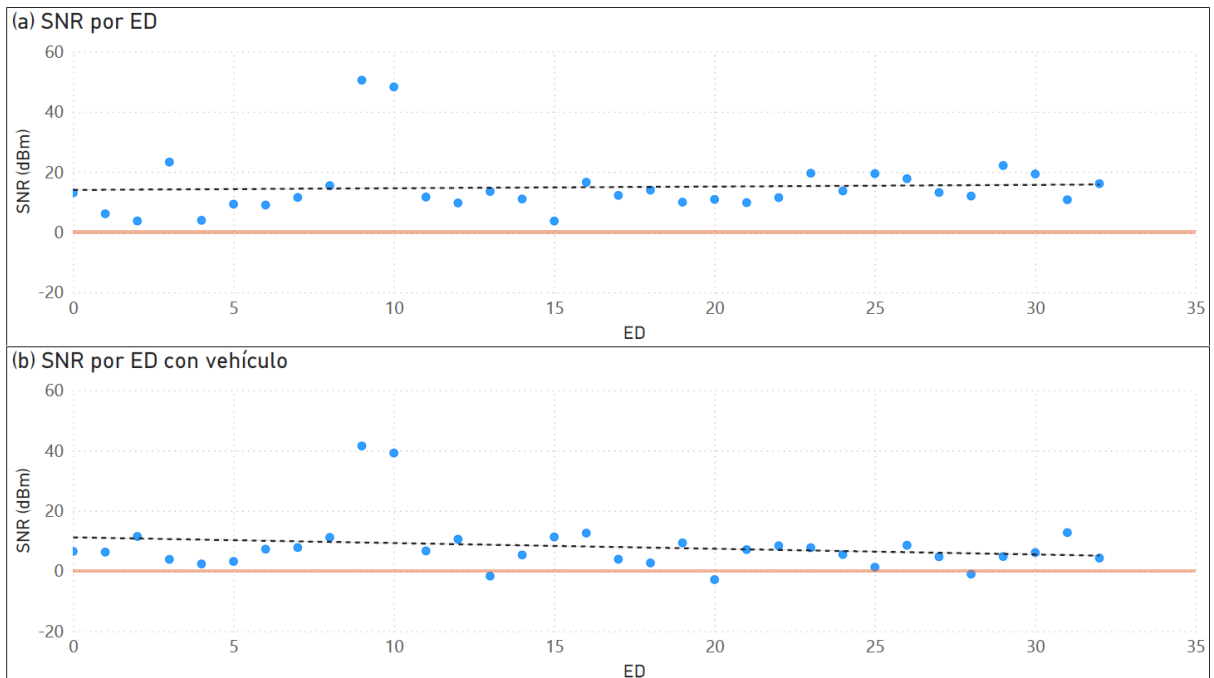
Figura 60. Mapa del SNR de los sensores con vehículo en Plaza Catalunya.

SNR según posición de los sensores con vehículo



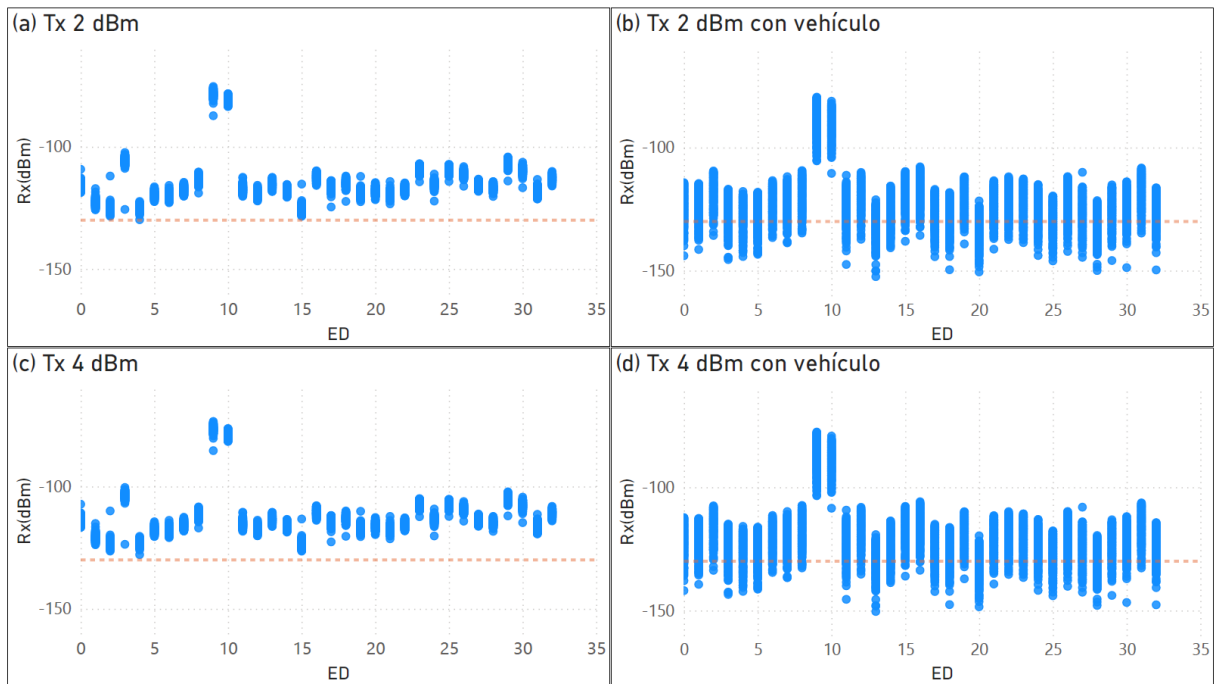
Fuente: Elaboración propia.

Figura 61. Visualización de los promedios del SNR según ED según configuración del análisis.



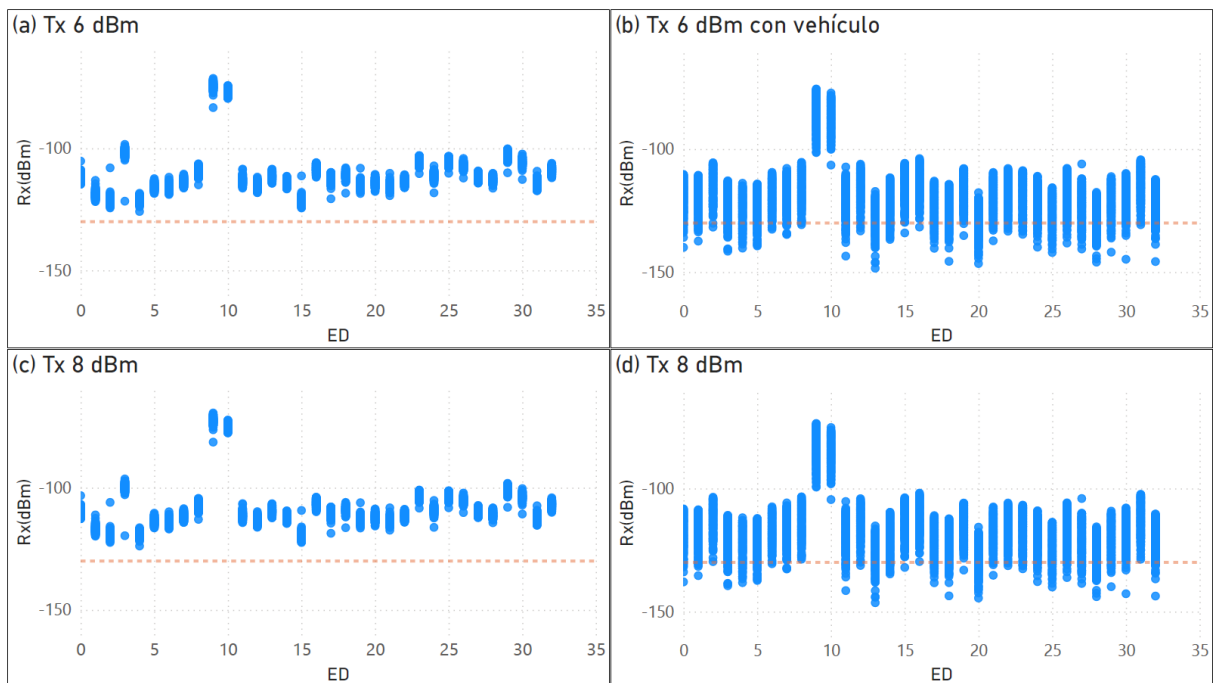
Fuente: Elaboración propia.

Figura 62. Relación del Tx 2 y 4 según Rx y ED según configuración del análisis.



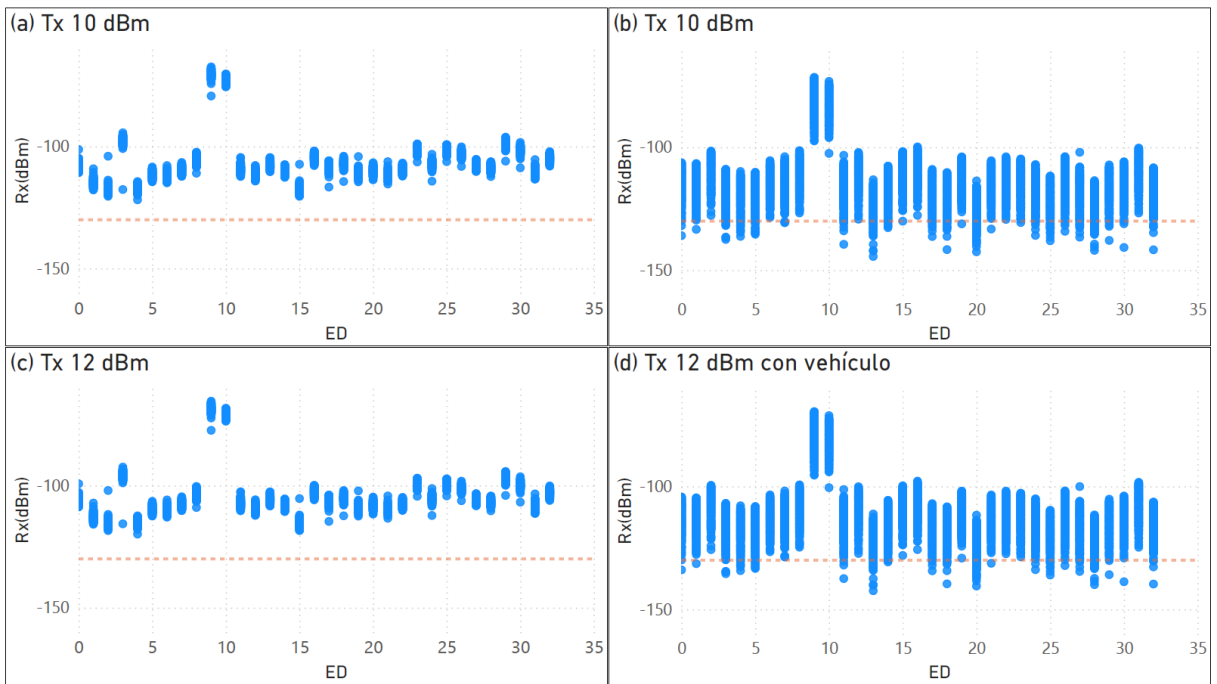
Fuente: Elaboración propia.

Figura 63. Relación del Tx 6 y 8 según Rx y ED según configuración del análisis.



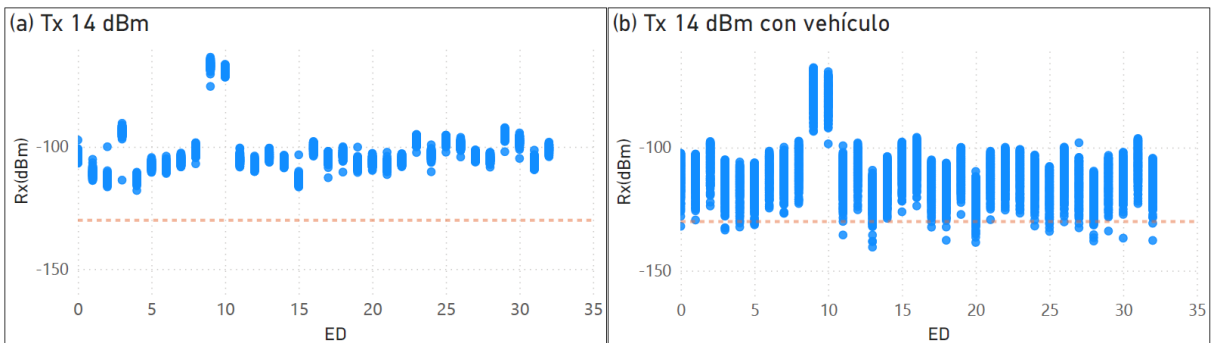
Fuente: Elaboración propia.

Figura 64. Relación del Tx 10 y 12 según Rx y ED según configuración del análisis.



Fuente: Elaboración propia.

Figura 65. Relación del Tx 14 según Rx y ED según configuración del análisis.



Fuente: Elaboración propia.

Anexo D. Código implementado

En este anexo se engloban la totalidad de los programas desarrollados para la realización de las simulaciones, tratamiento y visualización de los datos.

D.1. Código de los escenarios simulados

Aquí queda figurado el programa que se emplea para las simulaciones.

```
/*
 * This script simulates a complex scenario with multiple gateways and end
 * devices. The metric of interest for this script is the throughput of the
 * network.
 */

#include "ns3/animation-interface.h"
#include "ns3/building-allocator.h"
#include "ns3/building-penetration-loss.h"
#include "ns3/buildings-helper.h"
#include "ns3/buildings-module.h"
#include "ns3/class-a-end-device-lorawan-mac.h"
#include "ns3/command-line.h"
#include "ns3/constant-position-mobility-model.h"
#include "ns3/correlated-shadowing-propagation-loss-model.h"
#include "ns3/double.h"
#include "ns3/end-device-lora-phy.h"
#include "ns3/forwarder-helper.h"
#include "ns3/gateway-lora-phy.h"
#include "ns3/gateway-lorawan-mac.h"
#include "ns3/log.h"
#include "ns3/lora-helper.h"
#include "ns3/mobility-helper.h"
#include "ns3/netanim-module.h"
#include "ns3/network-server-helper.h"
#include "ns3/node-container.h"
#include "ns3/periodic-sender-helper.h"
#include "ns3/pointer.h"
#include "ns3/position-allocator.h"
#include "ns3/random-variable-stream.h"
#include "ns3/simulator.h"

#include <algorithm>
#include <ctime>
#include <fstream>
#include <iostream>
#include <ostream>
```

```
// Añadido ADR
#include "ns3/config.h"
#include "ns3/core-module.h"
#include "ns3/end-device-lorawan-mac.h"
#include "ns3/hex-grid-position-allocator.h"
#include "ns3/lora-device-address-generator.h"
#include "ns3/lora-phy-helper.h"
#include "ns3/lorawan-mac-helper.h"
#include "ns3/network-module.h"
#include "ns3/ptr.h"
#include "ns3/rectangle.h"
#include "ns3/string.h"
#include "ns3/trace-helper.h"

#include <stdio.h>

// Añadido Modelo 3GPP
#include "ns3/abort.h"
#include "ns3/buildings-channel-condition-model.h"
#include "ns3/three-gpp-v2v-propagation-loss-model.h"
#include <ns3/three-gpp-propagation-loss-model.h>

// Añadido Energia escenario 3
#include "ns3/basic-energy-source-helper.h"
#include "ns3/file-helper.h"
#include "ns3/lora-radio-energy-model-helper.h"
#include "ns3/names.h"

using namespace ns3;
using namespace lorawan;

NS_LOG_COMPONENT_DEFINE("ComplexLorawanNetworkExample");

// Network settingsheader
int nDevices = 33; // Num. ED
int nGateways = 1; // Num. GW
double radius = 15000; // CObertura teórica de la gateway
double simulationTime = 43200; // 43200 = 12 h // 86400 segundos = 1 día

int packet_size = 11; // Sólo Payload, sin Header de LoRaWAN

// Modelo del canal
bool realisticChannelModel = true;

int appPeriodSeconds = 20; // Periodicidad del ED --> 20 / 180 / 360 / 540 /
720 / 900

// Salida del documento
bool print = true;
```

```

// Llamada de cambio de Data rate y Tx con ADR activado
void
OnDataRateChange(uint8_t oldDr, uint8_t newDr)
{
    NS_LOG_DEBUG("DR" << unsigned(oldDr) << " -> DR" << unsigned(newDr));
}

void
OnTxPowerChange(double oldTxPower, double newTxPower)
{
    NS_LOG_DEBUG(oldTxPower << " dBm -> " << newTxPower << " dBm");
}

// Función principal
int
main(int argc, char* argv[])
{
    // Loggers
    LogComponentEnable("LoraChannel", LOG_LEVEL_INFO);
    LogComponentEnable("EndDeviceLorawanMac", LOG_LEVEL_ALL);
    LogComponentEnable("ClassAEndDeviceLorawanMac", LOG_LEVEL_ALL);
    LogComponentEnable("GatewayLorawanMac", LOG_LEVEL_ALL);

    LogComponentEnableAll(LOG_PREFIX_FUNC);
    LogComponentEnableAll(LOG_PREFIX_NODE);
    LogComponentEnableAll(LOG_PREFIX_TIME);

    /*****
     * Setup *
     *****/

    // Modo ED para que el NS actue en consideración
    bool adrEnabled = false;
    std::string adrType = "ns3::AdrComponent";

    Config::SetDefault("ns3::EndDeviceLorawanMac::DRControl",
BooleanValue(false));
    // Comentar si se activa el DRControl i viceversa
    Config::SetDefault("ns3::EndDeviceLorawanMac::DataRate",
UIntegerValue(5));
    // // -----
    Config::SetDefault("ns3::EndDeviceLorawanMac::MType",
EnumValue(LorawanMacHeader::UNCONFIRMED_DATA_UP));
    // // El ED de Winext solo permite 2 intentos extra para el ACK, 3 porque
se empieza por el
// primero
    Config::SetDefault("ns3::EndDeviceLorawanMac::MaxTransmissions",
IntegerValue(3));

```

```

// Create the time value from the period
Time appPeriod = Seconds(appPeriodSeconds);

// Características de retansmisión LoRa, poner lo mismo que en la GATEWAY
MobilityHelper mobility;
mobility.SetPositionAllocator("ns3::UniformDiscPositionAllocator",
                              "rho",
                              DoubleValue(radius),
                              "X",
                              DoubleValue(580.0),
                              "Y",
                              DoubleValue(-50));
mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");

/*****
 * Crear el modelo del canal *
 *****/

ObjectFactory propagationLossModelFactory =
    ObjectFactory("ns3::ThreeGppV2vUrbanPropagationLossModel");
Ptr<ThreeGppV2vUrbanPropagationLossModel> m_lossModel =
    propagationLossModelFactory.Create<ThreeGppV2vUrbanPropagationLossModel>();

Ptr<ChannelConditionModel> losCondModel =
CreateObject<BuildingsChannelConditionModel>();
m_lossModel->SetAttribute("ShadowingEnabled", BooleanValue(true));
m_lossModel->SetAttribute("Frequency", DoubleValue(868.0e6));
m_lossModel->SetAttribute("BuildingPenetrationLossesEnabled",
BooleanValue(true));
m_lossModel->SetAttribute("ChannelConditionModel",
PointerValue(losCondModel));

Ptr<PropagationDelayModel> delay =
CreateObject<ConstantSpeedPropagationDelayModel>();
Ptr<LoraChannel> channel = CreateObject<LoraChannel>(m_lossModel, delay);

/*****
 * Create the helpers *
 *****/

// Create the LoraPhyHelper
LoraPhyHelper phyHelper = LoraPhyHelper();
phyHelper.SetChannel(channel);

// Create the LorawanMacHelper
LorawanMacHelper macHelper = LorawanMacHelper();

```

```

// Create the LoraHelper
LoraHelper helper = LoraHelper();
helper.EnablePacketTracking();

// Create the NetworkServerHelper
NetworkServerHelper nsHelper = NetworkServerHelper();

// Create the ForwarderHelper
ForwarderHelper forHelper = ForwarderHelper();

// // Create the LoraFrameHeader
LoraFrameHeader frameHdr = LoraFrameHeader();

/*****
 * Create End Devices *
*****/

// Create a set of nodes
NodeContainer endDevices;
endDevices.Create(nDevices);

// Assign a mobility model to each node
mobility.Install(endDevices);
// Make it so that nodes are at a certain height > 0
int n_ed = 0; // Iterador para operaciones
for (NodeContainer::Iterator j = endDevices.Begin(); j !=
endDevices.End(); ++j)
{
    Ptr<MobilityModel> mobility = (*j)->GetObject<MobilityModel>();
    Vector position = mobility->GetPosition();
    // Alineado con el Norte

    if (n_ed < nDevices)
    {
        // unidad (metros)
        // Primera columna
        if (n_ed <= 3)
        {
            position.x = n_ed * 0.5;
            position.y = n_ed * (-6);
        }
        else if ((n_ed >= 4) && (n_ed <= 7))
        {
            position.x = n_ed * 0.5;
            position.y = n_ed * (-3.5) - 7.5; // Acumulado de las plazas
de discapacitados
        }
        else if ((n_ed >= 8) && (n_ed <= 14))
        {

```

```

        position.x = n_ed * 0.5;
        position.y = n_ed * (-3.5) - 14.5; // separación
    }
    // Segunda columna
    else if ((n_ed >= 15) && (n_ed <= 24))
    {
        position.x = n_ed * 0.3 + 36; // offset x respecto el (0,0)
        position.y = n_ed * (-3.5) + 1; // offset y respecto el (0,0)
    }
    else if ((n_ed >= 25) && (n_ed <= 32))
    {
        position.x = n_ed * 0.3 + 36;
        position.y =
            n_ed * (-3.5) - 12.5; // 13.5 de separación + 1 que
arrartra del offset (0,0)
    }
    position.z = 0.01;
    mobility->SetPosition(position);
    ++n_ed;
}
}

// Create the LoraNetDevices of the end devices
uint8_t nwkId = 54;
uint32_t nwkAddr = 1864;
Ptr<LoraDeviceAddressGenerator> addrGen =
    CreateObject<LoraDeviceAddressGenerator>(nwkId, nwkAddr);

// Create the LoraNetDevices of the end devices
macHelper.SetAddressGenerator(addrGen);
phyHelper.SetDeviceType(LoraPhyHelper::ED);
macHelper.SetDeviceType(LorawanMacHelper::ED_A);
macHelper.SetRegion(LorawanMacHelper::EU);
helper.Install(phyHelper, macHelper, endDevices);
// Añadido para Escenario 3 //
NetDeviceContainer endDevicesNetDevices = helper.Install(phyHelper,
macHelper, endDevices);
// ----- //

// Conectar trazabilidad por cada ED
for (NodeContainer::Iterator j = endDevices.Begin(); j !=
endDevices.End(); ++j)
{
    Ptr<Node> node = *j;
    Ptr<LoraNetDevice> loraNetDevice = node->GetDevice(0)-
>GetObject<LoraNetDevice>();
    Ptr<LoraPhy> phy = loraNetDevice->GetPhy();
}

```



```

/*****
 * Create Gateways *
 *****/

// Create the gateway nodes (allocate them uniformly on the disc)
NodeContainer gateways;
gateways.Create(nGateways);

Ptr<ListPositionAllocator> allocator =
CreateObject<ListPositionAllocator>();
allocator->Add(Vector(580, -50, 112.3)); // Torre Alfons XIII
mobility.SetPositionAllocator(allocator);
mobility.SetMobilityModel("ns3::ConstantPositionMobilityModel");
mobility.Install(gateways);

// Create a netdevice for each gateway
phyHelper.SetDeviceType(LoraPhyHelper::GW);
macHelper.SetDeviceType(LorawanMacHelper::GW);
helper.Install(phyHelper, macHelper, gateways);

/*****
 * Handle buildings *
 *****/

double xLength = 20;
double deltaX = 10;
double yLength = 30;
double deltaY = 5;
double zHeight = 20;
int gridWidth = 30;
int gridHeight = 20;
double offX = -250; // Oeste-Este
double offY = -500; // Norte-Sur

if (realisticChannelModel == false)
{
    gridWidth = 0;
    gridHeight = 0;
}

// Edificios
Ptr<GridBuildingAllocator> gridBuildingAllocator;
gridBuildingAllocator = CreateObject<GridBuildingAllocator>();
gridBuildingAllocator->SetAttribute("GridWidth",
UIntegerValue(gridWidth));
gridBuildingAllocator->SetAttribute("MinX", DoubleValue(offX));
gridBuildingAllocator->SetAttribute("MinY", DoubleValue(offY));
gridBuildingAllocator->SetAttribute("LengthX", DoubleValue(xLength));
gridBuildingAllocator->SetAttribute("LengthY", DoubleValue(yLength));
gridBuildingAllocator->SetAttribute("DeltaX", DoubleValue(deltaX));

```

```

gridBuildingAllocator->SetAttribute("DeltaY", DoubleValue(deltaY));
gridBuildingAllocator->SetAttribute("Height", DoubleValue(zHeight)); //
Revisar altures
gridBuildingAllocator->SetAttribute("LayoutType",
StringValue("ColumnFirst"));
gridBuildingAllocator->SetBuildingAttribute("NRoomsX", UIntegerValue(2));
gridBuildingAllocator->SetBuildingAttribute("NRoomsY", UIntegerValue(4));
gridBuildingAllocator->SetBuildingAttribute("NFloors", UIntegerValue(2));

// Constructor
BuildingContainer bContainer = gridBuildingAllocator->Create(gridHeight *
gridWidth);

// Unimos el modelo de los edificios con los ED y GW, imprescindible!
BuildingsHelper::Install(endDevices);
BuildingsHelper::Install(gateways);

// Capturar los datos de las posiciones de los edificios, para la
representación por Python
if (print)
{
    std::ofstream myfile;
    myfile.open("buildings.csv");
    std::vector<Ptr<Building>>::const_iterator it;
    int j = 1;
    for (it = bContainer.Begin(); it != bContainer.End(); ++it, ++j)
    {
        Box boundaries = (*it)->GetBoundaries();
        myfile << j << "," << boundaries.xMin << "," << boundaries.yMin <<
", "
        << boundaries.zMin << "," << boundaries.xMax << "," <<
boundaries.yMax << ","
        << boundaries.zMax << std::endl;
    }
}

/*****
 * Install applications on the end devices *
*****/

Time appStopTime = Seconds(simulationTime);
PeriodicSenderHelper appHelper = PeriodicSenderHelper();
appHelper.SetPeriod(Seconds(appPeriodSeconds));
appHelper.SetPacketSize(packet_size);

ApplicationContainer appContainer = appHelper.Install(endDevices);
appContainer.Start(Seconds(0));
appContainer.Stop(appStopTime);

```

```

/*****
 * Install Energy Model *
 *****/

BasicEnergySourceHelper basicSourceHelper;
LoraRadioEnergyModelHelper radioEnergyHelper;

// Configuración del modelo de energia
// Batería
basicSourceHelper.Set("BasicEnergySourceInitialEnergyJ",
DoubleValue(207360)); // Energy in J
basicSourceHelper.Set("BasicEnergySupplyVoltageV", DoubleValue(3.3));

// Energía para el ED
radioEnergyHelper.Set("StandbyCurrentA", DoubleValue(0.006));
radioEnergyHelper.Set("TxCurrentA", DoubleValue(0.068));
radioEnergyHelper.Set("SleepCurrentA", DoubleValue(0.000015));
radioEnergyHelper.Set("RxCurrentA", DoubleValue(0.01075));

// Modelo de consumo fijo, desactivar si existe ADR activado
radioEnergyHelper.SetTxCurrentModel("ns3::ConstantLoraTxCurrentModel",
                                     "TxCurrent",
                                     DoubleValue(0.068));

// Instalar para el primer sensor
EnergySourceContainer sources = basicSourceHelper.Install(endDevices);
Names::Add("/Names/EnergySource",
           sources.Get(0)); // Sembla que ja funciona pel numero que posa
aquí

// Instalar el modelo para el ED
DeviceEnergyModelContainer deviceModels =
    radioEnergyHelper.Install(endDevicesNetDevices, sources);

/*****
 * Get output *
 *****/

// Generación del archivo para el nivel de la bateria
FileHelper fileHelper;
fileHelper.ConfigureFile("battery-level",
FileAggregator::SPACE_SEPARATED);
fileHelper.WriteProbe("ns3::DoubleProbe",
"/Names/EnergySource/RemainingEnergy", "Output");

/*****
 * Create Network Server *
 *****/

// Create the NS node

```

```

NodeContainer networkServer;
networkServer.Create(1);

// Create a NS for the network
nsHelper.SetGateways(gateways);
nsHelper.SetEndDevices(endDevices);
nsHelper.EnableAdr(adrEnabled);
nsHelper.SetAdr(adrType);
nsHelper.Install(networkServer);

// Create a forwarder for each gateway
forHelper.Install(gateways);

NS_LOG_DEBUG("Completed configuration");

//////////
// Simulation //
//////////

// // Connect our traces
Config::ConnectWithoutContext(
    "/NodeList/*/DeviceList/0/$ns3::LoraNetDevice/Mac/$ns3::EndDeviceLoraw
anMac/TxPower",
    MakeCallback(&OnTxPowerChange));
Config::ConnectWithoutContext(
    "/NodeList/*/DeviceList/0/$ns3::LoraNetDevice/Mac/$ns3::EndDeviceLoraw
anMac/DataRate",
    MakeCallback(&OnDataRateChange));

Simulator::Stop(appStopTime); //+ Hours(1));

NS_LOG_INFO("Running simulation...");

Simulator::Run();

Simulator::Destroy();

//////////
// Print results to file //
//////////
NS_LOG_INFO("Computing performance metrics...");

LoraPacketTracker& tracker = helper.GetPacketTracker();
std::cout << tracker.CountMacPacketsGlobally(Seconds(0), appStopTime +
Hours(1)) << std::endl;

return 0;
}

```

D.2. Representación de los edificios

A continuación, se representa la totalidad del código para mostrar los edificios.

```
import os
import pandas as pd
import numpy as np
import csv
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from mpl_toolkits.mplot3d.art3d import Poly3DCollection

# Definir el Path
path = os.path.dirname(os.path.abspath(__file__)) #

# Datos del CSV
data = []

def plot_box(ax, x, y, z, width, height, depth):
    # Definir las coordenadas de las 8 caras de los edificios
    vertices = [
        [x, y, z],
        [width, y, z],
        [width, height, z],
        [x, height, z],
        [x, y, depth],
        [width, y, depth],
        [width, height, depth],
        [x, height, depth],
    ]

    # Definir las 12 líneas de unión
    edges = [
        [0, 1],[1, 2],[2, 3],[3, 0], # Top face
        [4, 5],[5, 6],[6, 7],[7, 4], # Bottom face
        [0, 4],[1, 5],[2, 6],[3, 7], # Vertical edges
    ]

    # Creamos los vértices junto a las uniones
    for edge in edges:
        ax.plot(
            [vertices[edge[0]][0], vertices[edge[1]][0]],
            [vertices[edge[0]][1], vertices[edge[1]][1]],
            [vertices[edge[0]][2], vertices[edge[1]][2]],
            "k",
        )

# Creación de los edificios
```

```

fig = plt.figure()
ax = fig.add_subplot(111, projection="3d")

# Rlectura del CSV --> DataFrame
data = pd.read_csv(path + "\\buildings.csv")

for i in range(len(data.values)):
    plot_box(
        ax,
        data.values[i][1],
        data.values[i][2],
        data.values[i][3],
        data.values[i][4],
        data.values[i][5],
        data.values[i][6],
    )

ax.set_xlabel("X-axis (m)")
ax.set_ylabel("Y-axis (m)")
ax.set_zlabel("Z-axis (m)")
ax.set_title("3D Edificios")

ax.set_xlim(-300, 700)
ax.set_ylim(-500, 500)
ax.set_zlim(-10, 150)
# Vista
# ax.view_init(0, 100, 0)

# Impresión
plt.show()

```

D.3. Digestor de datos para los distintos escenarios

Este es el *script* de Python para convertir los datos a un formato relacional.

```

import os
import glob
import pandas as pd
import math
import numpy as np

# Para debugar
from tqdm import tqdm
import time

# Definir el Path
path = os.path.dirname(os.path.abspath(__file__)) #

```

```
# Características generales
# Igual al número de dispositivos, la lista empieza por 0
number_gateway = 33

# Definir las palabras claves de búsqueda.
keywords = [
    "PacketToSend:",
    "Propagation:",
    "dstNode = ",
    "Received packet:",
    "ACK packet",
]

# Datos posiciones ED (latitude / longitude)
pos_ed = [
    [41.981988, 2.823137],
    [41.981942, 2.823145],
    [41.981885, 2.823154],
    [41.981838, 2.823163],
    [41.981810, 2.823169],
    [41.981776, 2.823178],
    [41.981747, 2.823182],
    [41.981716, 2.823187],
    [41.981623, 2.823207],
    [41.981593, 2.823211],
    [41.981563, 2.823218],
    [41.981531, 2.823220],
    [41.981497, 2.823232],
    [41.981469, 2.823232],
    [41.981440, 2.823236],
    [41.982012, 2.823561],
    [41.981979, 2.823557],
    [41.981947, 2.823557],
    [41.981916, 2.823553],
    [41.981884, 2.823549],
    [41.981852, 2.823548],
    [41.981816, 2.823549],
    [41.981789, 2.823548],
    [41.981758, 2.823554],
    [41.981727, 2.823560],
    [41.981619, 2.823580],
    [41.981590, 2.823585],
    [41.981557, 2.823589],
    [41.981525, 2.823596],
    [41.981490, 2.823606],
    [41.981460, 2.823607],
    [41.981431, 2.823615],
    [41.981398, 2.823619],
```

```

]

# Debug
Num_Logs = 0
Data_Iterations = 1
Log_iteraciones = 3
first_digest = 0
second_digest = 0

# Modalidad 0 = Desactivado / 1 = Activado
Ack = 0
Adr = 0

# //////////////////////////////////////#
# /                                     /#
# /           Inicio Programa          /#
# /                                     /#
# //////////////////////////////////////#
with tqdm(total=Data_Iterations, desc="Data_Progress", unit="iteration") as
databar:
    for i in range(Data_Iterations):
        # Creación de una lista vacía donde poner los datos
        data = [] # Datos en crudo
        data_final = [] # Datos procesados

        # Tabla de datos final
        Time_arr = [] # Tiempo del envío
        EndD = [] # Número de dispositivo
        SF_num = []
        Tx = [] # Potencia de envío
        Rx = [] # Potencia de recepción
        Dst = [] # Distancia
        Delay = [] # Duración en el aire
        Signal_t_Noise_R = [] # SNR
        ED_Packet = [] # Paquete enviado
        Received = [] # Marca para saber si la gateway lo recibió
        Time_Received = [] # Tiempo en que la gateway lo recibió
        ACK_msg = [] # Señal que el sensor ha recibido el ACK de la gateway
        Msg_B_ACK = [] # Mensajes enviados por el dispositivo antes de
recibir el ACK
        subband = [] # Banda en la que se ha enviado el mensaje
        latitude = [] # Posición del ED
        longitude = [] # Posición del ED

        # Debido a que la gateway puede tratar varios mensajes a la vez
        # se genera una memoria previa a su volcado final

        mem_Received = []
        mem_Time_Received = []

```



```

mem_EndD = []
mem_Pkt_Received = []
mem_Send = 0
mem_Pkt_recived = 8 # Max 8. Canales totales de la gateway
with tqdm(total=Log_iteraciones, desc="Log_Progress",
unit="iteration") as pbar:
    # Loop a través de los archivos
    for file in glob.glob(os.path.join(path, f"log{Num_Logs}.out")):
        with open(file, "r") as f:
            # Leer los archivos
            contents = f.read()
            first_digest = np.sum(contents.count(f"\n"))
            with tqdm(
                total=first_digest, desc="Digestor 1",
unit="iteration"
            ) as sbar:
                # Búsqueda de palabras clave
                if any(keyword in contents for keyword in keywords):
                    # Extraer los datos relevantes
                    for line in contents.split("\n"):
                        if any(keyword in line for keyword in
keywords):
                            data.append([line])
                            sbar.update(1)

                pbar.update(1)
                second_digest = len(data)
                with tqdm(total=second_digest, desc="Digestor 2",
unit="iteration") as tbar:
                    # Marca para saber si la gateway ya procesado varios a la vez
                    m_received = 0
                    # Buscador de datos
                    for i in range(0, len(data), 1):
                        # Volcar la tupla de caracteres
                        t_data = str(data[i])

                        if t_data.find(f"PacketToSend:") > 0 and not mem_Send:
                            # Primera iteración
                            # if mem_Send == 0:
                            mem_Send = 1
                            EndD.append(
                                int(
                                    t_data[t_data.find("s ") + 1 :
t_data.find("Class") - 1]
                                )
                            )
                            ED_Packet.append(t_data[t_data.find("Send: ") + 6 : -
2])

```

```

# Mirar si la línea es de datos o un msg captado por la
gateway
elif t_data.find(f"dstNode = {number_gateway}") > 0:
    mem_Send = 0 # Resetear la marca de mem_Send
    t_data = str(data[i - 1]) # Recuperar el mensaje
    if t_data.find("Propagation:") > 0:
        Time_arr.append(
            float(
                t_data[t_data.find("+") + 1 :
t_data.find("s ") - 1]
            )
        )
        SF_num.append(float(t_data[t_data.find(" SF:") + 4
: -2]))
        Tx.append(
            float(
                t_data[
                    t_data.find("txPower=") + 8 :
t_data.find("db")
                ]
            )
        )
        Rx.append(
            float(
                t_data[
                    t_data.find("rxPower=")
                    + 8 : t_data.find("distance")
                    - 5
                ]
            )
        )
        Dst.append(
            float(
                t_data[
                    t_data.find("distance=")
                    + 9 : t_data.find("delay")
                    - 3
                ]
            )
        )
        Delay.append(
            float(
                t_data[
                    t_data.find("delay="+) + 7 :
t_data.find("ns")
                ]
            )
        )
        Signal_t_Noise_R.append(

```

```

float(
    Rx[-1] + 174 - 10 * math.log10(125000) - 6
) # El 6 es la forma del Chip SX1276 "Noise
Figure"
)
Received.append(0)
Time_Received.append(0)
ACK_msg.append(0)
Msg_B_ACK.append(0)
subband.append(
    float(
        t_data[
            t_data.find("freq_band: ")
            + 11 : t_data.find(" SF:")
        ]
    )
)
latitude.append(pos_ed[EndD[-1]][0])
longitude.append(pos_ed[EndD[-1]][1])
elif (
    packet:") > 0
):
    mem_Pkt_Received.append(t_data[t_data.find("et: ") + 4
: -2])
    mem_Time_Received.append(
        float(t_data[t_data.find("+") + 1 : t_data.find("s
") - 1])
    )
    # Buscar si existe previamente Max. 8
    for z in range(mem_Pkt_recived):
        if mem_Pkt_Received[-1] == ED_Packet[-1 - z]:
            # Cambiar el indexado
            Received[-1 - z] = 1
            Time_Received[-1 - z] = mem_Time_Received[-1]
            break
    # ACK Si salta, añadir al ultimo mensaje del sensor al que
pertenece
elif t_data.find("ACK packet") > 0:
    node_ack = 0 # Node que ha recibido el ACK
    # m_node = 0 # Memoria del índice del node
    msg_b_node = 0 # número de msg del node anteriores
    m_array = [] # Memoria array

    # Igualar el array ED
    m_array = EndD.copy()

    # Buscar el sensor al que pertenece
    node_ack = int(

```

```

        t_data[t_data.find(" ") + 1 : t_data.find("
EndDevice")]
    )

    # Capturar el numero de mensajes anteriores
    msg_b_node = int(
        t_data[t_data.find("after ") + 6 : t_data.find("
transmi")]
    )

    # Una vez se tienen los dos se busca el último sensor
    for i2 in range(mem_Pkt_recived):
        if m_array[-1 - i2] == node_ack:
            ACK_msg[-1 - i2] = 1
            Msg_B_ACK[-1 - i2] = msg_b_node
            break

    # Agumentar puntero de progresion
    tbar.update(1)
    pbar.update(1)
    # Revisar que la simulación no se haya entrecortado, asegurando
    último mensaje
    if len(Time_arr) != len(EndD):
        EndD.pop(-1)
    # Convertir los datos a pandas-DataFrame
    for i in range(len(EndD)):
        data_final.append(
            [
                Time_arr[i],
                EndD[i],
                SF_num[i],
                Tx[i],
                Rx[i],
                Signal_t_Noise_R[i],
                Dst[i],
                Delay[i],
                ED_Packet[i],
                Received[i],
                Time_Received[i],
                ACK_msg[i],
                Msg_B_ACK[i],
                subband[i],
                latitude[i],
                longitude[i],
            ]
        )
    df = pd.DataFrame(
        data_final,
        columns=[

```

```
        "Time (second)",
        "ED",
        "SF",
        "Tx(dBm)",
        "Rx(dBm)",
        "SNR(dBm)",
        "Dst(m)",
        "Delay(ns)",
        "ED_Packet",
        "Received",
        "Time_Received (second)",
        "ACK_Received",
        "Msg_Before_ACK",
        "Sub-Banda (MHz)",
        "Latitude",
        "Longitude",
    ],
)

# Guardar el DataFrame en el archivo de Excel
print("\n Construyendo el Excel...")
df.to_excel(path + f"\\log_escenari{Num_Logs}.xlsx", index=True)
# Actualizar Log
pbar.update(1)
# Actualizar Num Logs y Data
Num_Logs += 1
databar.update(1)
```