

Juan Manuel Palacios González

**ESTUDIO DEL SISTEMA DE CONEXIÓN
INALÁMBRICA MEDIANTE NODOS DE
COMUNICACIÓN CON SALIDA A INTERNET**

TRABAJO DE FIN DE GRADO

Dirigido por German Galià Beltran y Ramón María Villarino Villarino
Grado de Ingeniería de Sistemas y Servicios de Telecomunicaciones



UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2021

Agradecimientos

Me gustaría expresar mi agradecimiento a la empresa *ETECNIC* por acogerme y darme una idea para el desarrollo de mi trabajo final de grado. Al tutor de la empresa German Galià por brindarme la oportunidad de realizar el trabajo y asesorarme en su desarrollo. Asimismo, agradecer a mi tutor académico Ramón María Villarino por las explicaciones y los conocimientos académicos para orientarme y guiarme durante el desarrollo del trabajo, y también, a Carlos Sánchez por reunirse conmigo para entrevistarse y ampliar mi conocimiento en las redes WSN.

Finalmente, quisiera agradecer profundamente a mis padres por su apoyo incondicional a lo largo de estos meses de trabajo.

Resumen

En este trabajo se detalla el desarrollo de un sistema de red inalámbrica mediante el uso de la tecnología LoRaWAN, para dar conectividad y salida a Internet a los sensores de cargadores para coches eléctricos en un parking subterráneo y así, gestionar la información de estos sensores desde el exterior a través de la base de datos almacenada en la nube.

El trabajo se compone de dos partes principales. La primera parte, consiste en el estudio de viabilidad sobre los cuatros estándares WSN más conocidos dentro del ámbito de las telecomunicaciones. En este estudio veremos reflejada la explicación detallada de los cuatro estándares, para posteriormente, compararlos, sacar conclusiones y escoger el más adecuado para el desarrollo de un prototipo que cumpla con unos requisitos específicos detallados a lo largo del proyecto.

La segunda parte, se basará en el desarrollo de un prototipo escogido en función del estudio de viabilidad previo, el protocolo con el que realizaremos el despliegue será LoRaWAN. En este apartado detallaremos la realización del prototipo, los pasos que hemos seguido para poder crear una red inalámbrica con LoRa y mostraremos con un prototipo como realizamos la comunicación entre sensores y la nube.

La idea de realizar este servicio viene brindada por la empresa ETECNIC, empresa especializada en ingeniería eléctrica y movilidad sostenible. Conocedores de la dificultad de dar conectividad a Internet en espacios subterráneos, se propone este estudio para buscar la manera óptima de conseguir un sistema de monitorización *wireless* y adaptarlo a sus sensores de cargadores para coches eléctricos.

Índice

1. Introducción y Objetivos.....	14
1.1. Presentación y contexto.....	14
1.2. Misión y objetivos del trabajo	15
1.3. Puesta en escena al objetivo del estudio.....	16
2. Antecedentes.....	19
2.1. Redes inalámbricas de sensores (WSN).....	19
2.1.1. Características generales.....	19
2.1.2. Aplicaciones	21
2.1.3. Topología.....	22
2.1.4. Variabilidad del canal	25
2.1.5. Consumo energético	26
2.1.6. Seguridad	26
2.2. Estándares de comunicación WSN	27
2.2.1. Bluetooth (IEEE 802.15.1)	27
2.2.2. Wi-Fi (IEEE 802.11.ac).....	30
2.2.3. Zigbee (IEEE 802.15.4).....	35
2.2.4. LoRaWAN (LoRa)	41
2.3. Comparación y elección del estándar	45
2.4. Entornos de desarrollo	48
2.4.1. IDE Arduino	48
2.4.2. Hive MQ.....	50
3. Prototipo con LoRaWAN.....	52
3.1. Reafirmación al uso de esta tecnología	52
3.2. Planificación del proyecto.....	53
3.3. Desarrollo del proyecto	56
3.3.1. Estudio del entorno	56
3.3.2. Material y recursos.....	59
3.3.3. Diseño.....	64
3.3.4. Aprovechamiento de materiales y recursos	65
3.3.5. Programación de dispositivos	65
3.3.6. Seguridad del prototipo demostrador.....	75
3.3.7. Diagrama de Gantt.....	76
3.3.8. Presupuestos	76
3.3.9. Conclusiones sobre prototipo LoRaWAN	77
4. Conclusiones y Líneas futuras	79
4.1. Conclusiones.....	79
4.2. Líneas futuras	80
5. Planos	81
5.1. Plano general de la instalación en la planta -1.....	81

5.2.	Plano general de la instalación en la planta -2	82
6.	Referencias y Webgrafía.....	83
7.	Anexos	85
7.1.	Código <i>end device</i> del envío de paquetes LoRa	85
7.2.	Código <i>gateway</i> de recepción LoRa junto a su potencia recibida.....	86
7.3.	Creación y envío de la información de los sensores para cargadores de EV	87
7.4.	Recepción LoRa al <i>gateway</i> y subida de la información a HiveMQ.....	89
7.5.	Diagrama Gantt de la fase 1	93

Listado de acrónimos

ACK	Acknowledgement
AES	Advanced Encryption Standard
AP	Acces Point
APL	Application Layer
APS	Application Support Sublayer
CMAC	Cipher-based Message Authentication Code
CSMA-CA	Carrier Sin Multiple Access Collision Avoidance
CTR	Click Through Rate
ETSE	Escola Tècnica Superior d'Enginyeria
EV	Electric Vehicle
FSK	Frequency Shift Keying
GPON	Gigabit-capable Passive Optical Network
HAL	Hardware Abstraction Layer
IDE	Integrated Development Environment
IoT	Internet of Things
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
ISO	International Organization for Standardization
LPWAN	Low Power Wide Area Network
LQI	Link Quality Indicator
MAC	Media Access Control
MQTT	Message Queue Telemetry Transport
NWK	Network Layer
OTAA	Over the Air Activation
PHY	Physical Layer
RBS	Radio Base Station
RF	Radio Frecuencia
TCP	Transmission Control Protocol
TFG	Trabajo de Final de Grado
TKIP	Temporal Key Integrity Protocol
TLS	Transport Layer Security
TTN	The Things Network
URV	Universitat Rovira i Virgili
USB	Universal Serial Bus
WBS	Work Breakdown Structure
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WSAN	Wireless Sensors and Actuator Networks
WSN	Wireless Sensor Networks

Listado de figuras

<i>Figura 1.1. Seguimiento del trabajo final de grado</i>	16
<i>Figura 2.1. Diagrama de bloques general de un nodo WSN [3]</i>	20
<i>Figura 2.2. Aplicaciones WSN [4]</i>	22
<i>Figura 2.3. Ejemplo infraestructura WSN [5]</i>	22
<i>Figura 2.4. Topología no estructurada [5]</i>	23
<i>Figura 2.5. Topología estrella [6]</i>	23
<i>Figura 2.6. Topología bus [6]</i>	24
<i>Figura 2.7. Topología árbol [3]</i>	24
<i>Figura 2.8. Topología mesh [3]</i>	25
<i>Figura 2.9. Topología cluster [3]</i>	25
<i>Figura 2.10. Estados de operación nodo en modo sleep [3]</i>	26
<i>Figura 2.11. Scatternet Bluetooth [10]</i>	28
<i>Figura 2.12. Arquitectura Bluetooth [10]</i>	29
<i>Figura 2.13. Ejemplo de una infraestructura Wi-Fi</i>	32
<i>Figura 2.14. Ejemplo de estructura árbol Zigbee</i>	36
<i>Figura 2.15. Ejemplo de encaminamiento Zigbee tras caer un nodo [3]</i>	37
<i>Figura 2.16. Arquitectura Zigbee [3]</i>	37
<i>Figura 2.17. Bandas de frecuencia Zigbee [3]</i>	38
<i>Figura 2.18. Frame en la capa física [3]</i>	38
<i>Figura 2.19. Protocolos de seguridad en una frame Zigbee [3]</i>	39
<i>Figura 2.20. Arquitectura de una red LoRaWAN [22]</i>	43
<i>Figura 2.21. Arquitectura de una frame LoRa [22]</i>	43
<i>Figura 2.22. Arquitectura LoRaWAN [24]</i>	44
<i>Figura 2.23. Seguridad en LoRaWAN [25]</i>	44
<i>Figura 2.24. Interfaz IDE Arduino [27]</i>	49
<i>Figura 2.25. Funcionamiento de Hive MQ</i>	51
<i>Figura 3.1. Ejemplo básico de una red LoRa [32]</i>	52
<i>Figura 3.2. Estructura WBS</i>	55
<i>Figura 3.3. Plano general de la infraestructura</i>	57
<i>Figura 3.4. Ubicación del parking</i>	58
<i>Figura 3.5. Coordenadas geográficas</i>	58
<i>Figura 3.6. Coordenadas UTM</i>	58

<i>Figura 3.7. Acceso 1</i>	59
<i>Figura 3.8. Acceso 2</i>	59
<i>Figura 3.9. TTGO LoRa32 V1.0 [34]</i>	60
<i>Figura 3.10. Diagrama pin de TTGO LoRa32 V1.0 [34]</i>	61
<i>Figura 3.11. Gateway TTN más cercano</i>	62
<i>Figura 3.12. Distancia entre gateway y ubicación del entorno</i>	62
<i>Figura 3.13. Dragino LG01-P-868-EU [35]</i>	63
<i>Figura 3.14. Ejemplificación del prototipo demostrador</i>	67
<i>Figura 3.15. Monitor serie del end device, se observa el envío de los paquetes sincronizadamente...</i>	69
<i>Figura 3.16. Monitor serie del gateway, se observa la recepción de los paquetes sincronizadamente junto a su potencia recibida</i>	69
<i>Figura 3.17. Recepción de mensaje y envío mediante LoRa al gateway</i>	70
<i>Figura 3.18. Llegada del paquete LoRa al gateway</i>	71
<i>Figura 3.19. Datos para el acceso a HiveMQ</i>	72
<i>Figura 3.20. Código de la IDE de Arduino para la vinculación al Wi-Fi y bróker HiveMQ</i>	72
<i>Figura 3.21. Conexión a HiveMQ sin usuario ni contraseña</i>	72
<i>Figura 3.22. Suscripción al tópico HiveMQ desde la IDE de Arduino para subir la información</i>	73
<i>Figura 3.23. Suscripción al tópico HiveMQ para visualizar la información</i>	73
<i>Figura 3.24. Recepción de paquetes LoRa del end device y subida de estos a HiveMQ</i>	74
<i>Figura 3.25. Lectura de los paquetes LoRa en la base de datos HiveMQ</i>	74
<i>Figura 3.26. Instalación y funcionamiento de los módulos end device (izquierda) y gateway (derecha) LoRa</i>	75

Listado de tablas

<i>Tabla 2.1. Características Bluetooth Classic y BLE</i>	<i>28</i>
<i>Tabla 2.2. Mejoras Wi-Fi 6 sobre Wi-Fi 5</i>	<i>31</i>
<i>Tabla 2.3. Comparación de los estándares de comunicación WSN</i>	<i>46</i>
<i>Tabla 3.1. Ubicación del entorno</i>	<i>56</i>
<i>Tabla 3.2. Características de la extensión.....</i>	<i>57</i>
<i>Tabla 3.3. Especificaciones TTGO LoRa32 V1.0.....</i>	<i>60</i>
<i>Tabla 3.4. Características Dragino LG01-P-868-EU</i>	<i>63</i>
<i>Tabla 3.5. Datos del entorno de medición.....</i>	<i>68</i>
<i>Tabla 3.6. Relación distancia - potencia recibida.....</i>	<i>68</i>
<i>Tabla 3.7. Presupuesto del prototipo demostrador</i>	<i>77</i>
<i>Tabla 3.8. Presupuesto del prototipo inicial.....</i>	<i>77</i>

1. Introducció i Objectius

1.1. Presentació i context

La redacció de este projecte surge de la necessitat de realitzar un Treball de Fin de Grau per a l'obtenció del **Grado de Ingeniería de Sistemas y Servicios de Telecomunicaciones** en la Escola Tècnica Superior de Enginyeria de la **Universitat Rovira i Virgili**, amb la col·laboració de la empresa **ETECNIC** per a donar suport i oferir la determinació en la que basar el TFG. Pese a tractar-se d'un projecte acadèmic, amb les limitacions que ell implica, se ha tingut en compte les normatives vigents i se han emprat dades reals en la mesura de lo possible. En altres casos, i davant la impossibilitat d'accedir a aquests dades reals, se han emprat dades fictícies, sempre senyalant-los adequadament i intentant utilitzar informació acorde amb el tipus d'instal·lació i l'entorn.

El estudi se enmarca en la **conectivitat a Internet de múltiples sensores** de carregadors per a vehicles elèctrics en un espai subterràneu (parking de vehicles elèctrics), per a així, gestionar la informació de aquests sensors en una base de dades en la núvol. A més, aprofitant la mateixa connectivitat, se pretén que l'usuari pugui tant accedir a Internet com interactuar amb l'aplicació mòbil de la empresa.

En conseqüència, *ETECNIC* lo que busca es adaptar un servei que ja ofereix en espais exteriors a entorns tancats amb una baixa cobertura i difícil interacció entre dispositius *wireless*. Per aquest motiu, l'entorn de treball ofert per a estudiar i crear un desplegament de red inalàmbrica es un parking subterràneu. El parking ha de tenir més d'una planta amb un mínim de cent places de aparcament per a cada una, que les parets i les pilars siguin de material robuste on no permetin cap tipus de perforació per a passar cablejat entre elles i que contingui places adaptades a la càrrega de vehicles elèctrics.

Los sensors per a carregadors de vehicles elèctrics ja estaran instal·lats en l'entorn en el que se desplegarà la red *wireless*, funcionaràn correctament i se encarregaran de captar la informació del vehicle per a que sea transmesa a la base de dades. La base de dades la trobarem en la núvol i se irà actualitzant segons vagi arribant la informació dels sensors, l'accés a la informació de la base de dades se aconseguirà mitjançant l'aplicació mòbil instal·lada per l'usuari en el seu dispositiu. En aquesta aplicació l'usuari podrà tenir accés a la informació de se vehicle i realitzar altres accions que la empresa ofereix als clients que usen els carregadors i se donen d'alta en l'aplicació mòbil.

Com es pot esperar, per a la funcionalitat i monitorització de les accions mencionades en l'anterior paràgraf, serà necessari l'accés a una red amb sortida a Internet. Aquest motiu nos fa comprendre que volem desenvolupar un projecte basat en el **IoT** (Internet of Things), i que hauréem de conèixer les xarxes inalàmbrics que se troben dins d'aquest concepte tecnològic per a poder implementar la millor solució al servei que nos han demanat des de la empresa *ETECNIC*.

Es necesario aclarar que se trata de un trabajo académico, de modo que el proceso consistirá en el aprendizaje de un tema específico según la aplicación de lo aprendido durante el grado. Por lo tanto, partiendo del objetivo fundamental de realizar una red de sensores IoT y basándonos en un caso real dentro de una empresa especializada en la movilidad eléctrica, se requiere una investigación previa para posteriormente, según los conocimientos y aptitudes del alumno, proponer un prototipo adecuado que satisfaga, en la medida de lo posible, la misión principal del trabajo final de grado.

1.2. Misión y objetivos del trabajo

La misión u objetivo fundamental del trabajo consistirá en **realizar un estudio exhaustivo de las principales redes *wireless***, compatibles en un entorno adverso y capaces de realizar la comunicación entre los sensores de cargadores a la nube mediante una correcta salida a Internet.

Descartamos la investigación y el planteamiento de realizar un prototipo con redes cableadas porque básicamente requieren de una instalación fija y muy elaborada, presentan mucha incompatibilidad entre diferentes sistemas, son más costosas y no permiten movilidad ni flexibilidad en el entorno de trabajo. Estos inconvenientes nos obligan a enfocarnos en las redes inalámbricas como opción más certera para un proyecto de IoT.

Una vez consigamos tener el estudio de viabilidad de las diferentes redes *wireless*, deberemos comparar y escoger el/los estándar/es adecuado/s para el servicio que nos han demandado desde la empresa. El protocolo escogido deberá presentar un gran alcance en las comunicaciones entre nodos, ser robusto a la hora de combatir interferencias y de fácil instalación. También, tendrá que ser compatible con las tramas que envíen/soliciten los sensores de cargadores y capaz de dar salida a Internet para que esas tramas puedan ser procesadas por la base de datos de la empresa.

Finalmente, una vez escogido el/los estándar/es compatible/s, basándonos en el estudio realizado, deberemos exponer un prototipo que dé solución al problema solicitado. El prototipo estará basado en un caso real para buscar ofrecer una solución IoT estandarizada y adaptada en los máximos parkings posibles, o incluso en entornos similares para su uso en otras aplicaciones. En función del estándar escogido, detallaremos el prototipo que realizaremos y lo ejemplificaremos con una simulación para reafirmar el uso de esta tecnología como solución al problema.

En la *Figura 1.1* mostramos el seguimiento que tendrá nuestro trabajo, se compone de tres aspectos. El primero, después de las múltiples reuniones que nos han inducido a las bases fundamentales del proyecto, es realizar una puesta en escena para marcarnos los objetivos principales que determinarán el transcurso del trabajo. El segundo y más importante, realizar un estudio de viabilidad de las redes *wireless* compatibles con el entorno de desarrollo.

Finalmente, crear un prototipo final que ejemplifique el uso de un protocolo *wireless* adecuado al problema.



Figura 1.1. Seguimiento del trabajo final de grado

1.3. Puesta en escena al objetivo del estudio

La primera cuestión que me planteé cuando *EETECNIC* me propuso hacer esta investigación, fue cómo *el Corte Inglés*, *Mercadona* o el *Hospital Joan XXIII* de Tarragona, consiguen tener cobertura a Internet en sus instalaciones subterráneas o parkings. Conocía de sobra estos lugares y había estado conectado a sus redes, pero nunca antes me había planteado como lo habían conseguido al tratarse de un entorno que presenta tantas dificultades de conexión.

Para saciar mi curiosidad y poder ampliar mi visión en el estudio con un ejemplo profesional, me puse en contacto con Carlos Sánchez Leiva, trabajador de operaciones dentro del departamento de Ingeniería Radio de Barcelona, en la compañía *Telefónica Móviles España (Movistar)*, con más de 20 años de experiencia en el sector y, además, docente en la ETSE de la URV. La reunión online tuvo lugar el día 10 de marzo del 2021, vía la aplicación Teams y se expusieron las inquietudes relacionadas con mi proyecto.

En la reunión se expuso el tema principal que me movió a contactar con el Sr. Sánchez: cómo se conseguía dar cobertura inalámbrica en espacios subterráneos en los lugares citados anteriormente. Como supuse desde un primer momento, estos lugares tenían cobertura gracias a la colaboración de empresas de telefonía como *Movistar*, anteriormente denominada *Telefónica*, que es aquella para la que nuestro entrevistado trabaja.

El Sr. Sánchez era conocedor de que, a uno de esos lugares, *el Corte Inglés* de Tarragona, se les había dado cobertura a Internet en el año 2017 desde la empresa *Telefónica Móviles*. Me consiguió facilitar documentación relativamente actual junto a sus explicaciones, pude entender detalladamente cómo se desplegó la conexión en ese edificio y cómo lo hacían en otros espacios similares a este.

Básicamente, la obra se realizó desplegando la fibra óptica (GPON) que tenían en un punto cercano de Tarragona, hasta las coordenadas donde se encontraba el establecimiento en el que se iba a realizar la infraestructura. Una vez la fibra se desplegó hasta el lugar, se repartió mediante una instalación premeditada por las diferentes plantas del edificio. Se utilizaron equipos especializados para fibra óptica, tales como un RBS para dar cobertura global, puntos de alimentación, tres repetidores para amplificar la señal vía fibra, y finalmente, divisores y antenas desplegados por todas las plantas del edificio. De esta manera se consiguió que en todos los lugares del establecimiento se asegurara una cobertura aceptable y fiable.

La arquitectura de la instalación requiere el uso de tres tecnologías diferentes para poder satisfacer las necesidades de conexión. Sin necesidad de entrar en mucho detalle, vamos a explicar cómo está distribuida la instalación y así, poder hacernos una idea de cómo se garantiza la cobertura en esos espacios:

- Las antenas emiten y reciben la señal a través de ondas de RF. Estas van conectadas a través de cables radio (**tecnología ERICSSON**) al repetidor correspondiente de su planta.
- Una vez en el repetidor, se cambia cambiamos de tecnología, en este caso conectándose desde los repetidores al RBS vía **fibra óptica**.
- En el RBS, aparte de poderse realizar la conexión vía fibra entre los repetidores y la red de distribución, también es posible dar cobertura al exterior del edificio vía **RF**. Ya que el dispositivo se localiza en la azotea, al estar dotado de largo alcance, éste puede dar cobertura a los usuarios del exterior, tal y como harían las antenas dentro de la edificación.

La cobertura que ofrecida corresponde a la banda de **telefonía móvil**. Las antenas permiten nodos de conexión 3G/4G/5G dependiendo del terminal que tenga el usuario. Gracias a esto, además de poder recibir llamadas y tener conexión de datos móviles en un dispositivo, la empresa propietaria del establecimiento puede equipar su propia red interna de conexión a Internet, es decir, permitir dar un servicio completo de Wi-Fi en todos sus espacios, tanto a sus empleados como a los clientes.

Como segundo y último tema de la reunión, me surgió la duda mientras el Sr. Sánchez explicaba, qué pasaría si se diera el caso de que no se pudiera desplegar fibra óptica hasta el lugar donde queremos hacer la instalación.

El entrevistado me aclaró que la solución a este problema debe ser en primer lugar desplegar fibra óptica, siempre que sea posible, por el buen servicio que se da al cliente. En caso de no poder ser así, se daría cobertura celular vía radio o satélite a base de ondas de RF desde la estación o torre de emisión (situada a kilómetros de distancia) hasta el establecimiento, para así, asegurar una señal óptima en el lugar de despliegue de la red. Una vez llegue una buena señal de cobertura a la zona, mediante antenas internas y externas, se da cobertura a todo el establecimiento.

En este último caso, se utilizan otro tipo de tecnologías diferentes a las expuestas basadas en fibra óptica a la hora de satisfacer las necesidades solicitadas por la empresa. A pesar de ser una solución principalmente para dar cobertura de telefonía móvil, gracias a este despliegue, también habilita la posibilidad de dar cobertura a Internet vía Wi-Fi, en caso de que se demandara este servicio.

Como conclusión de la reunión, se observa que el tipo de cobertura difiere según si el espacio es cerrado o abierto y que en este caso una sola tecnología de transmisión no vale para dar cobertura a un espacio, sino que se necesita una combinación de varias para conseguir el objetivo deseado.

Este ejemplo nos ayuda a entender que, para poder desplegar cualquier tipo de red en un establecimiento cerrado, este debe presentar una buena cobertura inalámbrica en todas sus zonas si se quiere facilitar el despliegue de una infraestructura inalámbrica vía Wi-Fi u utilizando otros protocolos. De no ser así, difícilmente se podrá transmitir/recibir la información hacia/desde el exterior, y dar un servicio adecuado que cumpla los estándares de IoT a los usuarios que quieran conectarse a la red para satisfacer sus necesidades.

A la hora de buscar un protocolo, en primer lugar, deben cumplirse los requisitos del lugar en el que deseemos desplegarlo y buscar cubrir las dificultades marcadas por el entorno que se nos propone. Una vez conocido el lugar de desarrollo, debemos reconocer los diferentes estándares WSN actuales que creamos que mejor se adaptan a él. Por eso, antes de empezar a planificar el proyecto debemos estudiar la actualidad de los protocolos WSN y tomar una decisión sobre cuál/es son los adecuados según las características que estudiaremos en el siguiente apartado.

2. Antecedentes

2.1. Redes inalámbricas de sensores (WSN)

Para el desarrollo de mi estudio, las mejores redes que se pueden utilizar para dar conectividad a Internet son las denominadas redes de sensores inalámbricos (en inglés, Wireless Sensor Networks, **WSN**). Otro tipo de red que no fuese sin cable o *wireless* no cubrirá los requisitos expuestos en el apartado anterior. Además, este tipo de tecnología se encuentra en pleno auge por la gran flexibilidad que da a la hora de comunicar los datos y su fácil instalación a la hora de carecer de obra civil.

Los datos en estas redes se propagan en un medio de transmisión no guiado mediante ondas electromagnéticas y antenas. Esto hace que los usuarios tengan facilidad a la hora de moverse por este entorno y se minimice el uso de cableado. Se considera una de las tecnologías clave para implementar el Internet de las cosas o **IoT**.

Las redes de sensores son a menudo bidireccionales, permitiendo configurar los dispositivos sin necesidad de actuación humana, enviar comandos o actuar sobre el ambiente. En este último caso, se les conoce como **WSAN** (Wireless Sensors and Actuator Networks) [1].

Las WSN suelen estar formadas por un grupo de sensores con ciertas capacidades sensitivas y de actuación inalámbrica, los cuales permiten formar redes *ad hoc* o, también denominadas, distribuidas. Este tipo de redes se refieren típicamente a cualquier conjunto de redes donde todos los nodos tienen el mismo estado dentro de la red y son libres de asociarse con cualquier otro dispositivo de red *Ad hoc* en el rango de enlace. Las redes *Ad hoc* se refieren generalmente a un modo de operación de las redes inalámbricas IEEE 802.11 [2].

2.1.1. Características generales

A continuación, vamos a describir las características de las redes WSN. Entraremos en profundidad y perspectiva dentro de esta tecnología desglosando la información en apartados, puesto que esta información es de fundamental consideración para poder finalizar con los estándares WSN y elegir el más idóneo para nuestro proyecto o prototipo.

Anteriormente hemos mencionado algunos aspectos de dicha tecnología, pero no hemos introducido las principales características por las que he decidido fijarme en esta tecnología de red y no en otras. Las principales características de WSN son:

- Se trata de una red de sensores **distribuidos espacialmente**, capaces de **monitorear** las condiciones físicas o ambientales de un lugar específico (temperatura, concentración de gas, presión, humedad...) y con **autonomía** tanto energética como humana.
- Los sensores en un WSN se pueden utilizar tanto para **detectar** como para **pasar los datos** de otros sensores a una ubicación principal donde la información se

transfiere/recibe a una red más grande o donde se encuentra el **nodo de gestión de la red**.

- Cada nodo de la red tiene **varias partes**: transceptor de radio (con una antena), un microcontrolador, un circuito electrónico que interactúa con los sensores y una fuente de energía (batería o algún circuito de recolección de energía).
- **Los nodos se pueden colocar en una posición fija o pueden ser móviles**.

Los objetos inalámbricos conectados a una red se direccionan como nodos. Cuando integran un sensor, se direccionan como nodos de sensores inalámbricos, conformando una red de sensores inalámbricos. Por lo general, estos nodos suelen presentar un estilo común de topología, aunque pueden variar según el estándar que se utilice. En la siguiente figura podemos observar un ejemplo en bloques de un nodo *wireless*.

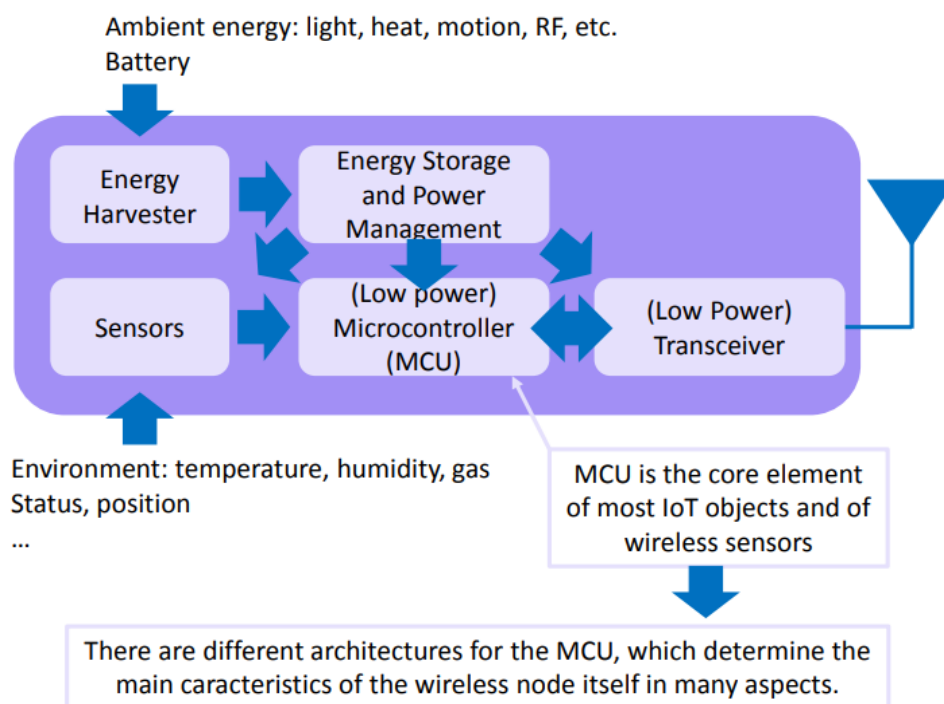


Figura 2.1. Diagrama de bloques general de un nodo WSN [3]

El *hardware* del nodo sensorial está formado por seis subsistemas básicos:

- *Energy Harvester*: A través de la captación de señales del ambiente se consigue generar energía para alimentar a la batería.
- *Energy Storage o Batería*: Encargado de alimentar al MCU, los sensores y el transductor a través de la energía almacenada.
- *Sensores*: Captan la información del entorno y la transmiten al MCU.
- *MCU*: Es el elemento principal del nodo, en cada nodo varía según la función que se le desee dar al sensor. Se encarga de procesar toda la información y es donde encontramos el *software* de funcionamiento del nodo.

- *Transceptor*: Es el dispositivo que se encarga de transformar o convertir una determinada manifestación de energía de entrada, en otra diferente de salida entre la antena y el MCU.
- *Antena*: Transmite y recibe las ondas electromagnéticas del entorno para establecer la comunicación entre nodos.

2.1.2. Aplicaciones

Aunque nuestro estudio se centre en la búsqueda de una WSN orientada a dar cobertura en un espacio subterráneo, es importante **revisar** algunas **aplicaciones** que se han realizado con este tipo de redes, teniendo en cuenta principalmente la transmisión de datos en tiempo real que es una de las características más importantes y necesarias en este tipo de tecnologías.

Las variables que inciden en la transmisión de datos se comportan en algunos casos de forma diferente de acuerdo con el espacio donde se apliquen. No es lo mismo enviar una trama o paquete por el espacio abierto que dentro de un edificio, puesto que hay interferencias importantes que se pueden presentar en cada caso.

Las aplicaciones más relevantes son:

- *Eficiencia energética*: Gracias a las redes de sensores se puede controlar el uso eficaz de la electricidad dentro de una ciudad, detectando el consumo elevado e injustificado de una calle o vivienda y corregirlos si fuera necesario.
- *Alta seguridad*: Existen lugares donde se precisen altos niveles de seguridad para evitar ataques terroristas, tales como centrales nucleares, aeropuertos, edificios del gobierno, etc. Aquí gracias a una red de sensores de detección se pueden controlar escenarios que con una simple cámara sería insuficiente.
- *Medioambiental y agrícola*: El control ambiental de hectáreas de boscosas o de grandes extensiones oceánicas, sería imposible sin las redes de sensores. Se puede hallar un sinnúmero de información ambiental, como temperatura, humedad, fuego, actividad sísmica. Estas mediciones ayudan a expertos a diagnosticar o prevenir un problema de urgencia y además minimiza el impacto ambiental de la presencia humana.
- *Industrial*: Dentro de las fábricas existen complejos sistemas de control de calidad o de tamaño. Los sensores industriales son idóneos para detectar estos errores de proceso y sacarlos fuera de la producción de la fábrica.
- *Automoción*: Las redes de sensores son el complemento ideal a las cámaras de tráfico, ya que pueden informar de la situación del tráfico en ángulos muertos que no cubren las cámaras y también pueden informar a conductores de la situación de la carretera en caso de atasco o accidente.

- *Medicinal*: La reducción del tamaño de los nodos sensores permite mejorar la calidad de vida en los hospitales, ya que implantando un sensor del tamaño de un simple chip a pacientes los médicos pueden controlar sus constantes vitales (pulsaciones, presión, nivel de azúcar en sangre, etc.) y darles el tratamiento apropiado.
- *Domótica*: Integrando un conjunto de redes de nodos de sensores, se puede asegurar el concepto de “*Smart Building*” y otorgarle al usuario un aumento del confort, seguridad, eficiencia energética, facilidad de comunicación y ocio dentro de una misma vivienda.

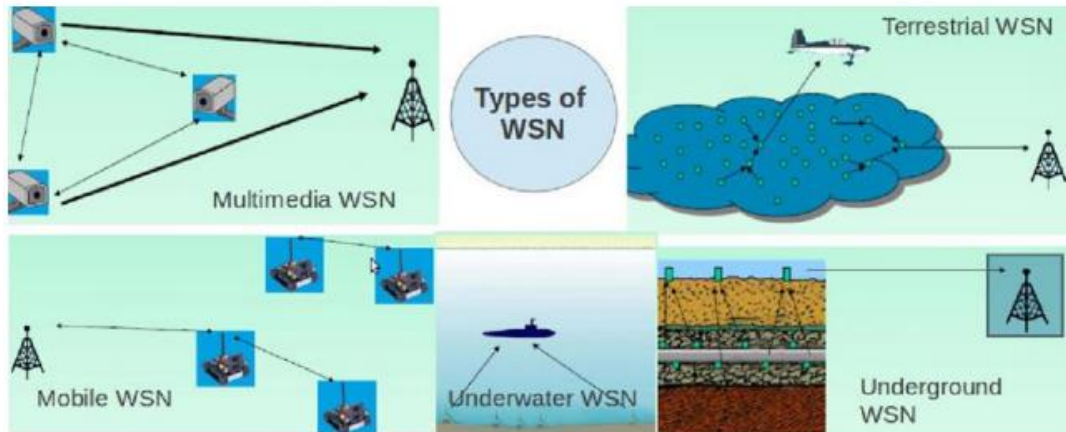


Figura 2.2. Aplicaciones WSN [4]

2.1.3. Topología

Una de las características más importantes de la tecnología WSN es la distribución que tienen los sensores dentro del área de conectividad o de trabajo. Como en nuestro estudio nos vamos a encontrar con un número elevado de sensores dentro de un área específica, como norma general, se procurará que sean **autónomos** y **autoorganizados**. En la siguiente figura mostramos un ejemplo común de infraestructura WSN. [5]

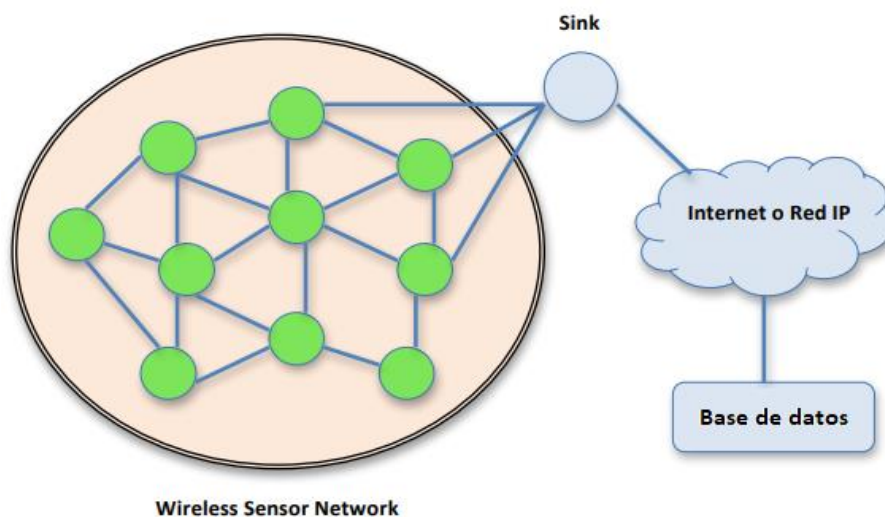


Figura 2.3. Ejemplo infraestructura WSN [5]

Como se puede observar, tenemos una red de nodos, que está conectada inalámbricamente entre sí dentro de una misma área de cobertura. Esa red se comunica con un punto central de gestión de información o *sink* (dependiendo de la infraestructura puede haber más de uno), que se encarga de dar cobertura, gestionar la información de la red de nodos y comunicar esta información a Internet o la red IP. Coloquialmente, se podría decir que el nodo *sink* es el cerebro de la red. Finalmente, desde Internet podremos acceder a esa base de datos donde hallamos la información proveniente de los sensores.

Como hemos podido ver en el ejemplo, la red de nodos se comunica al punto central *sink* mediante uno, dos, o incluso tres sensores enrutadores o *routers*. Estos *routers*, además de ser el vehículo de conexión entre nodos terminadores y el *sink*, también determinan la ruta de comunicación entre los nodos, y proporcionan y/o almacenan las claves de seguridad de los dispositivos de red para tener un mayor control de seguridad.

Por ello, dependiendo del uso que se le quieran dar a los sensores, existen diferentes tipos de topologías que cumplen con el requisito de proporcionar la información, pero que presentan diferencias a la hora de comunicarse entre los nodos sensoriales, los enrutadores y el *sink*. A continuación, procederemos a explicar con detalle las topologías más destacadas:

- **No estructurada:** En ella todos los nodos tienen la misma función y no presenta ningún orden estructurado entre su comunicación de nodos y coordinadores. La *Figura 2.4* muestra un ejemplo de este tipo.

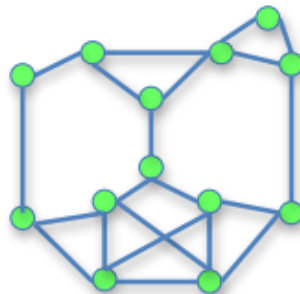


Figura 2.4. Topología no estructurada [5]

- **Estrella:** En esta topología tenemos un dispositivo central o coordinador que controla, gestiona y da conexión al resto de *end devices*. Es de las topologías más simples y efectivas. El principal problema es que si cae el coordinador central cae todo el sistema.

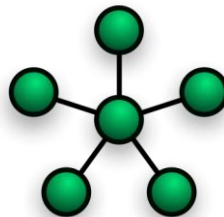


Figura 2.5. Topología estrella [6]

- **Bus:** Se basa en un canal central, el cual lleva la información a todos los *end devices*, en forma de ramificaciones, de modo, que ésta viaja de manera secuencial hacia los

nodos de la red. Su desventaja se basa en su distribución secuencial de datos, por lo que si se interrumpe el canal central, la red queda inutilizada. En la actualidad se encuentra en desuso, ya que solo permite un número limitado de nodos si no queremos saturar el canal.

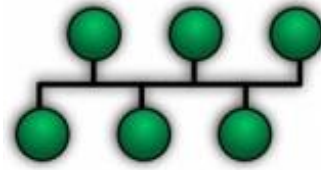


Figura 2.6. Topología bus [6]

- **Árbol:** Es una topología con un coordinador, varios enrutadores o *routers* y *end devices*, organizados de manera jerárquica. Los coordinadores y *routers* pueden preceder como “padres” a otros *routers* y nodos. El dispositivo conectado a un nodo es su “hijo”, entonces, todos los mensajes dirigidos a él tienen que ser encaminados a través de su “padre”. El principal problema es que, si falla un nodo padre, el hijo no recibe datos y a partir de él toda su “descendencia” queda deshabilitada. Normalmente la topología de árbol parte de un nodo *sink* o coordinador, como en el ejemplo de a continuación.

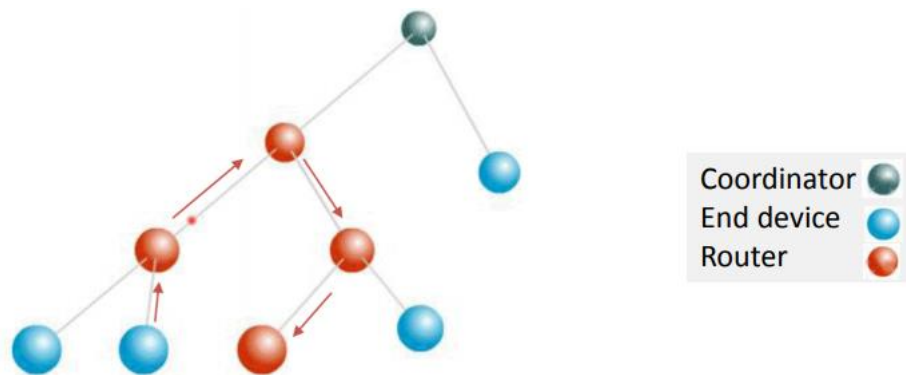


Figura 2.7. Topología árbol [3]

- **Mesh:** Es muy similar a la topología del árbol, pero sin la estructura jerárquica, permitiendo que las reglas de comunicación sean más flexibles. Las rutas se establecen de forma dinámica y puede haber conexión entre enrutadores vecinos, por lo que la red es más eficiente a la hora de propagar los mensajes, ya que posibilita la creación de rutas alternativas frente a la congestión del tráfico y las desconexiones de los nodos. El inconveniente principal es que esta topología es más compleja a la hora de saber cómo se gestiona y distribuye la información entre ella. El nodo coordinador de la imagen hace referencia a un nodo *sink*.

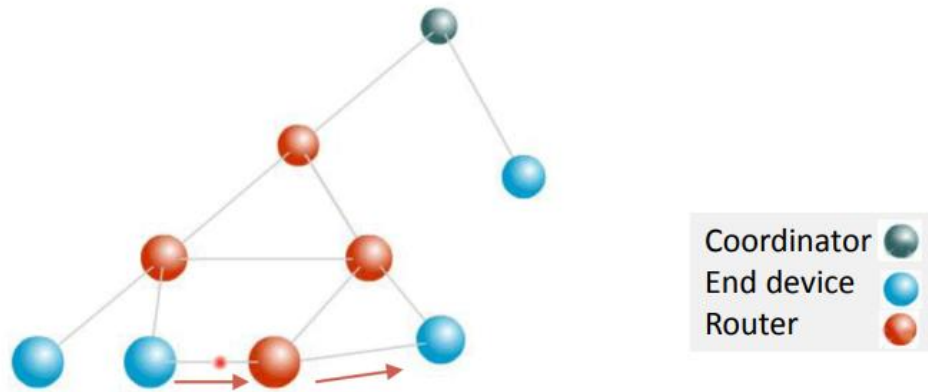


Figura 2.8. Topología mesh [3]

- **Basada en clusters:** Este tipo de topología es la más utilizada y es idónea para cuando hallamos múltiples nodos dentro de una gran extensión física. En la imagen se identifica como están distribuidos los dispositivos, básicamente hay tres elementos. Los *end devices*, que se encargan de medir y captar la información del entorno, los *routers* o *clusters* que reciben y transmiten la información de los sensores entre ellos y, finalmente, el coordinador o *sink* que se encarga de recibir y procesar la información recibida por los *clusters* con el fin de que el usuario pueda tratar con ella.

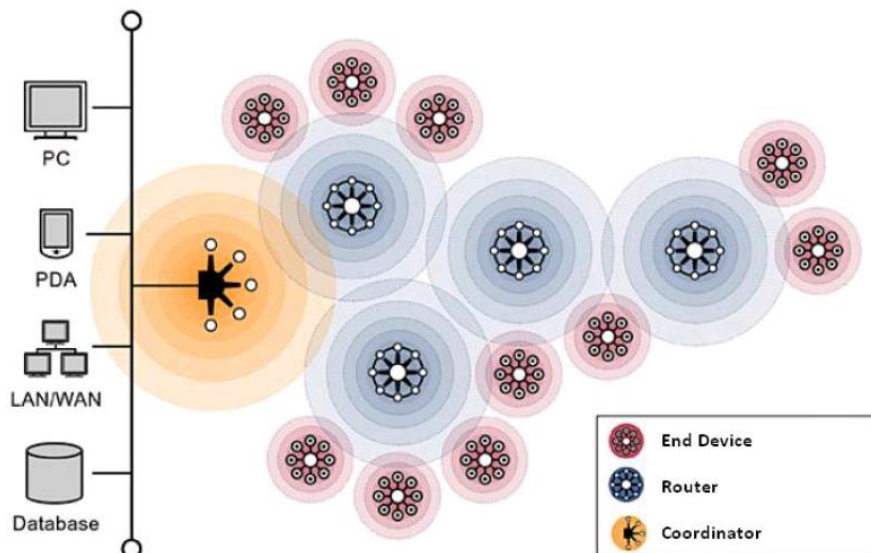


Figura 2.9. Topología cluster [3]

Podemos encontrar más topologías, como la de anillo, línea o *fully connected*. No obstante, de cara a este estudio hemos decidido centrarnos en las anteriores porque son las más relevantes y, además, algunas de ellas serán las que usaremos para nuestro prototipo final por consumo, seguridad y control de comunicación.

2.1.4. Variabilidad del canal

Como ventaja principal, una WSN no tiene necesidad alguna de infraestructura para poder operar, ya que sus nodos pueden actuar de emisores, receptores o *routers* a la hora de transmitir la información por el aire. Para que esto sea posible, la información entre los nodos sensoriales

y el nodo *sink* debe ser transmitida por el mismo canal mediante ondas electromagnéticas, donde debemos tener en cuenta las posibles atenuaciones e interferencias que encontremos en el entorno de trabajo. [5]

El canal radioeléctrico de comunicaciones presenta una respuesta temporal variante, en el que existen una serie de fenómenos tales como el ruido, la atenuación, la distorsión y las interferencias que puede producir errores en los datos transmitidos. Si no tenemos en cuenta estos aspectos y no examinamos bien el área de cobertura haciendo nuestro estudio, podríamos encontrarnos en un escenario en que los nodos detectan bien la información del entorno, pero no se establece la conexión entre ellos. Por lo tanto, la información se pierde y no obtenemos ningún objetivo.

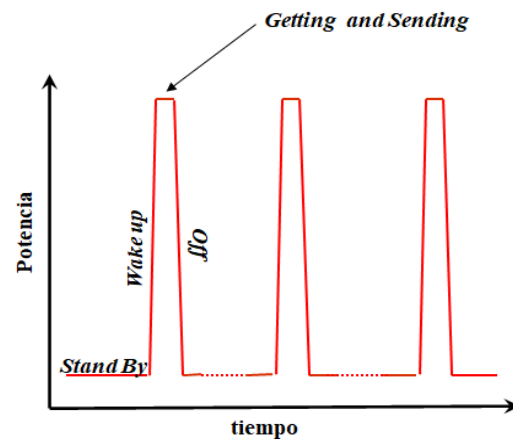


Figura 2.10. Estados de operación nodo en modo sleep [3]

2.1.5. Consumo energético

Otra característica a tener en cuenta es el factor de consumo de los sensores. El consumo debe ir ligado con la autonomía y la capacidad de proceso, ya que actualmente cuentan con una unidad de energía limitada. Un nodo sensorial o *end device* tiene que contar con un procesador de consumo ultra bajo, así como un transceptor radio que solo trabaje en momentos de clara necesidad, y además se debe disponer de un software que vele por este aspecto, disminuyendo todo lo posible el consumo.

Para conseguirlo debemos tener en cuenta aspectos de consumo como el “*energy harvesting*”¹ en los nodos sensoriales, que estos carguen la batería con las ondas de señal recibidas. O también, hacer que los nodos cuando no trabajen estén en modo “*sleep*”, es decir, que consuman la menor energía posible cuando no sean necesarios. Estos métodos son muy útiles a la hora de trabajar con nodos inalámbricos que no puedan estar conectados a una fuente constante de energía. [3]

2.1.6. Seguridad

El crecimiento en el uso y popularidad de las redes de sensores inalámbricos ha creado preocupación en lo que respecta a su seguridad. Los usuarios no quieren revelar sus datos a personas no autorizadas que podrían hacer un uso malicioso de la información divulgada. Esta preocupación es aún más relevante para entornos inalámbricos donde cualquier persona puede escuchar un mensaje enviado por RF.

¹ Concepto muy utilizado en los tags RFID. Es el proceso por el cual la energía derivada de fuentes externas es capturada y es almacenada para pequeños dispositivos autónomos inalámbricos.

Por lo tanto, incluso un útil y cómodo sistema podría no ser atractivo para los usuarios si no es seguro. Para abordar este problema, los investigadores en WSN han puesto en marcha varios protocolos de seguridad que se utilizarán en redes de sensores. Ejemplos de tales protocolos son TinySec y TinyECC. TinySec, una de la conocida capa de enlace de seguridad que utiliza CBC². [8] Así mismo, se están haciendo estudios para mejorar los protocolos de cifrado y los modos de operación MAC de tal forma que aumente la seguridad sin reducir el rendimiento de los nodos.

2.2. Estándares de comunicación WSN

Una vez introducidos en las características fundamentales de las redes WSN, debemos conocer y estudiar sus estándares más destacados. Analizando estos estándares y comparándolos entre sí, podremos encontrar el adecuado para dar comunicación a nuestra red de sensores en un parking.

Sabemos que para nuestro estudio necesitamos dar cobertura a un parking mediante redes **WLAN** o **WPAN**, ya que son idóneas para este tipo de áreas. Por lo tanto, los protocolos en los que nos vamos a centrar son el 802.15.1 (Bluetooth), el 802.11.ac (Wi-Fi) y el 802.15.4 (Zigbee).

Además, también estudiaremos una red **LPWAN** con el protocolo LoRaWAN. Aunque originalmente está diseñado para comunicaciones exteriores de largo alcance, la alta sensibilidad de los receptores LoRa y la menor atenuación de los obstáculos como paredes a estas frecuencias lo hacen una opción interesante para dar cobertura en un parking subterráneo.

2.2.1. Bluetooth (IEEE 802.15.1)

Bluetooth es una tecnología inalámbrica que se utiliza para transferir datos entre diferentes dispositivos electrónicos. La distancia de transmisión de datos es pequeña en comparación con otros modos de comunicación inalámbrica. En consecuencia, esta permite que los dispositivos electrónicos se comuniquen de forma inalámbrica entre sí creando una red WPAN.

Es un protocolo de enlace inalámbrico de microondas de alta velocidad y baja potencia, diseñado para conectar teléfonos, computadoras portátiles, PDA y otros equipos portátiles, resultando de esta forma casi transparente al usuario. [9]

En nuestro estudio nos podríamos haber decantado por **Bluetooth 4.0 Low Energy** (BLE), ya que encajaría perfectamente con los requisitos de partida propuestos en nuestro estudio. El

² Es un modo de operación es un algoritmo que utiliza un cifrador por bloques para proveer seguridad a la información, como confidencialidad y autenticidad. [7]

motivo principal por el que no lo hemos querido incluir es porque esta versión del protocolo está orientada al bajo consumo de los dispositivos.

En nuestro proyecto el consumo es relativamente poco importante, ya que básicamente nuestros sensores estarán conectados a una fuente de energía. Además, al utilizar BLE reduciríamos alcance de cobertura, que en la práctica es lo que menos nos interesa porque necesitaríamos un número superior de nodos en diferencia que con el Bluetooth clásico. A continuación, adjunto la tabla en la que baso mis afirmaciones.

	Bluetooth Classic	Bluetooth Low Energy
IEEE Standard	802.15.1	802.15.1
Frequency (GHz)	2.4	2.4
Max. raw bit rate (Mbps)	1-3	1
Typical data throughput (Mbps)	0.7-2.1	0.27
Max. range outdoor (m)	100	50
Relative power consumption	Medium	Very low
Battery Life	Days	Months to Years
Network Size	8 activos, 200 pasivos	Undefined

Tabla 2.1. Características Bluetooth Classic y BLE [9]

Bluetooth Classic se compone de un procesador de banda base, una radio y una antena. El procesador de banda base convierte los datos en señales, la radio emite según la frecuencia requerida y la antena envía la señal por radiofrecuencia o capta la señal de otro dispositivo bluetooth por el aire.

Utiliza una técnica de espectro ensanchado por salto de frecuencia (FHSS³), que es una de las dos técnicas de modulación básicas utilizadas en la transmisión de señales de espectro ensanchado. Las frecuencias se cambian repetidamente durante la transmisión de radio para ayudar a reducir el acceso ilegal de otros dispositivos no deseados o evitar que otros instrumentos telecomunicadores se crucen e interfieran en la señal.

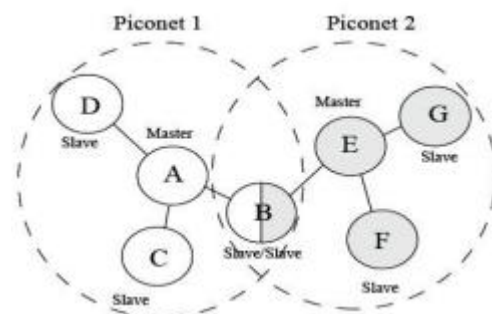


Figura 2.11. Scatternet Bluetooth [10]

³ Es una técnica de modulación en espectro ensanchado en el que la señal se emite sobre una serie de radiofrecuencias aparentemente aleatorias, saltando de frecuencia en frecuencia sincrónicamente con el transmisor.

En la tecnología Bluetooth se pueden conectar activamente hasta 8 dispositivos en configuración maestro/esclavo. Las piconets se pueden combinar para formar redes de dispersión, o *scatternet*, que proporcionan conectividad ilimitada de dispositivos pasivos o activos, para conseguir esto se forman un grupo de dispositivos conectados en forma *ad hoc*.

Una piconet comienza con dos dispositivos conectados y puede crecer hasta a ocho dispositivos. Todos los dispositivos Bluetooth son unidades iguales y tienen implementaciones idénticas, sin embargo, al establecer una piconet, una unidad actuará como maestra fija y las otras como esclavas mientras dure la conexión piconet. [10]

Arquitectura Bluetooth

La pila de protocolos de Bluetooth se puede dividir en dos componentes: el host y el controlador Bluetooth. El Host Controller Interface (HCI) proporciona una interfaz estándar entre el host y el controlador Bluetooth.

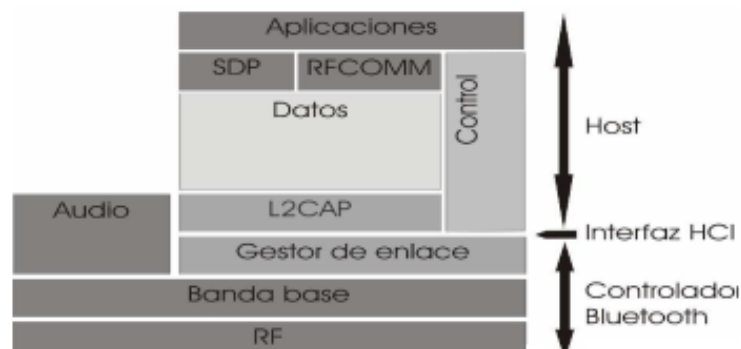


Figura 2.12. Arquitectura Bluetooth [10]

El *host* también es conocido como la capa alta de la pila de protocolos y generalmente se encuentra integrado con el software del sistema o sistema operativo del dispositivo.

El *controlador de Bluetooth* normalmente es un módulo hardware, como podría ser una PC card conectada al dispositivo en cuestión, aunque lo normal es que este módulo vaya ya integrado en el hardware del dispositivo.

Las capas altas pueden hacer uso del módulo radio a través de la *interfaz HCI*. A su vez, el módulo radio interactúa con el sistema host mediante un sistema estándar de entrada/salida, como puede ser PCMCIA, UART, o USB. [10]

Ventajas y desventajas de utilizar Bluetooth

Las principales **ventajas** de usar este protocolo son las siguientes:

- Puede transferir información de un dispositivo electrónico a otro sin línea de visión directa.
- Sus dispositivos son de muy bajo coste.

- Su movilidad es muy versátil, ya que no necesita ninguna toma de corriente o conexión a Internet para poder transferir datos.

En cambio, presenta una serie de **inconvenientes** como:

- Lenta transferencia de datos entre dos dispositivos en comparación con otros protocolos.
- Su seguridad es muy frágil, ya que la transferencia se realiza a través de ondas de radio. Esto supondría que un *hacker* puede acceder fácilmente a su sistema si no se consta de una vigilancia rigurosa.
- La máxima capacidad de dispositivos conectados a la vez es de ocho por piconet.
- Su incidencia a interferencias es muy elevada, ya que trabaja en la banda de frecuencias libre ISM de 2,4 GHz, la misma que Wi-Fi y Zigbee.
- El alcance máximo entre dos dispositivos es de 100 metros, mucho menor al de otros protocolos.

2.2.2. Wi-Fi (IEEE 802.11.ac)

El protocolo IEEE 802.11ac, es un estándar que permite la interconexión inalámbrica de dispositivos electrónicos. Los dispositivos habilitados con Wi-Fi (tales como ordenadores, *tablets*, *smartphones*, televisores, videoconsolas, etc.) pueden conectarse entre sí o a Internet a través de un punto de acceso (AP) de red inalámbrica. [11]

Wi-Fi es una marca de la *Wi-Fi Alliance*, la organización comercial que cumple con los estándares 802.11 relacionados con las WLAN. Se sabe que la última versión Wi-Fi, la 802.11ax o Wi-Fi 6, se encuentra actualmente (año 2021) en pleno desarrollo. [12]

El Wi-Fi 6 al ser una versión nueva y aprobada hace relativamente menos de un año, en el 2020, hemos decidido descartarla para nuestro estudio. La versión 802.11ac se encuentra más familiarizada con los dispositivos que vamos a trabajar y, además, en la última versión aún hay instrumentos, como *routers* y nodos terminadores, que no soportarían las mejoras del Wi-Fi 6 y se deberían de reciclar comprando otros dispositivos adaptados a esta nueva versión. Eso supondría un gasto más elevado e innecesario para nuestro estudio, ya que aún estamos hablando de una tecnología que se encuentra en pleno despliegue y las mejoras que nos ofrece no nos darían una funcionalidad superior en nuestro proyecto.

	Wi-Fi 5 (802.11.ac)	WIFI 6 (802.11ax)
BANDAS	5 GHz	2.4 y 5 GHz
ANCHO DE BANDA DEL CANAL	20, 40, 80, 80+80 y 160 MHz	20, 40, 80, 80+80 y 160 MHz
TAMAÑOS DE FTT	64, 126, 256, 512	256, 512, 1024, 2048

MODULACIÓN MÁS ALTA	256 QAM	1024 QAM
VELOCIDAD DE DATOS	433 Mbps por stream (80 MHz) Máx. 7000 Mbps	600 Mbps por stream (80 MHz) Máx. 10000 Mbps
OFDMA	OFDM en vez de OFDMA	Menor latencia y menos congestión con varios dispositivos
MU-MIMO	Solo para descarga	Simultánea de subida y bajada: mejor velocidad en redes con muchos dispositivos
TARGET WAKE TIME	No disponible	Mejor ahorro energético

Tabla 2.2. Mejoras Wi-Fi 6 sobre Wi-Fi 5 [12]

El WiFi es una tecnología RF que transmite información por el aire utilizando ondas de radio. Aunque los estándares anteriores del IEEE 802.11 se transmiten por la frecuencia de 2,4 GHz, para evitar las interferencias con otros protocolos que trabajan en la misma, la principal característica del **Wi-Fi 5** es que transmite en la frecuencia de 5 GHz, una frecuencia menos solicitada u ocupada.

Arquitectura de red Wi-Fi

En una conexión Wi-Fi tenemos un adaptador inalámbrico en un dispositivo que traduce los datos en forma de señal de radio y, a través de una antena, los transmite por el aire. Un *router* (AP), también inalámbrico, que es el que se encarga de recibir la señal y de decodificarla y envía la información captada a través de Internet a otros servidores mediante una conexión física por cable, normalmente a través de Ethernet. [13]

Evidentemente, todo este proceso ocurre exactamente igual, pero a la inversa, cuando es el usuario el que tiene que recibir información desde Internet. La frecuencia de operación es de 5 GHz, tal y como comentábamos anteriormente, sobre el estándar 802.11ac. [14] También, existe retrocompatibilidad entre estándares de adaptadores o AP, y en las últimas especificaciones se puede transmitir de forma simultánea en dos bandas diferentes (2,5 y 5 GHz) para lograr una mayor velocidad en la transferencia de datos. En la siguiente imagen mostramos el ejemplo de una topología Wi-Fi con repetidores AP.

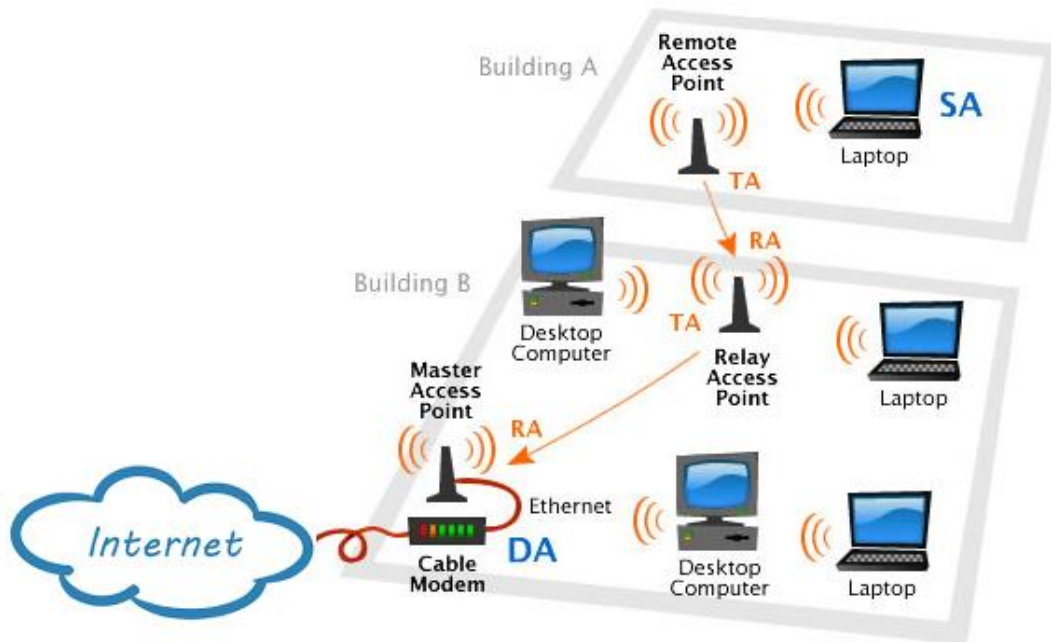


Figura 2.13. Ejemplo de una infraestructura Wi-Fi

Existen dos modos para configurar la arquitectura de una red inalámbrica: *ad hoc* e infraestructura. En el modo *ad hoc*, los dispositivos transmiten directamente punto a punto, mientras que en el modo infraestructura, los dispositivos se comunican a través de un punto de acceso que sirve de puente a otras redes.

- *Modo ad hoc*: Es un modo que funciona muy bien para pequeñas áreas y un número pequeño de dispositivos. Cuando se utiliza el modo ad hoc, todos los dispositivos de la red inalámbrica se comunican directamente entre sí, de igual a igual, en el modo de comunicación punto a punto. La red no tiene ninguna estructura o puntos fijos. No se requiere ningún punto de acceso para la comunicación entre dispositivos. [15]
- *Modo infraestructura*: Es el modo más común. Todos los dispositivos están conectados a la red inalámbrica con la ayuda de un punto de acceso (AP). Los puntos de acceso inalámbricos son generalmente *routers* que pasan los datos de la red inalámbrica a datos en una Ethernet cableada. La conexión de varios puntos de acceso a través de una red troncal Ethernet por cable puede extender aún más la cobertura de la red inalámbrica permitiendo que un dispositivo móvil salga del margen de cobertura de un punto de acceso y entre en el de otro. Consecuentemente, los clientes inalámbricos pueden moverse libremente del dominio de un punto de acceso a otro y seguir manteniendo la conexión de red sin cortes. [15]

Seguridad y cifrado

El problema de las redes inalámbricas Wi-Fi viene determinado por su seguridad, transmiten los datos en todas las direcciones a cualquier dispositivo que pueda estar escuchando, dentro de un rango limitado. Son vulnerables a las escuchas, por lo tanto, este tipo de redes requieren

un esfuerzo más centrado para mantener la seguridad en autenticación⁴, confidencialidad⁵ e integridad⁶.

El cifrado es opcional en las WLAN, pero sin él, cualquier dispositivo compatible con el estándar dentro del alcance de la red puede leer todo su tráfico. En función del cifrado que se aplique a las comunicaciones entre el *router* y los adaptadores inalámbricos, existen varias opciones:

- *WEP (Wired Equivalent Privacy)*: En el año 1999 era el más utilizado en el mundo, pero como es lógico, su uso ha ido disminuyendo con el paso de los años. En su llegada se descubrieron muchos fallos y agujeros de seguridad, lo que hizo que tuviera que ir mejorando. Aunque los principales fallos y agujeros fueron solucionados, lo cierto es que se trata de un cifrado poco fiable y fácil de explotar. [15]
- *WPA (WiFi Protected Access)*: WPA fue la respuesta a los principales fallos y vulnerabilidades de WEP. Las claves usadas por WPA son de 256 bits, a diferencia de los 128 bits usados por WEP, aunque no es la única mejora, ya que incorpora la comprobación de contenidos e integridad de mensajes para evitar que puedan ser interceptados y el uso del protocolo de clave temporal TKIP, ayuda a que un *router* no pueda ser atacado fácilmente como ocurría con WEP. [15]
- *WPA 2*: La principal diferencia con WPA es el uso del AES, que realiza un cifrado por bloques para permitir claves más largas y seguras, también se le añadió la implementación del CCMP, un protocolo mejorado de encriptación que sustituye a TKIP. [15]
- *WPA 3*: Incorpora el cifrado de 192 bits en vez de 129 bits, lo que hace que el cifrado sea más seguro y difícil de romper. Esto hace que sea más seguro incluso con contraseñas débiles, por lo tanto, una misma clave es más vulnerable a ataques en WPA2 que en WPA3. Cuenta también con un nuevo modo de configurar y conectar a la red dispositivos sin la necesidad de que tengan pantalla ni botones físicos. [16]

Una vez introducidos estos cifrados, el Wi-Fi 6 mejora considerablemente la confidencialidad y la integridad, ya que incorpora el estándar de seguridad WPA3. Se trata de un estándar que ofrece una mayor protección que sus predecesores como hemos podido leer y mejora las prestaciones del WPA2 que es con el que viene implementado el Wi-Fi 5.

⁴ Garantizar que los nodos son quién y lo que dicen ser.

⁵ Asegurar que los intrusos no pueden leer el tráfico de red.

⁶ Asegurar que los mensajes son entregados sin alteración.

Ventajas y desventajas de utilizar Wi-Fi (*IEEE 802.11.ac*)

Las redes Wi-Fi tienen una serie de beneficios clave como la gran cantidad de bloques de datos que pueden transmitir en formato video o imagen. Eso la hace bastante más singular respecto a otros protocolos que envían la información con una tasa más baja. A continuación, en una lista resumimos algunos **beneficios** de usar este estándar:

- Son redes baratas, sencillas y fáciles de instalar.
- Tiene un alcance entre 70 y 250 metros.
- Ofrecen una gran tasa de datos de transferencia a una velocidad bastante aceptable en comparación a la cantidad de información que se realiza en una transferencia (6,9 Gbps).
- Integración rápida y fácil de los dispositivos en la red, y una alta flexibilidad al modificar una instalación.
- Libertad de cambiar su ubicación sin perder la conexión en el área de cobertura.
- Al ser un estándar tan versátil y muy común en todo el mundo gracias a la regulación de la Wi-Fi Alliance, su compatibilidad permite ofrecer nuevos productos o servicios actualizados según avanza la tecnología.
- Al trabajar en la banda de 5 GHz transmite con menos interferencias a otros protocolos que trabajan en la banda de 2,5 GHz.

Por otra parte, existen también ciertos **inconvenientes** al trabajar con Wi-Fi:

- Su seguridad, probablemente es la más perjudicial de todas al estar tan globalizado. Es de los protocolos más inseguros, ya que si no se tiene mucho control es bastante fácil de *hackear* y penetrar en su red para alterar las comunicaciones.
- No garantiza una alta disponibilidad, si se conectan muchos dispositivos a un mismo AP se suele congestionar rápidamente la red y las transmisiones quedan estancadas.
- Se pueden sufrir interferencias si existen otras redes inalámbricas que trabajen en la misma banda o bien cuando otras fuentes de señales de radio estén presentes.
- En algunos edificios conseguir una cobertura consistente puede ser difícil, lo que conlleva a la aparición de puntos negros donde no hay cobertura en ciertas zonas del área.
- Presentan una latencia considerable a la hora de acceder a una red, produciendo en ocasiones retrasos en la conexión junto a pequeñas pérdidas de ella a la hora de transferir datos.

2.2.3. Zigbee (IEEE 802.15.4)

Zigbee es un estándar de comunicaciones inalámbricas de RF de muy bajo coste y consumo que permite el funcionamiento de redes muy simples y eficientes, para conectar sensores y dispositivos. Está basada en el estándar 802.15.4 del IEEE que regula las redes WPAN.

Las principales características de este protocolo son que las transmisiones son seguras a pesar de que el número de redes diferentes que convivan sea elevado, la cantidad de información por transmisión es pequeña y presentan un bajo consumo energético. [3]

Las redes Zigbee han captado nuestra atención para este estudio porque presentan una serie de características acordes a lo que nosotros buscamos y queremos en nuestro parking subterráneo. Estas singularidades son:

- Las redes son muy eficientes, simples y baratas.
- Apropriadas para montar redes donde el uso del cable es difícil y/o costoso.
- De fácil instalación, flexibles y modulares (capacidad de ampliación sin perder el servicio).
- Inteligencia propia para la configuración de la red y el encaminamiento de los mensajes.
- Funcionan en modo *full duplex*⁷.
- Uso libre, no requieren de licencia.
- Debido al gran número de aplicaciones, los dispositivos Zigbee son "altamente compatibles" entre los diversos fabricantes.

La estructura de red Zigbee es muy parecida a la estructura de una WSN explicada anteriormente. Contiene tres dispositivos diferentes, un nodo coordinador o *sink*, nodos enrutadores o *routers* y nodos terminadores o *end devices*. Cada uno se caracteriza por:

- *Coordinador*: Tiene que haber uno por red, gestiona la red determinando las rutas entre los nodos y suministra, además de almacenar, las claves de seguridad de los diferentes dispositivos.
- *Enrutador*: Interconecta los dispositivos separados más de un salto mediante técnicas de enrutamiento y direccionamiento. Son dispositivos que se encargan de enviar, recibir y enrutar la información.
- *Terminador*: Dispositivo que responde a las peticiones de los *routers* y/o coordinadores. Presenta funcionalidades mínimas y casi siempre está en modo '*sleep*', tiene poca memoria, es muy económico y hace las funciones de sensores y actuadores.

⁷ Es un término utilizado en telecomunicación para definir a un sistema que es capaz de mantener una comunicación bidireccional, enviando y recibiendo mensajes de forma simultánea.

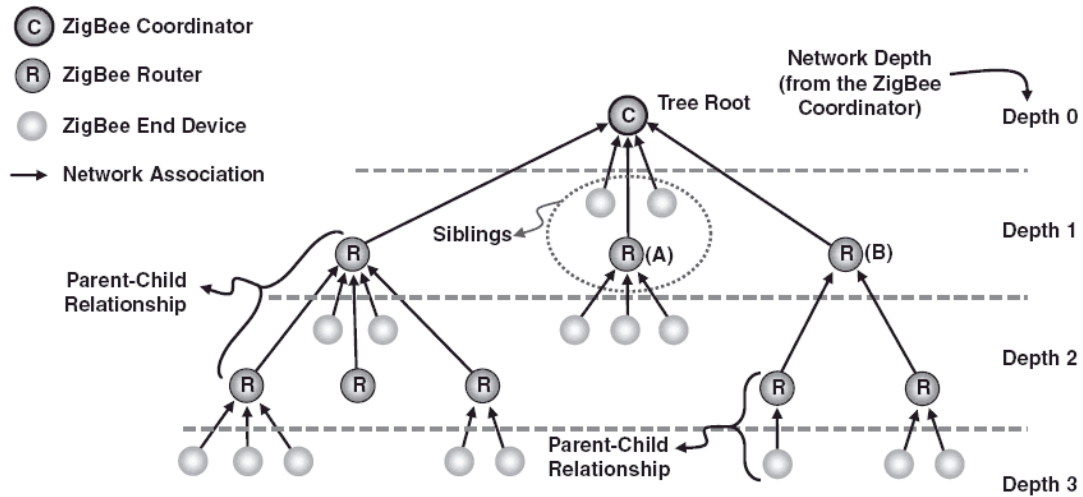


Figura 2.14. Ejemplo de estructura árbol Zigbee

Funcionalidad y direccionamiento

Según la funcionalidad de los dispositivos puede haber dos modelos distintos:

- *Funcionalidad completa o Full Function Devices (FFD)*
El nodo es activo (*router* o coordinador) y puede emitir autónomamente, dispone de memoria adicional y capacidad computacional. [3]
- *Funcionalidad reducida o Reduced Function Devices (RFD)*
Son nodos pasivos (*end devices*) que solamente son capaces de responder a peticiones con funcionalidades limitadas, bajo coste y gran simplicidad. [3]

El tipo de direccionamiento viene controlado por:

- Una red de 16 bits (2^{16} dispositivos): efectuado por el coordinador que tendrá la dirección 0.
- IEEE (MAC, 64 bits): cada dispositivo tiene su propia dirección MAC.
- En una red Zigbee se produce un proceso de unión o separación de la red de forma dinámica. Este proceso se le comunica al coordinador.

Contiene una tarea basada en crear una tabla de enlaces lógicos entre nodos que se relacionan entre sí, para que en caso de caer un nodo poder restablecer el camino por otros nodos contiguos y no perder la conexión. Esto es posible si los nodos están radioeléctricamente conectados y tienen la funcionalidad FFD. En el ejemplo de a continuación lo veremos más detallado.

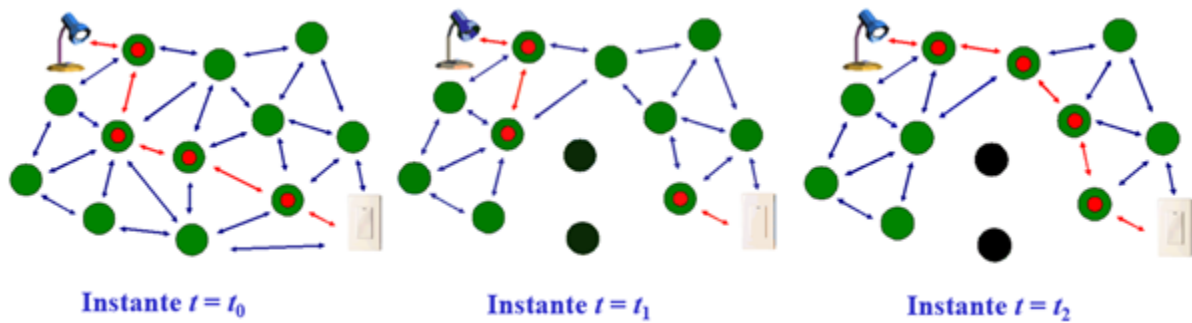


Figura 2.15. Ejemplo de encaminamiento Zigbee tras caer un nodo [3]

Arquitectura Zigbee

El estándar Zigbee se basa en una estructura de capas en forma de pila como en el modelo ISO (International Organization for Standardization) para la interconexión de sistemas abiertos (OSI). Cada una presenta sus funciones y se relacionan con las vecinas superior e inferior. La capa superior utiliza el servicio de puntos de acceso (SAP) para hacer una petición (*request*) de un servicio. La capa inferior, en cambio, confirma (*response*) a la capa superior que el servicio se ha efectuado exitosamente.

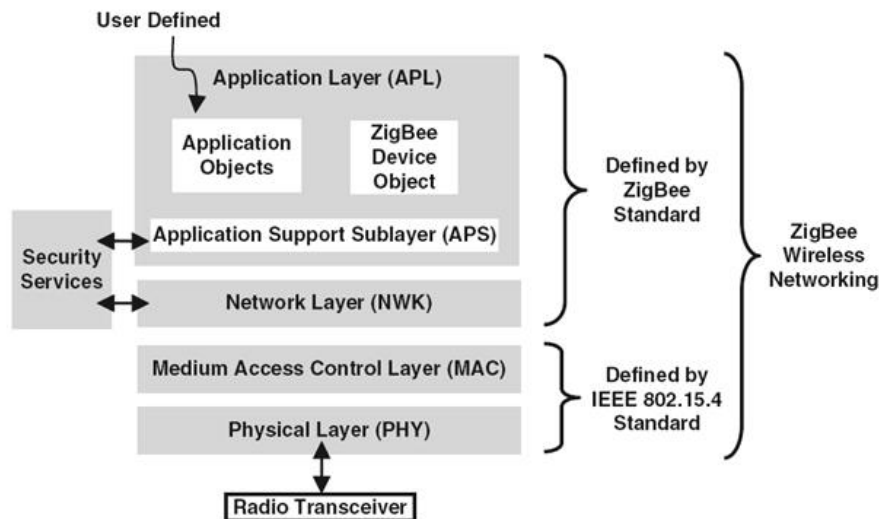


Figura 2.16. Arquitectura Zigbee [3]

Las capas de abajo a arriba son: física, enlace, red, soporte a la aplicación y aplicación. Las capas, física (PHY) y de enlace (MAC) se basan en el estándar IEEE 802.15.4 y son independientes del estándar Zigbee, son las que están en contacto con el *hardware*.

Las bandas de frecuencia ISM están divididas en 27 canales y se utiliza *Frequency Hopping*⁸ para evitar interferencias. Estos canales se dividen en dos bandas:

⁸ Técnica que se utiliza para alternar la frecuencia de emisión entre canales.

- *Banda alta*: portadora a 2.4 GHz. No hay licencia, velocidades superiores (256 kbps) y más canales (16), baja potencia, ciclos de trabajo pequeños.
- *Banda baja*: 868 MHz en Europa y 915 MHz en EUA. Ofrecen menos usuarios e interferencias, permiten una mejor penetración de la señal. Ofrece más cobertura que en la banda alta.

La **capa física** se encarga de activar/desactivar el transmisor receptor de radio, de transmitir/recibir los datos, elegir el canal de operación haciendo una estimación de la energía del canal determinando si está o no libre y, finalmente, generar el LQI (Link Quality Indicator), que a partir del nivel de la señal puede estimar la calidad de los datos recibidos.

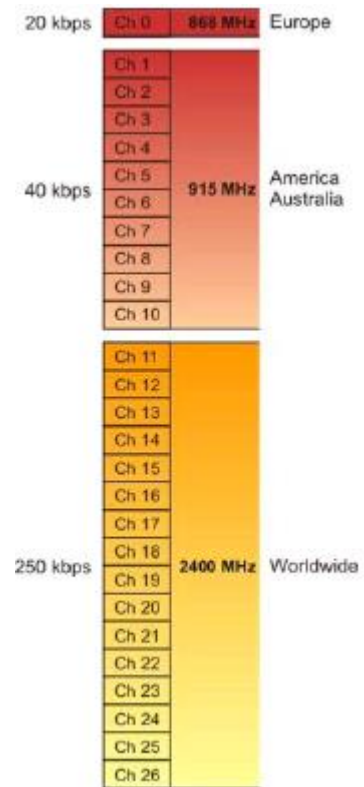


Figura 2.17. Bandas de frecuencia Zigbee [3]

Los paquetes en la capa física se estructuran de la siguiente manera:

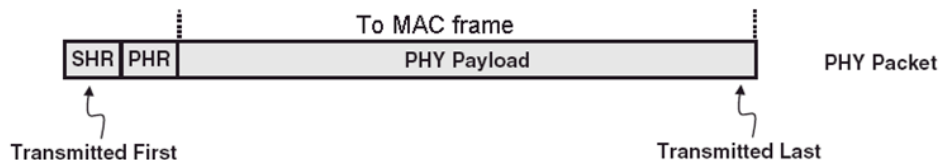


Figura 2.18. Frame en la capa física [3]

SHR: (*Synchronization Header*): se trata de una cabecera de sincronización.

PHR: (*PHY Header*): contiene información sobre la longitud de la trama.

PHY payload: Contiene los datos y pedidos por el destinatario. Es el *frame* que irá hacia la capa superior de enlace (MAC).

La **capa de enlace** es la capa encargada de acceder al medio y de empaquetar los datos que después se encaminarán a la capa superior (capa de red). El acceso al canal se lleva a cabo mediante CSMA-CA (Carrier Sin Multiple Access Collision Avoidance). El nodo escucha el canal enviando una línea con que lo quiere ocupar, recibe una aceptación (ACK) del receptor conforme puede continuar la transmisión. Con este mecanismo se evitan colisiones, pero el rendimiento es algo ínfimo. Los dispositivos acceden a un canal comprobando si está libre y evitando la interrupción de una comunicación ya iniciada.

Capa de aplicación y seguridad

Como hemos podido observar en la *Figura 2.16*, la arquitectura ZigBee consta de varios componentes en las capas de Control de Acceso al Medio (MAC), la física (PHY) y la de red Zigbee (NWK).

La **capa de aplicación** de ZigBee se subdivide en la subcapa APS, la capa ZDO (Zigbee Device Objects) y los objetos de aplicación vienen definidos por cada uno de los fabricantes, por tanto, cada dispositivo consta de su propia APL (Application Layer) y se gestiona según las funcionalidades que determinen los fabricantes.

El protocolo Zigbee ofrece cuatro servicios relativos a la seguridad de la información. Este control y seguridad se encuentra fundamentalmente en la capa de red y cumple con los siguientes requisitos:

- Control de accesos, el nodo *router*/coordinador tiene una lista de dispositivos verificados que deben acceder a la red.
- Utiliza mecanismos de filtrado, los datos están encriptados con un código de 128 bits.
- Protección de las tramas ante la posible modificación por parte de terceros.
- Secuencias de refresco para comprobar que las tramas no han sido modificadas o dañadas.

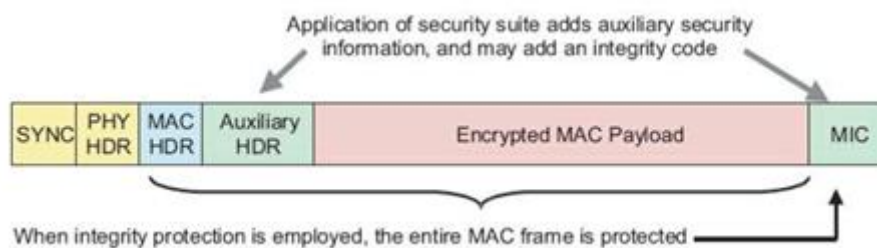


Figura 2.19. Protocolos de seguridad en una frame Zigbee [3]

Formación y unión a una red Zigbee

Resumidamente y para concluir el proceso de la creación de una red Zigbee, los pasos para su formación son los siguientes:

- *Buscar los canales radio*, el coordinador buscará el que tiene menos actividad. Intentará evitar los canales que están usando una red Wi-Fi.
- *Asignar el identificador de red (PAN-ID)*, el coordinador asigna el PAN-ID a la red, evitando asignar el mismo ID de otras redes que operen en el mismo canal.
- *Iniciar la red*, el coordinador se autoconfigura y empieza a descubrir nuevos nodos.

Una vez tenemos creada la red, hay que unirse a ella, los pasos para cumplir este proceso final son los siguientes:

- *Buscar la red*, el nuevo nodo escanea el canal de radio, encuentra las redes operativas y se une a aquella a la que identifica.
- *Selección del nodo padre*, el nuevo nodo es capaz de elegir el *router*/coordinador al que se debe unir. Suele ser el que tiene la señal más fuerte.
- Envío de la solicitud para *unirse al nodo padre* mediante un mensaje.
- *Aceptación o rechazo de la solicitud de unión*, los *routers*/coordinador deciden si el nuevo nodo formará parte de la red asignándole una dirección.

Ventajas y desventajas de utilizar Zigbee

Después de haber expuesto las características detalladas del protocolo Zigbee, desglosaremos la información de una manera menos técnica para concluir con los aspectos relevantes que nos interesan en el uso de este estándar.

Las **ventajas** que nos han hecho fijarnos para la realización del estudio, han sido las siguientes:

- Tiene un bajo consumo energético, ya que los nodos si no trabajan suelen estar en *standby*. No suelen necesitar alimentación externa por el uso del *energy harvesting* en sus baterías al recargarse. Las baterías pueden durar años.
- Utiliza técnicas de ensanchamiento del espectro (DSSS) para mejorar el comportamiento de la señal ante el efecto multicamino.
- Se pueden crear redes de hasta 65535 nodos, sencillas de manejar, escalables y flexibles.
- Es óptimo para redes de baja tasa de transferencia de datos e ideal para conexiones punto a punto y punto a multipunto.
- Diseñado para el direccionamiento de información y el refrescamiento de la red.
- Opera en las bandas libres de ISM para conexiones inalámbricas.
- Reduce tiempos de espera en el envío y recepción de paquetes.
- Válido para múltiples topologías de red: estática, dinámica, estrella, malla, árbol.
- Provee conexiones seguras entre dispositivos. 128-bit AES de cifrado. [17]
- Son más baratos y de construcción más sencilla, al ser inalámbricos permiten su utilización en estructuras ya edificadas sin grandes remodelaciones.

Por consiguiente, las **desventajas** que presenta Zigbee son:

- Presenta una cobertura relativamente baja (de 10 a 100 metros) en comparación a otros estándares. Aumenta el alcance en espacios abiertos o si se usan nodos *router* como intermediarios.
- La tasa de transferencia es muy baja, solo manipula textos pequeños comparados con otras tecnologías.
- Tiene una menor cobertura que otros estándares porque pertenece a redes inalámbricas de tipo WPAN.

- Al trabajar en la banda de 2,4 GHz tiene que lidiar con una gran cantidad de interferencias de otros protocolos que comparten la misma frecuencia (Wi-Fi, Bluetooth, telefonía, etc.).
- Se debe revisar bien los parámetros de los productos que se quieran conectar mediante Zigbee. Al no definir los parámetros permitidos en la capa de aplicaciones puede generar la aparición de dispositivos no compatibles entre sí, aunque cumplan fielmente los estándares en la capa de red (ZigBee) y la capa física (IEEE 802.15.4).

2.2.4. LoRaWAN (LoRa)

LoRaWAN (Long Range Wide Area Network) es una de las mejores tecnologías inalámbricas para IoT. Su popularidad se debe al robusto esquema de modulación basado en la modulación Chirp Spread Spectrum⁹ (CSS). [19] Es un estándar adaptado para redes de baja potencia y de área amplia (LPWAN), diseñado específicamente para dispositivos de bajo consumo de alimentación, que operan en redes de alcance local, regional, nacional o global. *The Things Network*, trabaja con LoRaWAN y está montando una gran red de dispositivos IoT por todo el mundo. [20]

Los receptores LoRa presentan una sensibilidad muy elevada que, dependiendo del factor de dispersión utilizado, puede ser tan bajo como -148 dBm¹⁰. Como resultado de esta sensibilidad, se pueden diseñar enlaces de radio de varios kilómetros de longitud. LoRa también ofrece ventajas tanto en el bloqueo como en la selectividad y puede lograr una alta inmunidad a las interferencias con un bajo consumo de energía y osciladores de bajo coste.

LoRa se basa en una modulación de baja potencia diseñada para enviar pequeños paquetes de datos a largas distancias, operando con una batería. Puede demodular señales que están entre 7,5 dB y 20 dB por debajo del fondo de ruido. [19]

En la actualidad, la tecnología LoRa está administrada por la *LoRa Alliance*, quien certifica a todo fabricante de hardware que desee trabajar con esta tecnología. Al conseguir esto hace que sus dispositivos estén estandarizados y puedan ser compatibles entre sí.

Como podemos ver las redes LoRaWAN presentan características muy interesantes que encajan con nuestro estudio a pesar de ser una red LPWAN. Es una tecnología WSN emergente que hay que tener en cuenta. Es ideal para conexiones a grandes distancias, para redes de IoT en las que se necesitan sensores que no dispongan de corriente eléctrica de red y para construir redes privadas de sensores y/o actuadores. Además, incluye una serie de propiedades adicionales a tener en cuenta:

⁹ Es un sistema de espectro ensanchado en el que la portadora de radiofrecuencia se modula con un periodo fijo y una secuencia de ciclo de trabajo fija.

¹⁰ Dato extraído del dispositivo *Semtech SX1276* LoRa. [18]

- Alta tolerancia a las interferencias.
- Bajo Consumo (hasta 10 años con una batería).
- Largo alcance, de 10 a 20 km.
- Baja transferencia de datos (hasta 255 bytes).
- Conexiones bidireccionales seguras mediante encriptación de extremo a extremo.
- Frecuencias de trabajo ISM: 868 MHz en Europa, 915 MHz en América y 433 MHz en Asia. Aunque la tecnología puede operar en cualquier frecuencia por debajo del 1 GHz, cualquier persona o empresa puede hacer uso de ella sin necesidad de licencia.
- Modulación FSK (Frequency Shift Keying). [21] Efectiva para el bajo consumo de energía.
- Es una tecnología que ofrece un muy bajo coste en componentes y programación.
- Conexión de infinidad de sensores y equipos a redes públicas o privadas (hasta 1 millón de nodos en red).

Para entender su funcionamiento debemos conocer los parámetros de comunicación LoRa a la hora de crear la red. Por ello hay que tener en cuenta:

- *Canal dentro de la banda*: frecuencia central que representa el canal.
- *Spreading factor (SF)*: define el número de bits usados para codificar un símbolo. Cuanto mayor sea SF, menor será la velocidad de transferencia, aunque la inmunidad a interferencias será mayor.
- *Coding rate (CR)*: es la forma de codificar para poder tener una mejor corrección de errores. Según la técnica especificada, añade símbolos de control para saber si los datos son correctos o no.
- *Bandwidth (BW)*: indica el ancho de banda frecuencial que vamos a usar. [20]

Arquitectura de red

Este protocolo está basado en una topología en forma de estrella de largo alcance, ya que con ella se preserva la vida útil de las baterías y los nodos trabajan punto a punto sin existir intermediarios.

El protocolo LoRaWAN se compone básicamente de dos dispositivos, *gateways* y *end devices*:

- *Gateways (antenas)*: son los encargados de recibir y enviar la información a los nodos. Pueden escuchar hasta 8 *end devices* a la vez.
- *End devices (terminadores)*: son los dispositivos finales que envían y reciben información hacia el *gateway*. Sirven para diferentes aplicaciones y tienen diferentes requisitos. Para optimizar una variedad de perfiles de aplicaciones finales, LoRaWAN utiliza diferentes clases de dispositivos (A, B o C) que compensan la latencia de la comunicación del enlace descendente de la red con la vida útil de la batería.

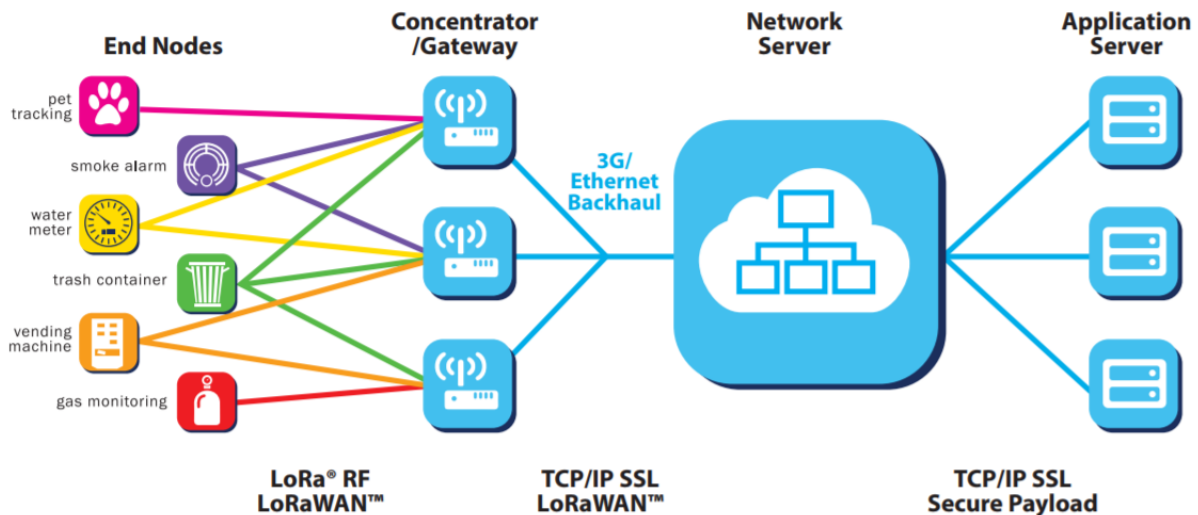


Figura 2.20. Arquitectura de una red LoRaWAN [22]

Como se puede observar en la *Figura 2.20* que muestra la arquitectura de una red LoRaWAN, los nodos de red no están asociados con una puerta de enlace específica. En cambio, los datos transmitidos por un nodo generalmente son recibidos por múltiples *gateways*. Cada puerta de enlace reenviará el paquete recibido desde el nodo final al servidor de red basado en la nube a través de algún *backhaul* [23] (ya sea celular, Ethernet, satelital o LTE). La inteligencia y la complejidad del servidor de red, que administra la red y filtra los paquetes recibidos redundantes, realizará verificaciones de seguridad, programará reconocimientos a través de la puerta de enlace óptima y realizará una velocidad de datos adaptada.

Si un nodo es móvil o se está moviendo, no se necesita un traspaso de *gateway* a *gateway*. Los nodos son asincrónicos y se comunican cuando tienen datos listos para enviar, ya sea por eventos del sistema o porque están programados para trabajar en ciertos *slots* de tiempo. Este tipo de protocolo generalmente se conoce como el método Aloha. [3]

Se puede implementar una red con una cantidad mínima de infraestructura y, a medida que se necesita capacidad, se pueden agregar más *gateways*, lo que aumenta las velocidades de datos, reduce la cantidad de escuchas a otras puertas de enlace y aumenta la capacidad en un factor del orden de 6 o 8 veces más.

LoRaWAN define el protocolo de comunicación y la arquitectura del sistema para la red (MAC), y se encarga de unir diferentes dispositivos LoRa gestionando sus canales y parámetros de conexión: canal, ancho de banda, cifrado de datos, etc. Por otra parte, que la capa física (PHY) LoRa habilita el enlace de comunicación de largo alcance punto a punto mediante FSK. [3]

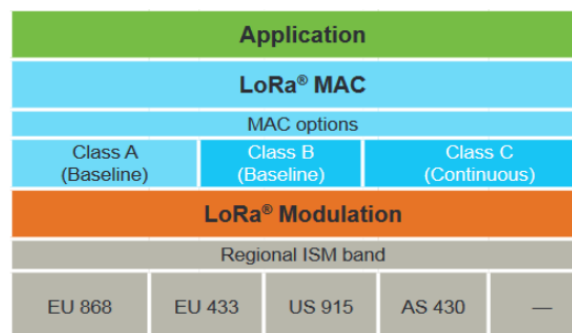


Figura 2.21. Arquitectura de una frame LoRa [22]

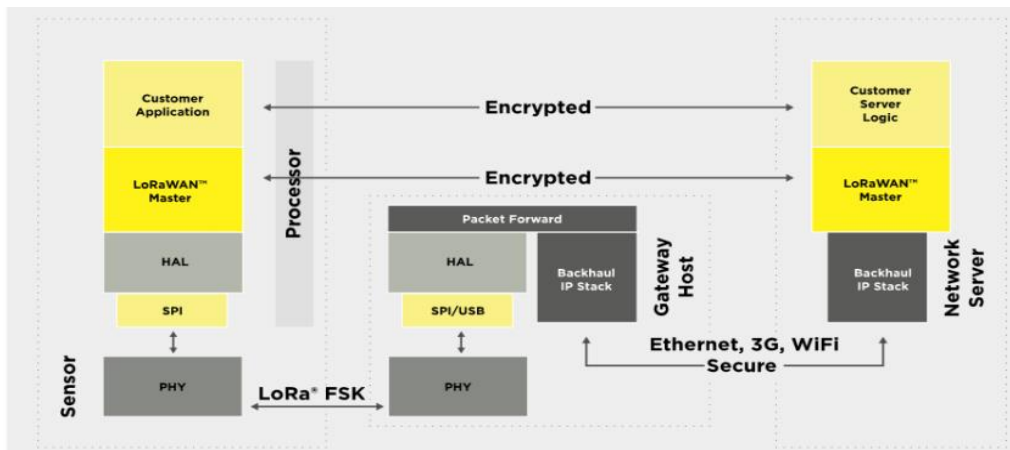


Figura 2.22. Arquitectura LoRaWAN [24]

Seguridad

LoRaWAN proporciona varias capas de cifrado, que hacen uso del algoritmo AES combinando diferentes modos de comunicación: CMAC, para proteger la integridad y CTR, para la encriptación. Cada dispositivo tiene su propia clave AES-128 (AppKey) y un identificador único global que se utilizan en el momento de autenticación del dispositivo. Las claves de datos protegen las comunicaciones a través de estas tres capas de seguridad:

- *Network Session Key*: Clave de 128 bits que garantiza seguridad con respecto al nivel de red.
- *Application Session Key*: Clave de 128 bits que garantiza seguridad y cifrado de los datos de extremo-extremo, desde el dispositivo hasta el servidor.
- *Application Key*: clave de 128 bits que se utiliza para despliegues OTAA¹¹. [25]

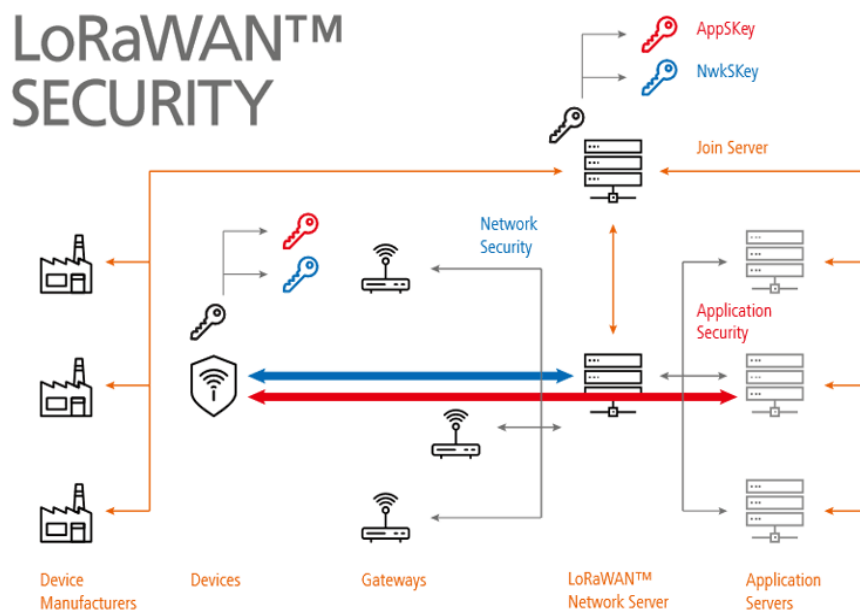


Figura 2.23. Seguridad en LoRaWAN [25]

¹¹ Es la forma preferida y más segura de conectarse con una red IoT. Los dispositivos realizan un procedimiento de unión con la red, durante el cual se asigna un *DevAddr* dinámico y se negocian las claves de seguridad con el dispositivo.

Ventajas y desventajas de utilizar LoRaWAN

Una vez tenemos expuestas las características que nos han hecho proponer una red LPWAN como propuesta a nuestro estudio, es el momento de desglosar la información exponiendo sus ventajas y desventajas.

Las principales **ventajas** que nos ofrece LoRaWAN son:

- Seguridad. Codificación de extremo a extremo, la información viaja encriptada, minimizando el riesgo de robo de datos.
- Bajo coste. Se trata de una tecnología relativamente barata en comparación a los otros protocolos. Puede configurar y gestionar su propia red.
- Amplio margen de cobertura (hasta 20 km). Tipología estrella
- Alta tolerancia a las interferencias.
- Bajo consumo de energía. Bajo las condiciones adecuadas, las baterías pueden durar años.
- Bajo nivel de complejidad a nivel técnico. Los nodos de LoRa están formados por dispositivos transceptores que cuentan con interfaces de comunicación que les permiten ser utilizados junto con un microcontrolador (Arduino). Con LoRa simplemente se le solicita los datos al microcontrolador y se envían a través del canal al *gateway* que se encargará de subir los datos a Internet.
- Cobertura global. Se pueden enviar datos a través de LoRaWAN siempre y cuando estén cerca de uno de sus *gateways* distribuidos alrededor.
- Interoperabilidad de las diversas redes LoRaWAN en todo el mundo.
- Permite la interconexión entre objetos inteligentes sin la necesidad de instalaciones locales complejas. Además, otorga amplia libertad de uso al usuario final, al desarrollador y a las empresas que quieran instalar su propia red para IoT porque opera en las bandas libres de ISM.

Por otro lado, como **desventajas** se destaca:

- Se utiliza para enviar paquetes pequeños de datos cada cierto tiempo, no a tiempo real.
- No permite el envío de información multimedia ni llamadas telefónicas.
- No se aconseja el uso bidireccional instantánea entre nodo y *gateway* por su alta latencia de red. [26]
- Imposibilidad de transmitir y recibir datos de manera simultánea
- Necesidad obligatoria de sus *gateways* en el país que se quiera trabajar con este protocolo.

2.3. Comparación y elección del estándar

El propósito del estudio una vez hemos investigado los estándares anteriores, es proponer una solución con uno, o dos, de ellos haciendo una comparación según las explicaciones realizadas

anteriormente. Detallar el motivo que nos ha llevado a esa elección y, principalmente, identificar cuál sería la mejor opción para conseguir el propósito de este proyecto, dar cobertura inalámbrica a una red de sensores en un espacio subterráneo.

Para conseguir esto, la mejor manera de exponer la información es mediante una tabla, donde destacaremos cada estándar relacionando los requisitos que se deben destacar para la realización del prototipo. Haciendo esta simplificación esquemática con los valores más relevantes de cada protocolo, creemos que es la mejor manera de resumir los apartados anteriores en los que ya hemos detallado las especificaciones de cada estándar y sacar conclusiones.





	 ZigBee®	 Wi Fi™	 LoRaWAN™	 Bluetooth™
Alcance típico	10 - 100 m	70 - 250 m	10 - 20 km	10 - 30 m
Velocidad de datos	20 - 250 kbps	6,9 Gbps	250 bps - 50 kbps	3 Mbps
Cantidad de datos por transmisión	Baja	Alta	Muy baja	Media
Bandas de trabajo	868 MHz - 2,4 GHz	5 GHz	868 - 433 MHz	2,4 GHz
Seguridad	Alta	Muy baja	Muy alta	Muy baja
Latencia	Baja	Alta	Muy alta	Alta
Núm. nodos por red	265 - 65535	32	62500	8
Tolerancia a interferencias	Media	Baja	Alta	Baja
Duración baterías	Meses - Años	Horas	Años	Días
Consumo	3 μ A - 30 mA	20 - 400 mA	20 nA - 10 mA	0,2 - 40 mA
Precio	Bajo	Medio - Alto	Muy Bajo	Medio
Configuración	Simple	Compleja	Media	Compleja
Trabajo a tiempo real	Preciso	Muy preciso	Poco preciso	Muy preciso
Adaptado a smartphones	No	Sí	No	Sí

Tabla 2.3. Comparación de los estándares de comunicación WSN

Como hemos podido ver en la *Tabla 2.3*, el estándar que nos interesa para desarrollar nuestro proyecto es un híbrido entre **LoRaWAN** y **Wi-Fi 5**. Bluetooth quedaría totalmente descartado por adaptarse a los requisitos que debe cumplir para ser óptima a la hora de desarrollar una red WSN en un parking subterráneo. Apenas cumple dos o tres características que nosotros buscamos para nuestra red de sensores.

Zigbee, en cambio, podría haber sido una opción bastante interesante para sustituir a LoRaWAN. Cumple con creces con los requisitos para ser una tecnología adecuada para el proyecto, ya que permite un envío de tamaño de paquetes más elevado entre transmisiones que LoRa.

Los principales motivos que nos han llevado a descartar Zigbee como solución han sido su bajo alcance (10 - 100 m) entre sensores, *routers* y coordinador. Esto nos causaría problemas para cubrir la red en un área extensa, deberíamos utilizar una cantidad elevada de nodos y eso supondría un coste mucho más elevado e ineficiente para el proyecto. Además, la banda de trabajo (2,4 GHz) causaría mucha interferencia en el entorno de cobertura, si se diera el caso de que hubiese más dispositivos conectados alrededor con otras tecnologías WSN. Zigbee (WPAN) es un protocolo orientado para aplicaciones como la domótica.

Sin duda alguna, el estándar por excelencia para el desarrollo de nuestro proyecto es LoRaWAN. Su largo alcance (10 - 20 km) permite cubrir un área como la de un parking (LPWAN) con la necesidad de un solo *gateway*, dos si fuera un parking muy extenso. También, su banda de trabajo en Europa (868 y 433 MHz) permite evitar interferencias con otras tecnologías WSN e incluso, atravesar edificios u objetos robustos con sus ondas de RF. Además, las bandas de trabajo (ISM) son bandas abiertas en las que cualquier compañía puede trabajar y desarrollar sin dar parte a ninguna empresa propietaria de la banda de radiofrecuencia.

Los dos problemas que nos podrían llevar a replantearnos el uso de LoRa son su elevada latencia, la comunicación bidireccional entre dispositivos y *gateway* podría generar *delay* de 1-2 segundos en función del tiempo real a la hora de transmitir, para nuestro caso no influye este aspecto, pero si se quisiera precisar más en otro sí que sería algo deficiente. El segundo problema es el tamaño de los paquetes que LoRa puede enviar/recibir. Al tratarse de una tecnología de tan largo alcance no permite que los paquetes tengan mucha extensión ni que se transmitan a grandes velocidades. En nuestro proyecto no siempre enviaremos grandes paquetes de datos, pero puede ser que haya casos en los que sí, entonces en este aspecto LoRa no cumpliría con los requerimientos de nuestro estudio y deberemos de programar esa transmisión elevada de información sin perder datos relevantes.

Por ese motivo, lo ideal para este proyecto es utilizar un híbrido entre el estándar Wi-Fi 5 y LoRaWAN. Las carencias en latencia y cantidad de transmisión de datos con el protocolo de Wi-Fi quedarían totalmente cubiertas, incluso nos ofrecería una velocidad más alta de transmisión de datos. El problema de Wi-Fi 5 sería su baja seguridad, a pesar de que con Wi-Fi 6 se va a mejorar mucho en ese aspecto con el WEP3 en un futuro muy próximo.

Actualmente, al ser un estándar muy globalizado y de uso masivo presenta muchos problemas a la hora de hablar de seguridad.

Análogamente, con Wi-Fi 5 tenemos también el inconveniente de su alcance (70 - 250 m), si dotáramos todos los nodos con este estándar el coste de la infraestructura inalámbrica sería descabellado como nos pasaría con Zigbee. Al trabajar en la banda de 5 GHz tendríamos menos interferencia con otros estándares, pero no lograríamos cubrir mucha zona de cobertura, que al final es lo que buscamos en nuestro estudio. Además, con Wi-Fi solo es posible conectar un máximo de 32 nodos si queremos que el sistema funcione de manera óptima. Este protocolo estaría más orientado a parkings subterráneos de pocas plazas y plantas.

La seguridad, dependiendo del uso que vayamos a darle a los dispositivos debemos tenerla como prioritaria o no. En nuestro caso, al conectar una red de sensores de coches eléctricos no es un aspecto primordial para el proyecto, y con las prestaciones de seguridad que ya adapta el Wi-Fi 5 podríamos satisfacer nuestras necesidades sin problema alguno. LoRa, por el contrario, presenta mucha más seguridad que el Wi-Fi, los datos se encriptan de extremo a extremo, y todos los dispositivos de la red están perfectamente identificados con una clave a la hora de comunicarse. En seguridad LoRa es más efectivo, aunque Wi-Fi ya cumple con los requerimientos que este proyecto solicita.

Una vez conocidas las características de cada estándar, para el desarrollo del estudio procederemos a la implementación de una solución con los dos estándares justificados anteriormente, LoRaWAN y Wi-Fi. Buscaremos la realización de uno o varios prototipos que resuelvan los problemas de dar conexión en zonas de difícil cobertura y desde el marco teórico buscaremos el diseño más estandarizado para poder implementarlo en el máximo número de parkings subterráneos posibles.

2.4. Entornos de desarrollo

2.4.1. IDE Arduino

Se recurre al entorno Arduino como sistema de monitorización y automatización de los módulos LoRa para el diseño de nuestro prototipo a baja escala. Al realizar un prototipo escalable a la propuesta comercial, pero realizado dentro del ámbito de un proyecto académico, parece lógico recurrir a tecnologías libres para el diseño de nuestro sistema WSN. Por ello, se presenta a modo introductorio algunos conceptos de la IDE de Arduino, donde basamos la programación de nuestros módulos LoRa que vienen adaptados al lenguaje de este entorno.

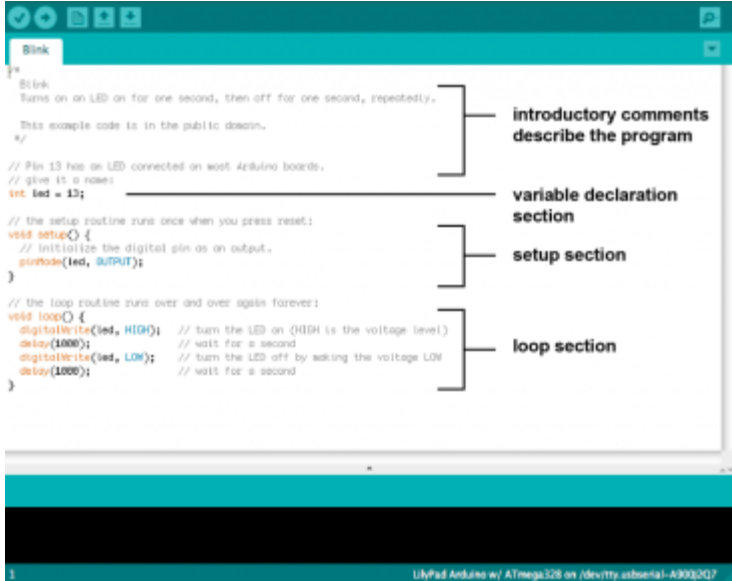
Arduino es una compañía de *software* y *hardware* abierto que viene definido como un proyecto internacional enfocado al diseño y manufacturación de placas de desarrollo *hardware*, para construir dispositivos digitales e interactivos que puedan dar una funcionalidad específica en el marco tecnológico. El módulo Arduino suele ser empleado como el MCU de cualquier sistema formado por sensores y actuadores, de tal forma que, una vez definido un conjunto de

estos dispositivos, se puede programar en la placa el código que permita automatizar cualquier proceso. Gracias a su gran versatilidad y flexibilidad, se pueden aportar soluciones tecnológicas en cualquier ámbito, como puede ser: domótica, industria, robótica, agricultura, IoT, comunicaciones.

El entorno de programación de Arduino, del cual se muestra una imagen de su interfaz en la *Figura 2.24*, emplea un lenguaje similar al lenguaje de programación C + +, donde se introducen las líneas de comandos que componen lo que se denomina *sketch*. Existen dos bloques principales en su interfaz:

- *void setup*: en este bloque se realiza la configuración del dispositivo y la asignación de pines (entradas y salidas analógicas y digitales).
- *void loop*: se corresponde con el bucle que contiene el programa principal, donde se incluyen todas las sentencias necesarias para obtener la funcionalidad deseada.
- *comentarios*: es una construcción del lenguaje de programación destinada a insertar anotaciones legibles al programador en el código fuente del *sketch*.
- *declaración de variables*: espacio donde asignamos un valor fijo, en memoria, a elementos específicos durante el desarrollo del programa.

Una vez finalizado el programa, se compila y se puede cargar en el módulo Arduino mediante una conexión por cable USB.



```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
25
```

reducir el número de líneas de código, ya que simplemente se requiere importar una librería y definir las variables de este tipo, similar a la programación orientada a objetos. Este aliciente facilita en gran medida las tareas de programación. [28]

Comunidad Arduino

Un factor del éxito de Arduino ha sido la comunidad que está apoyando este proyecto y que día a día publica nuevo contenido, divulga y responde a las dudas. En Internet hay disponible todo tipo de cursos, tutoriales, herramientas de consulta y proyectos que ayudan a que se pueda usar Arduino con facilidad. Además, en [29] se ve que Arduino playground es un wiki donde todos los usuarios de Arduino pueden contribuir, siendo el lugar donde publicar y compartir código, diagramas de circuitos, tutoriales, actualizaciones, cursos y, sobre todo, el lugar donde preguntar cuando existen dudas sobre un problema relacionado con su entorno. Haciéndolo así, la base de datos de conocimiento por excelencia de Arduino.

2.4.2. *Hive MQ*

Hive MQ es un MQTT bróker y una plataforma de mensajería basada en cliente/suscriptor diseñada para el movimiento rápido, eficiente y fiable de datos hacia y desde dispositivos IoT conectados. Utiliza el protocolo MQTT para el envío instantáneo y bidireccional de datos entre su dispositivo y los sistemas de base de datos de la empresa.

Protocolo MQTT

MQTT fue creado por el Dr. Andy Stanford-Clark de (*IBM*) y Arlen Nipper (de *Arcom*, ahora *Eurotech*) en 1999. Cuando se les planteó el reto de encontrar la manera de enviar los datos de los sensores de los oleoductos en el desierto a sistemas SCADA externos (control de supervisión y adquisición de datos), decidieron utilizar una topología de publicación/suscripción basada en TCP/IP o cifrada mediante TLS, que se basaría en los eventos para mantener bajos los costos de transmisión de los enlaces satelitales.

Aunque MQTT todavía está estrechamente asociado con *IBM*, ahora es un protocolo abierto que es supervisado por la Organización para el Avance de los Estándares de Información Estructurada (OASIS). Está enfocado al envío de datos en aplicaciones donde se requiere muy poco ancho de banda. Además, sus características le permiten presumir de tener un consumo realmente bajo, así como precisar de muy pocos recursos para su funcionamiento.

Estas características han hecho que rápidamente se convierta en un protocolo muy empleado en la comunicación de sensores y, consecuentemente, dentro del Internet de las Cosas. [30]

Uso de Hive MQ

Hive MQ está diseñado para abordar algunos de los desafíos técnicos clave a las que se enfrentan las organizaciones al crear nuevas aplicaciones de IoT que incluyen, por ejemplo:

- Creación de aplicaciones confiables y escalables de IoT para negocios críticos.
- Entrega rápida de datos para satisfacer las expectativas de los usuarios finales de productos de IoT receptivos.
- Menor coste de operación mediante el uso eficiente de *hardware*, red y recursos en la nube.
- Integración de datos de IoT en sistemas empresariales existentes.

En la *Figura 2.25* ejemplificamos el papel comunicativo que tiene el broker Hive MQ dentro de cualquier aplicación de IoT.

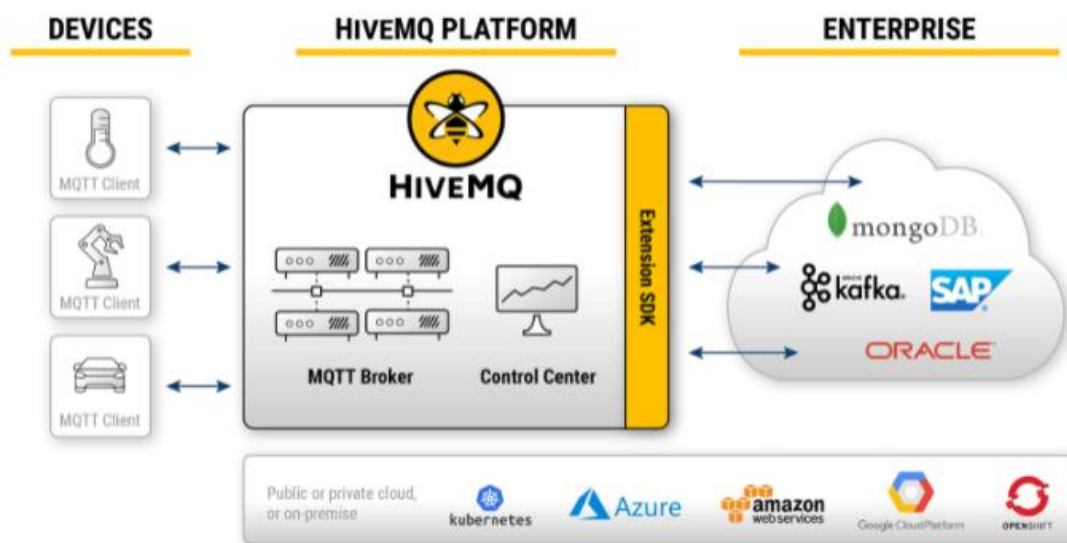


Figura 2.25. Funcionamiento de Hive MQ [30]

Nos hemos fijado en Hive MQ porque es un bróker abierto compatible con infinidad de aplicaciones y protocolos orientados a soluciones IoT. Al ser Hive MQ compatible con Arduino y su IDE se podrá acceder a una base de datos desde el *sketch* del *gateway*, que es uno de los objetivos para desarrollar el prototipo demostrador. Por lo tanto, el uso de este bróker como base de datos en la nube constituye una opción adecuada para poder realizar pruebas con el prototipo final que explicaremos más adelante.

3. Prototipo con LoRaWAN

3.1. Reafirmación al uso de esta tecnología

El protocolo con el que vamos a realizar nuestro prototipo se basará fundamentalmente en LoRaWAN. Las características detalladas que nos han hecho implementar la solución con este estándar están expuestas en el apartado anterior, pero querría destacar cuatro características importantes que han sido claves para la elección de LoRa:

- Su gran alcance en las comunicaciones.
- La alta sensibilidad de los receptores.
- La gran capacidad a la hora de combatir las interferencias.
- Su bajo coste.

El prototipo que vamos a realizar consistirá en desplegar en un parking una red de sensores LoRa. Cada plaza de carga para coche eléctrico tendrá su dispositivo LoRa que recibirá la información de los cargadores y la comunicará a un *gateway* LoRa. El *gateway* o pasarela se conectará a una red externa vía Wi-Fi para tener salida a Internet, enviando la información que recibida de los cargadores a la base de datos del sistema. La comunicación es bidireccional, esto quiere decir que, si el cargador de coche necesitara recibir información de la nube, la comunicación entre dispositivos sería a la inversa. A continuación, en la ilustración ejemplificamos nuestro prototipo.

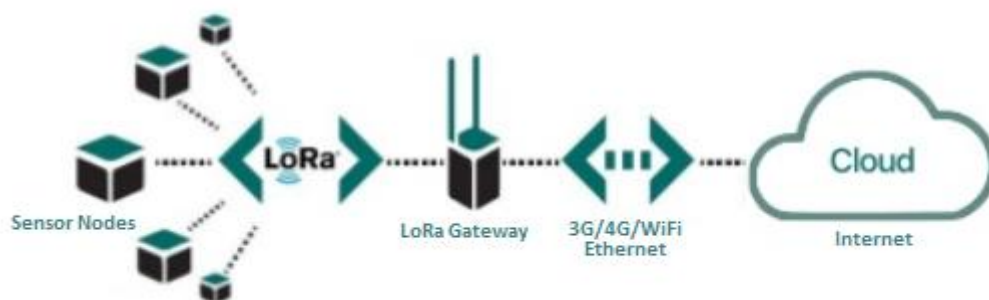


Figura 3.1. Ejemplo básico de una red LoRa [32]

Por tanto, en este prototipo vamos a combinar las funcionalidades de LoRa y Wi-Fi. LoRa será utilizado para enviar la información de los sensores de los cargadores, dentro de un área de alcance típica, hacia un *gateway* y este último se encargará de enviar esta información a una base de datos integrada en la nube, realizando el cambio de protocolo pertinente, en este caso a Wi-Fi.

3.2. Planificación del proyecto

Una vez estudiados los estándares de comunicación WSN y meditada la idea de cómo dar conectividad a una red de sensores de cargadores para coches eléctricos en un parking subterráneo, es momento de proyectar dicha idea y convertirla en realidad.

Para conseguir darle forma al prototipo tenemos que realizar una planificación rigurosa y organizada de cómo queremos distribuir las tareas. Esta planificación nos llevará al desarrollo final del proyecto paso a paso. Para ello, debemos separar el proyecto en sus fases principales, englobando una serie de trabajos, para después, ordenarlos y conseguir de esta forma planificar el prototipo entero de manera simplificada, ordenada y aproximando unos tiempos para la realización del proyecto completo.

Introducidos en este *mindset*¹², es momento de contrastarlo con nuestro trabajo. Por lo tanto, nuestro prototipo vendrá dividido por **dos fases** primordiales:

- *Fase de diseño, programación y asignación de recursos.* Es la fase más extensa del prototipo. En esta fase deberemos conocer el entorno donde vamos a realizar nuestra instalación, es decir, estudiar los planos del parking, encontrar los materiales y recursos idóneos para ese lugar (sensores, *gateways*, trabajadores, etc.), saber que necesitaremos para realizar la red WSN y programar los chips de cada dispositivo para que la información pueda ser enviada/recibida desde los sensores hasta la base de datos del sistema.
- *Fase de instalación y desarrollo.* Es la fase final del prototipo, sucesora a la anterior e importante para la ejecución del objetivo final. Deberemos instalar la red WSN basándonos en el diseño y programación de la anterior fase, además de comprobar que el desarrollo se ha realizado con éxito. También, esta fase contiene una fase de corrección y mejora, en caso de haber un error en el diseño es aquí donde lo corregiremos y comprobaremos mediante testeos que la instalación funciona como es debido.

Una vez introducidas las fases, es momento de descomponerlas en sus respectivas tareas. Como hemos mencionado anteriormente, esto nos ayudará a simplificar el trabajo y encontrarnos más cerca de nuestro objetivo paso a paso y respetando un orden ejecución. El desarrollo de la **primera fase** vendrá estructurado por los siguientes trabajos:

- *Estudio del entorno.* Nuestro caso es un parking subterráneo. Lo primero que tenemos que hacer es conocer el entorno donde vamos a desplegar nuestra red WSN. Mediante la visualización de planos debemos reconocer los puntos estratégicos donde instalar los dispositivos y tener una idea inicial de cómo distribuirlos por el espacio. Siempre hay

¹² Del inglés “mentalidad”, hace alusión a la mentalidad de lo posible, la mentalidad de creer que se puede realizar.

que buscar la instalación mínima de dispositivos para dar cobertura a toda la red, y así ahorrar en costes.

- *Búsqueda de materiales y recursos.* Conocido el entorno y los espacios donde se nos permitirá instalar los dispositivos es momento de buscar y estudiar el material específico para poder desplegar nuestra red en el parking. Debemos conocer la cantidad de plazas con cargadores para saber la demanda de *end devices* y posteriormente, según la demanda anterior buscar los *gateways* mínimos que consigan cubrir completamente el área de conexión.
- *Diseño de la red WSN.* Una vez estudiado el lugar de la instalación de red y hallados los materiales y recursos para la instalación, es momento de realizar el/los diseño/s sobre los planos. Se deberá especificar sobre el plano el lugar de instalación de cada dispositivo para conocer su ubicación en el parking. Es el trabajo más importante, y a su vez, el más laborioso. Toda la elaboración del proyecto se basará en el diseño que realicemos de la red junto a sus materiales.
- *Aprovisionamiento de materiales y recursos.* Los materiales necesarios para llevar a cabo el despliegue de la red dependerán del diseño realizado. Hay que comprar los materiales especificados en el diseño al precio más asequible, siempre teniendo en cuenta la relación calidad-precio.
- *Programación de los dispositivos.* Es el trabajo final de la primera fase muy importante para el desarrollo final del proyecto, ya que da paso a la segunda fase. Una vez tenemos adquiridos los recursos, es momento de programarlos según las especificaciones de cada dispositivo. En esta parte debemos de asegurarnos que cada periférico funciona como se ejemplifica en el diseño, hay que comprobar que se realizan correctamente las comunicaciones entre los dispositivos de la red y que cumplen las funciones programadas.

Posteriormente, conocidos los trabajos anteriores, es momento de comenzar con el desarrollo de la **segunda fase**. Se compone de tres partes:

- *Estructuración de la instalación.* Antes de realizar la instalación basada en los planos del diseño y la programación de los materiales, debemos realizar una planificación de cómo se desplegarán los materiales y recursos en el lugar. Haciendo este desglose, además de conseguir una rápida y eficaz instalación, evitaremos causar inconvenientes a los usuarios externos que se encuentren dentro del espacio de trabajo.
- *Instalación y despliegue de la red WSN.* Elaborada la planificación de la instalación, es momento de realizarla paso a paso buscando la máxima eficiencia en su desarrollo técnico. Este trabajo debe ir basado en la planificación anterior, es el más importante dentro de esta primera fase porque a partir de él hemos basado y programado todos los trabajos anteriores.

- *Testeo y corrección de errores.* Es el trabajo final del proyecto. Realizada la instalación ya solamente quedaría testear la comunicación de red y que todo funciona como se ha planteado a lo largo del proyecto. En caso de haber algún error, se notificaría y se buscaría la manera más adecuada para solucionar el problema.

Introducidas las bases estructurales del desarrollo del prototipo en el que basamos el estudio, es momento de proceder a su explicación detallada para darle forma y llegar al objetivo que buscamos desde el inicio de este apartado y, crear el prototipo idóneo para el servicio demandado por la empresa. Antes de proseguir con el siguiente apartado, mostraremos los trabajos mencionados con una estructura WBS del prototipo basado en LoRaWAN. Se trata de una metodología desarrollada en los Estados Unidos, utilizada para estructurar y definir todas las actividades de un proyecto de ingeniería llevado a cabo de una manera planificada y organizada.

Estructura WBS del prototipo LoRaWAN

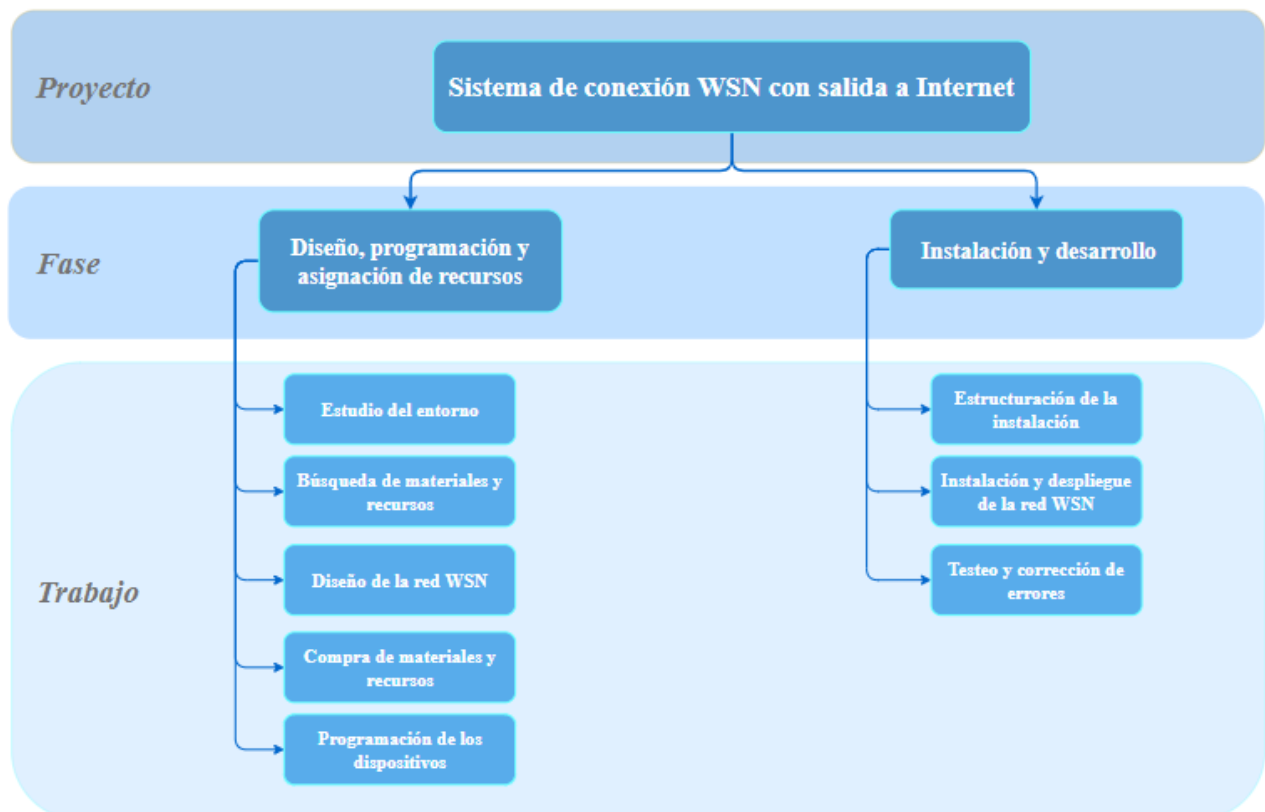


Figura 3.2. Estructura WBS

3.3. Desarrollo del proyecto

En este apartado expondremos el diseño y desarrollo de nuestro prototipo con LoRaWAN en función de las bases redactadas a lo largo del estudio. Simularemos técnicamente la realización del proyecto en un espacio real, cotidiano y adaptado a la instalación de sensores de cargadores para coches eléctricos. Para la realización de este apartado tendremos en cuenta las bases del modelo WBS, expuesto anteriormente, constituyendo la **primera fase** de este modelo la base de nuestro TFG. La segunda fase, orientada a la instalación de la red no será tratada en nuestro estudio, de forma que se pondrá como una posible línea futura.

En este caso nuestro parking consta de 2 pisos con un total 873 plazas de coches donde desplegaremos los correspondientes dispositivos LoRa en cada una de las plazas adaptadas al cargado de coches eléctricos. La ubicación del *gateway* LoRa estará lo más próximo de la superficie para poder tener una mejor cobertura y salida a Internet.

3.3.1. Estudio del entorno

El primer paso para una buena gestión del proyecto es conocer el entorno donde vamos a desplegar nuestra red WSN. Haciendo este análisis podremos saber los parámetros que necesitará nuestro *hardware*, y así, dar una funcionalidad completa a nuestro prototipo.

Ubicación del entorno de trabajo

Nombre de la edificación	Parking Subterráneo PCTCAN [33]	
Dirección	C/ Polígono Parque Científico y Tecnológico, 6, 39011 Santander, Cantabria	
Provincia	Cantabria	
Ayuntamiento	Santander	
Coordenadas Geográficas	Latitud	43° 27' 7,76'' N
	Longitud	3° 52' 28,34'' O
Coordenadas U.T.M.	X	429.233,06
	Y	4.811.401,47
	Huso	30

Tabla 3.1. Ubicación del entorno

Tipo de extensión

Parking/Aparcamiento		
Estación	Subterránea	
Nº pisos	2	
Nº plazas	843 de 10-13 m ²	30 de 20 m ²
Nº plazas adaptadas a EV¹³	163 de 10-13 m ²	22 de 20 m ²
Nº accesos	2 de entrada y salida	

Tabla 3.2. Características de la extensión

Planos de situación

Plano general del entorno de trabajo

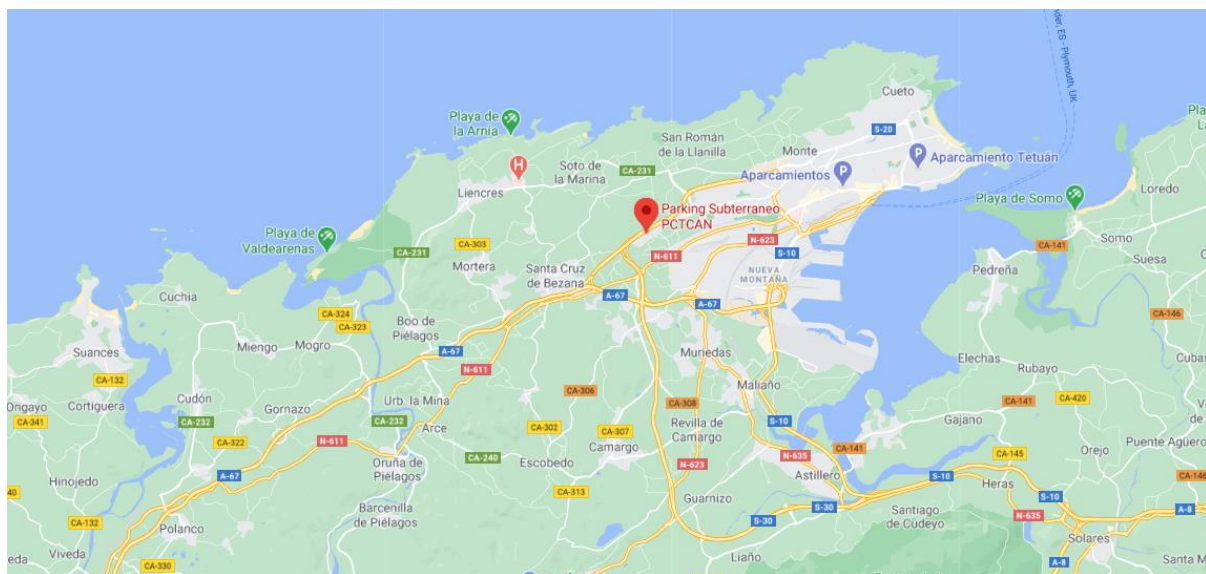


Figura 3.3. Plano general de la infraestructura

¹³ Electric Vehicle (EV), del inglés vehículo eléctrico.

Planos de la ubicación



Figura 3.4. Ubicación del parking

Dirección:

GD (grados decimales)*

Latitud:

Longitud:

GMS (grados, minutos, segundos)*

Latitud: N S ° ' "

Longitud: E O ° ' "

* Sistema Geodésico Mundial 1984 (WGS 84)

Figura 3.5. Coordenadas geográficas

Lugar:

GMS:

Decimales:

UTM:

Figura 3.6. Coordenadas UTM

Foto de los accesos al parking

Entrada/salida de vehículos 1:



Figura 3.7. Acceso 1

Entrada/salida de vehículos 2:



Figura 3.8. Acceso 2

3.3.2. Material y recursos

En esta parte del proyecto detallaremos el uso del *hardware* LoRa que emplearemos para crear el diseño de la red inalámbrica, una vez conocido el entorno donde vamos a desplegarla. Para tener una automatización del proceso comunicativo con el *hardware* que vamos a desplegar, debemos hacer un análisis comprensivo de cómo utilizar los módulos LoRa y la funcionalidad

que presentan. Hecho esto podremos programarlos y darles la funcionalidad que buscamos en nuestro estudio.

LoRa end device

El terminador o *end device* que utilizaremos será el *TTGO LoRa32 V1.0*. Este módulo irá conectado a cada sensor de carga de EV y se encargará de establecer la comunicación del sensor con el *gateway* mediante ondas de radiofrecuencia.

End devices TTGO LoRa32 V1.0

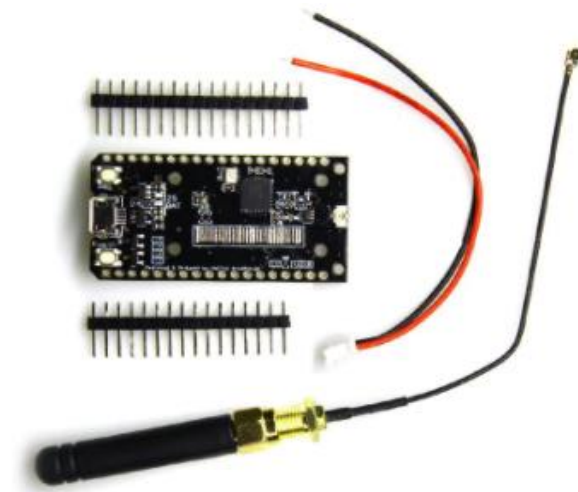


Figura 3.9. *TTGO LoRa32 V1.0* [34]

Hemos decidido escoger este dispositivo principalmente porque cumple con las características, mencionadas anteriormente, que buscamos en nuestro proyecto. La programación de este módulo es muy sencilla e intuitiva, además, presenta una fácil adaptación a la hora de vincularlo con otro dispositivo, ya que cumple con unos estándares muy estrictos que hacen posible su desarrollo en proyectos de IoT. En la *Tabla 3.3* veremos los principales requisitos del dispositivo.

Tensión de trabajo	3,3 - 7 V
Consumo de corriente trabajo	10-14 mA
Frecuencia de trabajo	868 MHz
Sensibilidad	-98 dBm
Potencia de transmisión	20 dBm
Tasa de transmisión	150 Mbps máx
Temperatura de trabajo	-40 °C - 90 °C
Consumo de corriente reposo	0,2 μ A dormido - 1,5 μ A ocupado

Tabla 3.3. *Especificaciones TTGO LoRa32 V1.0*

Complementariament, este terminador permete dar connectivitat Wi-Fi y Bluetooth, fent fàcil la seva adaptació a estàndards molt comuns en el àmbit de les comunicacions y escalable a programacions més complexes. En nostre projecte no se ha plantejat que els end devices tinguin connexió directa amb sortida a Internet via Wi-Fi perquè, com ja hem explicat, la comunicació utilitzant aquest protocol no és la apropiada, al tractar-se d'un espai subterràneu. Pot ser que, en un futur pròxim, amb el desenvolupament de noves tecnologies, es resolgui aquest problema y la característica mencionada el faci competent per al desenvolupament de diferents funcionalitats, sense necessitat de renovar el *hardware*.

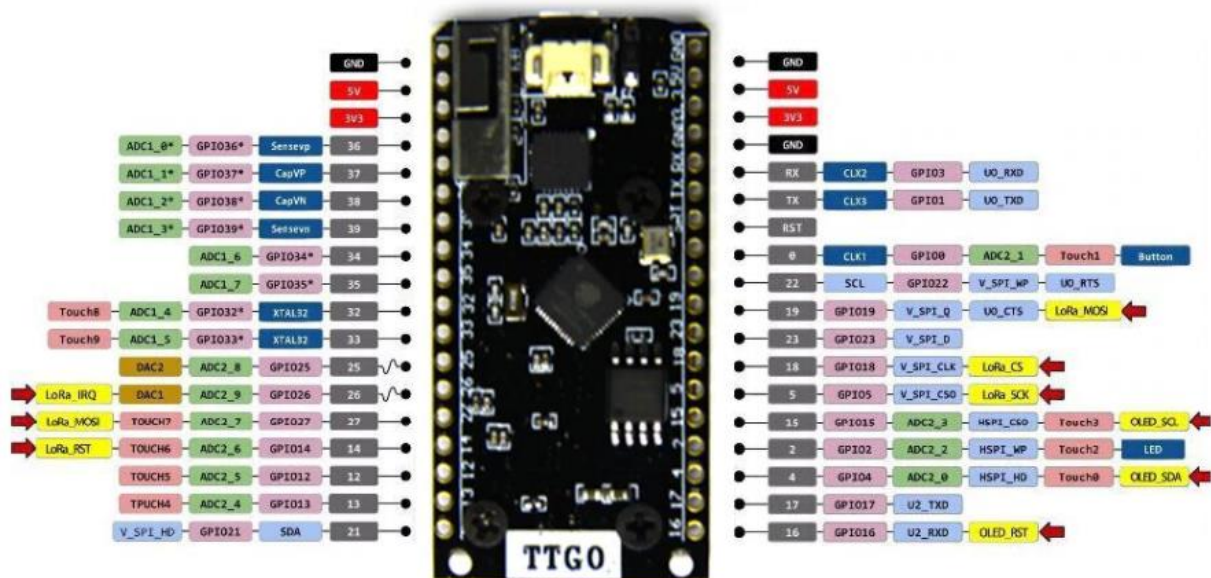


Figura 3.10. Diagrama pin de TTGO LoRa32 V1.0 [34]

LoRa gateway

Com ja se ha comentat a lo llarg del projecte, es requereix d'una pasarela per poder subir els dades dels sensors a Internet. En el cas del present projecte no hi ha cap disponible dins del radi d'abast dels sensors ja que, fent una cerca, la més propera que hem trobat es troba a Camargo (Santander), a 4,16 km de distància del parking on realitzarem l'obra. De manera que és necessari adquirir y configurar una, ja que la assignada per TTN (The Things Network) queda allunyada de la nostra ubicació.

Una pasarela és un *gateway* que inclou un concentrador LoRa que consecuentment, permet rebre paquets LoRa. Aquests paquets seran convertits a TCP/IP y enviats al servidor de red mitjançant una connexió del tipus LAN (Ethernet, Wi-Fi, 3G/4G). En conseqüència, una solució bastant econòmica per desenvolupar aquest apartat consisteix en utilitzar un *gateway* d'un únic canal. Existeixen pasareles de 8 canals que permeten l'accés d'un nombre molt superior de nodes finals, però el seu preu és bastant elevat.

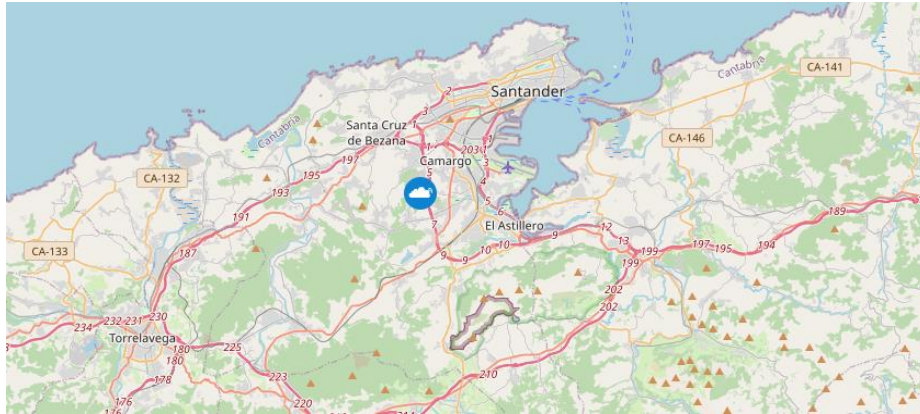


Figura 3.11. Gateway TTN más cercano

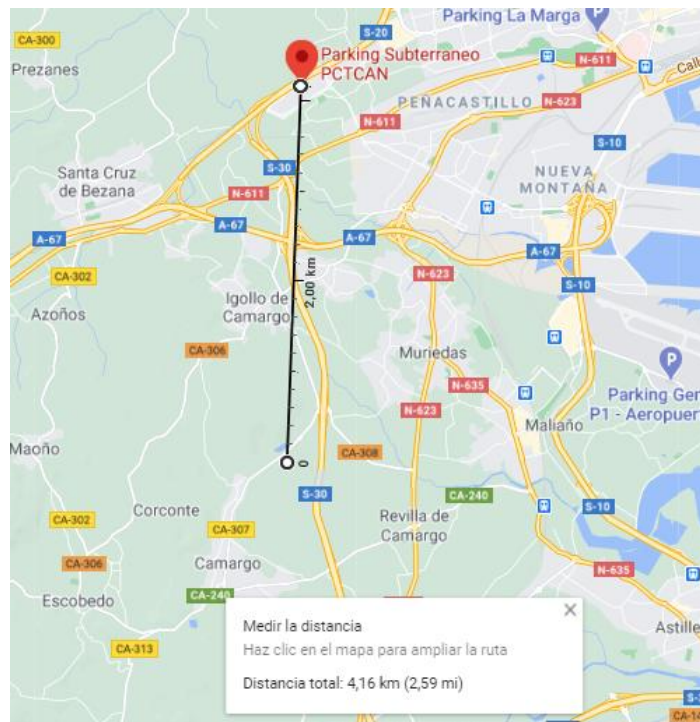


Figura 3.12. Distancia entre gateway y ubicación del entorno

Además, vamos a desplegar un número elevado de *end devices* dentro del espacio subterráneo y queremos la máxima eficiencia a la hora de transmitir la información, sin que haya colisiones cuando tengamos mucho tráfico de datos. Un *gateway* con salida a Internet que tenga conectividad LoRa con todos los sensores, es lo más idóneo para el desarrollo de nuestro estudio, siendo el encargado de comunicar la información entre los terminadores y la base de datos de la empresa.

Para que el *gateway* pueda tener salida a Internet sería necesario tener acceso a una red LAN cercana. Como la intención es poner la pasarela lo más cercana al exterior o en todo caso, junto a accesos cercanos de convivencia de usuarios, supondremos que la zona de instalación del *gateway* está dotada de dichos accesos a Internet.

Gateway Dragino LG01-P-868-EU



Figura 3.13. Dragino LG01-P-868-EU [35]

La *Figura 3.13* muestra la pasarela que utilizaremos para nuestro prototipo. Nos hemos decantado por este *gateway* por su fácil escalabilidad y por su programación intuitiva, que lo hace programable con diferentes IDEs, según las funciones que queramos para su uso. Al trabajar con tan solo un canal para todos los dispositivos y al encontrarse cercano a un espacio con conectividad a Internet, lo hacen la opción más económica en la propuesta de nuestro prototipo. Además, proporciona comunicación de amplio alcance de espectro ultra largo y alta inmunidad a las interferencias, características necesarias en el entorno donde vamos a desplegarlo.

En la *Tabla 3.4* presentaremos las propiedades que no han hecho decantarnos por este *gateway*.

Puerta de enlace de canal único de código abierto a 868 MHz
Interfaz Wi-Fi, puerto Ethernet y puerto host USB
Conexión a Internet a través de Wi-Fi, 3G o 4G
Arduino IDE compatible
Alcance máximo en LoRa: 5 - 10 km
Consumo de energía: 12 V (DC)
Wi-Fi: 802.11 b/g/n
Chip LoRa: SX 2176 / 78

Tabla 3.4. Características Dragino LG01-P-868-EU

Hasta aquí, se ha realizado una elección de diferentes opciones de hardware existentes dentro del ámbito de las comunicaciones basadas en LoRa. Se trataba de realizar una búsqueda técnica entre una serie de dispositivos existentes para poder determinar cuál de ellas puede ser la más interesante para implementar nuestro sistema de conectividad IoT en un parking. Por otra parte,

es necesario aclarar que estos módulos se pueden emplear de dos formas: bien utilizando enlaces punto a punto o bien formando parte de una red en estrella. En nuestro caso recurriremos a una red en estrella, en donde los nodos finales se conectan a un *gateway* para poder subir información a una base de datos en Internet.

Finalmente, podemos determinar que con tan solo dos dispositivos podemos ofrecer la conectividad que se especifica en el estudio. Dependiendo de la extensión del área necesitaremos un número diferente de ellos. Al ser una comunicación de red *wireless* no es necesario disponer de más material que los dispositivos mencionados, siendo una opción muy asequible e interesante para otros proyectos de IoT.

Como hemos mencionado, suponemos que un espacio mínimo del parking estará dotado de conectividad a Internet para poder configurar el *gateway*. De no ser así, obviamente necesitaríamos más recursos (router Wi-Fi, chips GSM con 3G/4G, cable Ethernet) que los expuestos anteriormente para dar esa conectividad a una zona y que la red funcione completamente. Esto nos supondría realizar un estudio adicional, tal y como se menciona en la entrevista con el Sr. Sánchez, ya que necesitamos hacer otro prototipo, con otro protocolo, adaptado a un entorno específico del parking. Preferimos no adentrarnos a este desarrollo adicional porque quedaría fuera de las bases de nuestro proyecto, dejándolo como una línea futura para la continuación de este proyecto.

3.3.3. *Diseño*

Conocidos los instrumentos que vamos a desplegar y adaptando su elección en base al entorno donde vamos a instalarlos, es el momento de crear el diseño sobre el plano y especificar la colocación de nuestros *end devices* y *gateway* en el parking adaptado a EV.

El parking cuenta con un total de 873 plazas, donde 185 están habilitadas para la carga de coches eléctricos. Por lo tanto, instalaremos un total de 185 módulos LoRa (85 en la planta -1 y 100 en la planta -2) en cada una de las plazas de vehículos eléctricos que se vincularan a un único *gateway*. El *gateway* lo configuraremos próximo al espacio de zona Wi-Fi cedida por el/los propietario/s del parking. Dicho espacio viene completamente adaptado para poder instalar una pasarela LoRa.

En los planos mostramos el diseño realizado para las dos plantas del parking subterráneo PCTCAN con sus respectivos terminadores y pasarela LoRaWAN. En el diseño de los planos no vienen dibujados los cargadores de coches eléctricos, aunque se supone que donde instalamos un módulo LoRa se requiere de un cargador vinculado a él. La conectividad entre *gateway* y los módulos LoRa será mediante ondas de RF, no especificamos la vinculación electromagnética entre dispositivos en el plano, sino sólo su posicionamiento dentro del entorno.

3.3.4. *Aprovisionamiento de materiales y recursos*

En esta sección debemos comprar los materiales y recursos en función del diseño y los análisis realizados en los apartados anteriores. En esta parte buscamos la mejor relación calidad-precio en páginas de compra seguras, intentando encontrar el máximo descuento posible a la hora de hacer un pedido elevado de los módulos LoRa.

Al contar con tan solo un *gateway* en el despliegue de este prototipo, tenemos que realizar una búsqueda exhaustiva para encontrarlo al mejor precio y con las prestaciones mencionadas en su análisis de compra.

Las páginas que hemos utilizado para la búsqueda y compra de los dispositivos han sido las comunes para la compra de dispositivos tecnológicos, unos ejemplos de ellas son: Amazon, AliExpress, Robotshop, Arduino, Dragino, etc.

3.3.5. *Programación de dispositivos*

Prototipo inicial

Conseguidos todos los propósitos anteriores llegamos al trabajo final de la fase 1 de nuestro prototipo. En este apartado vamos a detallar y exponer cómo se llevará a cabo la programación de los *end devices* y *gateway/s* para su uso dentro de la red del parking subterráneo. También, mencionaremos los entornos de desarrollo donde hemos programado los MCU de nuestros dispositivos para ser conocedores de nuestra programación al completo.

Para la realización de esta parte hemos un prototipo escalado respecto al expuesto en el estudio precedente. La búsqueda de nuestro trabajo de final de grado (TFG) siempre ha sido realizar el mejor estudio posible junto a su prototipo más afín, para así poder hacer realidad el servicio de IoT expuesto a lo largo de nuestro proyecto.

Limitaciones de programación

Uno de los problemas que nos obligan a escalar el proyecto, es que la exposición y presentación de nuestro proyecto al completo dentro del marco temporal que otorga el TFG es muy limitado para las magnitudes de nuestro prototipo inicial. Cabe destacar que para el desarrollo de este prototipo solo disponía de un mes, ya que la mayor parte del trabajo TFG ha consistido en realizar el estudio detallado en el apartado 2.

Por otra parte, la falta de acceso a los sensores de los cargadores de coches eléctricos y la base de datos del sistema de dichos cargadores nos ha nos ha impedido desarrollar nuestra primera idea. Al no poder trabajar con estos sensores que tienen que ir conectados a los terminadores LoRa, ni poder acceder a la base de datos donde se almacena la información que debe comunicarse con la pasarela LoRa, resulta inviable poder realizar la vinculación del protocolo LoRa con las tecnologías de comunicación mencionadas anteriormente.

Además, el factor económico también ha sido otro motivo de peso a considerar. El coste completo de nuestro prototipo original sería mucho más elevado al expuesto en este apartado. Es importante destacar que un caso es para un servicio empresarial y otro para un trabajo universitario orientado al aprendizaje.

Por lo tanto, por tiempo, costes y falta de recursos, se decide desarrollar un prototipo a baja escala para adentrarnos en la programación de los dispositivos, y así, en una línea futura implementarlo a la realidad expuesta en apartados anteriores.

Prototipo demostrador

Para suplir todas las limitaciones expuestas y poder completar la fase 1 de nuestro proyecto, hemos decidido simular nuestro prototipo inicial utilizando como material solo dos dispositivos *TTGO LoRa32 V1.0*. Con estos dos módulos conseguiremos simular, a baja escala, la idea inicial de nuestro trabajo para poder exponer una idea física del funcionamiento de nuestra red WSN dentro del parking subterráneo.

Como se puede ver, son los mismos dispositivos que los propuestos en el prototipo inicial, pero en este caso en vez de utilizar los módulos como *end devices* vinculados al *gateway*, utilizaremos uno de ellos como *end device* y el otro como *gateway*.

El dispositivo *end device* se encargará de gestionar la información que reciba para mandarla, mediante comunicación LoRa punto a punto, al *gateway*. La gestión de la información la llevará a cabo el propio *end device*, que mediante la IDE de Arduino programaremos la simulación de una información a enviar porque no tenemos acceso a los sensores de cargadores. Resumidamente, en el mismo *sketch* del *end device* generamos, encapsulamos y enviamos la información mediante LoRa al *gateway* para que posteriormente sea subida a la nube.

Una vez el *gateway* recibe la información, la procesa y la sube a una base de datos. De esa manera la información podrá ser legible y procesada remotamente, sin necesidad de estar vinculada a los módulos LoRa. Para conseguir esto, programaremos el módulo considerado como *gateway* a través de la IDE de Arduino. Además, en el *sketch* programaremos la detección de la información que nos llega del *end device* y la trataremos para poder subirla a la nube mediante el protocolo Wi-Fi. El servidor que utilizaremos para subir los datos en la nube será el servidor abierto Hive MQ.

Como hemos visto en las especificaciones, estos módulos son compatibles con Wi-Fi. Además de poder comunicarse mediante LoRa entre ellos, podemos utilizar uno de ellos como pasarela, para de esta forma poder subir la información que requiramos a la base de datos. Teniendo en cuenta esta consideración, es necesario configurar el módulo para que pueda conectarse a la red Wi-Fi del espacio en el que se encuentra; programación que se también tiene lugar en el *sketch* del *gateway*.

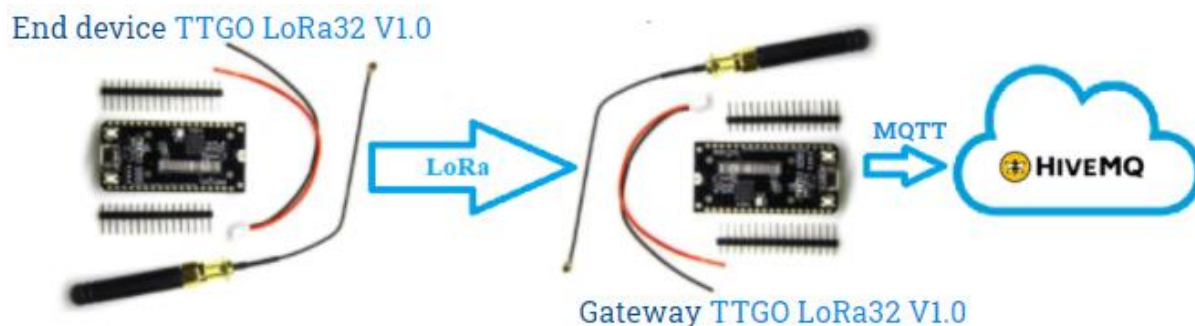


Figura 3.14. Ejemplificación del prototipo demostrador

Verificación de la comunicación LoRa

El primer paso de programación debe ser la implementación de la comunicación entre los dispositivos mediante el protocolo LoRa. Debemos comprobar que se establezca con éxito la comunicación y el envío de paquetes LoRa entre el *end device* y el *gateway*. Aprovechando la programación del establecimiento de comunicación LoRa, también hemos querido poner a prueba la potencia que puede llegar a recibir el *gateway*, en función de la distancia que se encuentran los dos dispositivos.

En los **anexos 1 y 2** podremos encontrar los *sketches* de Arduino donde se programan los dos módulos para la verificación de la comunicación LoRa entre dispositivos y la potencia recibida del paquete en función de la distancia entre dispositivos. Encontraremos dos códigos, uno para el *end device*, que enviará los paquetes LoRa, y otro para el *gateway*, que recibirá el paquete y mostrará su potencia recibida. Las librerías a incluir para la compilación de los *sketches* son: la *SPI.h* y la *LoRa.h*.

En la *Tabla 3.6* se representa la potencia, a través de la lectura del RSSI (en dBm), recibida al *gateway* en función de la distancia (en metros) desde la que el *end device* le envía el paquete LoRa. Cabe destacar que a la hora de hacer esta comprobación nos encontrábamos en un espacio abierto y con línea de visión directa entre los dos dispositivos, en la *Tabla 3.5* se puede observar las características del entorno.

Datos del entorno de medición:

Espacio	Playa de Ponent	
Dirección	C/ Passeig de Miramar, 42, 43840 Salou, Tarragona	
Provincia	Tarragona	
Ayuntamiento	Salou	
Coordenadas	Latitud	41° 4' 29,18'' N

Geogràfiques del Gateway	Longitud	1° 7' 50,35'' E
Información Meteorológica	Temperatura	21°
	Humedad	52 %
	Viento	18 km/h
	Precipitación	0 cm
	Presión	1015 hPa
	Visibilidad	12,9 km
	Índice UV	5

Tabla 3.5. Datos del entorno de medición

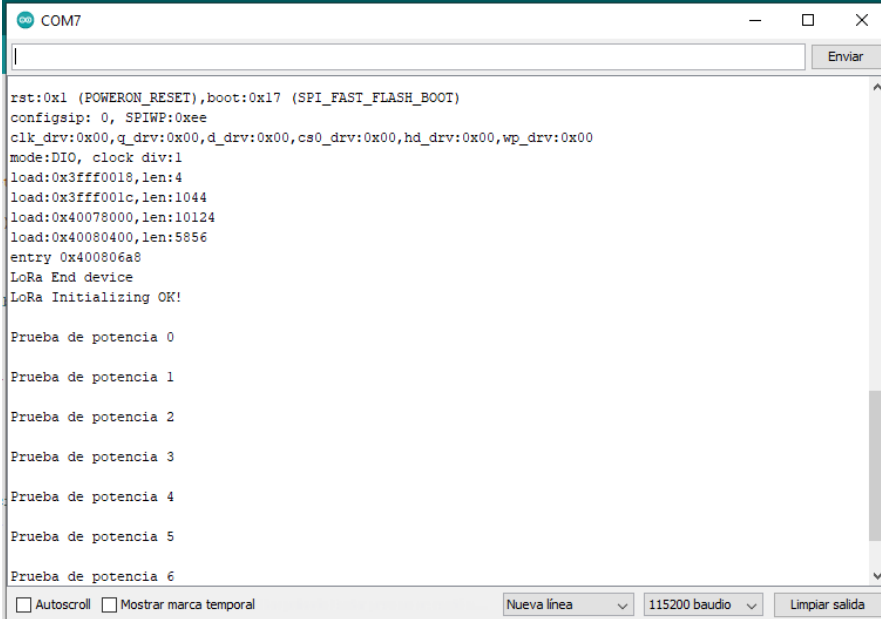
Distancia (m)	Potencia RSSI (dBm)
1	-58
10	-73
50	-98
100	-112
200	-115
450¹	-116

1. Máxima distancia de recepción de paquetes sin errores.

Tabla 3.6. Relación distancia - potencia recibida

Pruebas de funcionamiento:

A continuación, en las siguientes imágenes vamos a mostrar las capturas del ‘monitor serie’ de la IDE de Arduino para poder observar cómo se establece la comunicación LoRa entre los dos dispositivos y también, cómo puede verse la potencia recibida del paquete en el *gateway* LoRa.



```

COM7
rst:0x1 (POWERON_RESET),boot:0x17 (SPI_FAST_FLASH_BOOT)
config:0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:10124
load:0x40080400,len:5856
entry 0x400806a8
LoRa End device
LoRa Initializing OK!

Prueba de potencia 0

Prueba de potencia 1

Prueba de potencia 2

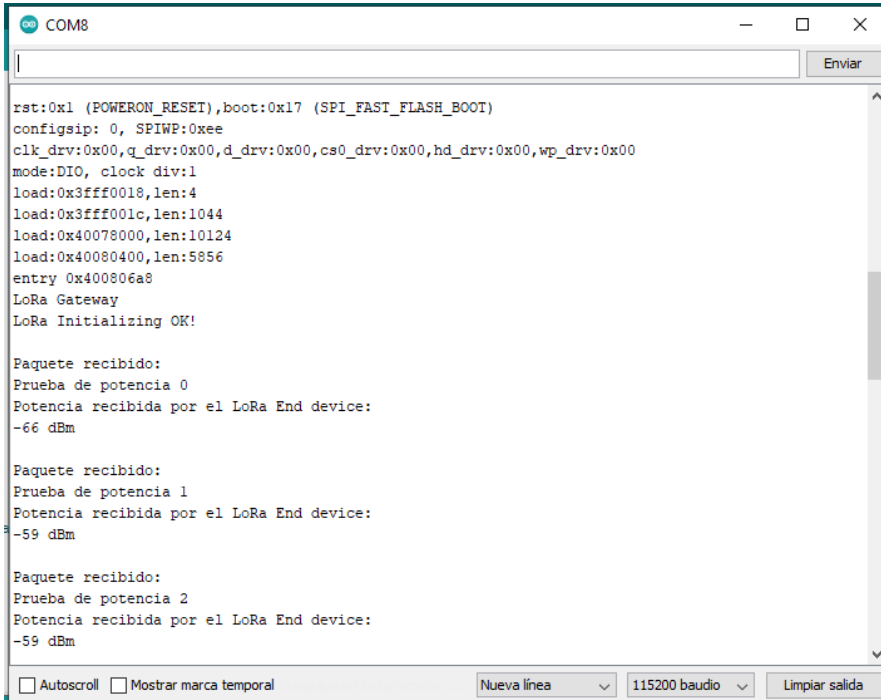
Prueba de potencia 3

Prueba de potencia 4

Prueba de potencia 5

Prueba de potencia 6
  
```

Figura 3.15. Monitor serie del end device, se observa el envío de los paquetes sincronizadamente



```

COM8
rst:0x1 (POWERON_RESET),boot:0x17 (SPI_FAST_FLASH_BOOT)
config:0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:10124
load:0x40080400,len:5856
entry 0x400806a8
LoRa Gateway
LoRa Initializing OK!

Paquete recibido:
Prueba de potencia 0
Potencia recibida por el LoRa End device:
-66 dBm

Paquete recibido:
Prueba de potencia 1
Potencia recibida por el LoRa End device:
-59 dBm

Paquete recibido:
Prueba de potencia 2
Potencia recibida por el LoRa End device:
-59 dBm
  
```

Figura 3.16. Monitor serie del gateway, se observa la recepción de los paquetes sincronizadamente junto a su potencia recibida

Creación y envío de la información de los sensores para cargadores de EV

Al no poder acceder a los sensores de cargadores para coches eléctricos, debemos simular la recepción de una información y posteriormente enviarla mediante el estándar LoRa al *gateway*. Para conseguir esto, debemos crear un juego de pruebas en el *sketch* (IDE Arduino) del *end device* que simule la obtención de un mensaje del sensor de coche para posteriormente, tratar ese mensaje, encapsularlo en un paquete LoRa y finalmente, enviarlo al *gateway*.

El código del *sketch* se encuentra implementado en el **anexo 3**, y como juego de pruebas hemos decidido realizar la simulación de coches que se conectan y desconectan a los cargadores, o también conocer si alguno de ellos está cargado completamente. Por lo tanto, recibiremos como mensaje la matrícula de cada coche junto a la palabra “*connected*”, “*disconnected*” o “*charged*”, según la acción que realice el vehículo, por ejemplo, “0000AAAconnected”. Haciendo esto conseguimos obtener un mensaje simulado que podría enviar un cargador de EV, y una vez lo leemos y convertimos a paquete LoRa, realizamos su envío al *gateway*.

Pruebas de funcionamiento:

En estas pruebas podemos observar como el juego de pruebas funciona correctamente. Se genera un mensaje aleatorio, se convierte en un paquete LoRa y el *end device* lo envía al *gateway*. En la *Figura 3.17* podemos observar la llegada del paquete al *gateway*.

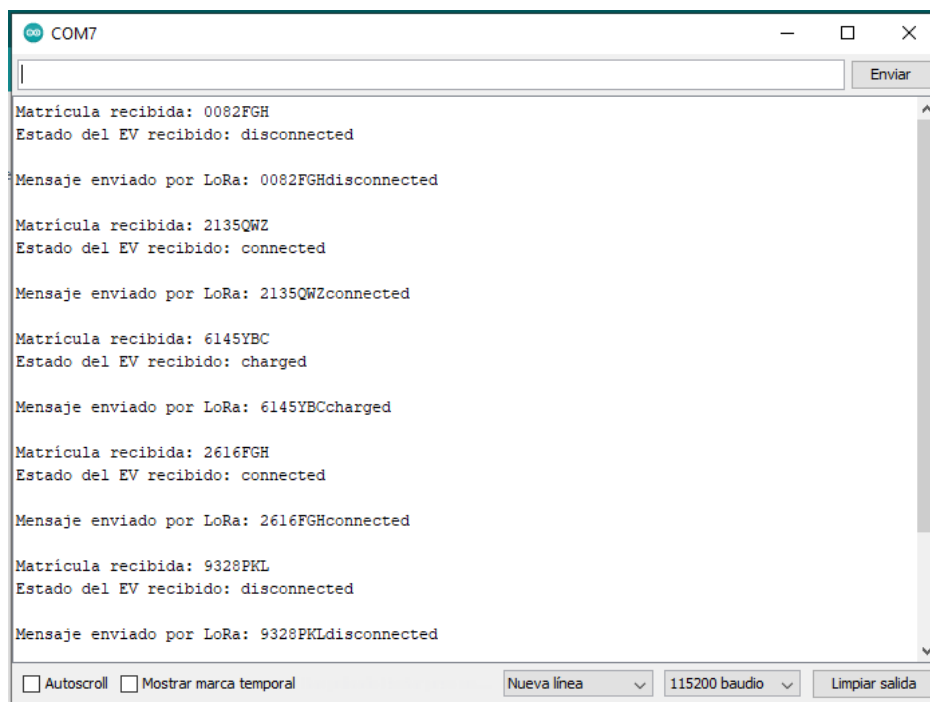
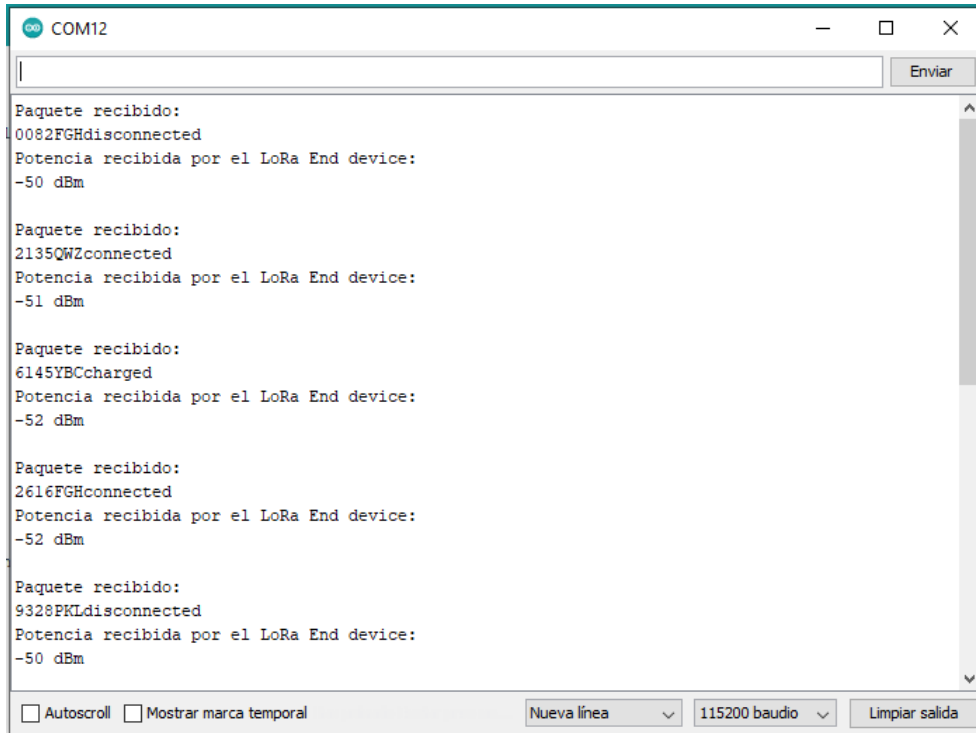


Figura 3.17. Recepción de mensaje y envío mediante LoRa al *gateway*



```
COM12
Paquete recibido:
0082FGHdisconnected
Potencia recibida por el LoRa End device:
-50 dBm

Paquete recibido:
2135QWZconnected
Potencia recibida por el LoRa End device:
-51 dBm

Paquete recibido:
6145YBCcharged
Potencia recibida por el LoRa End device:
-52 dBm

Paquete recibido:
2616FGHconnected
Potencia recibida por el LoRa End device:
-52 dBm

Paquete recibido:
9328PKLdisconnected
Potencia recibida por el LoRa End device:
-50 dBm

 Autoscroll  Mostrar marca temporal
Nueva línea 115200 baudio Limpiar salida
```

Figura 3.18. Llegada del paquete LoRa al gateway

Subida de la información a la base de datos Hive MQ

Conseguido el objetivo del apartado anterior y observando que la información llega correctamente al *gateway*, es momento de que en su *sketch* programemos la subida a la nube de estos paquetes LoRa mediante el protocolo MQTT. Conocemos que el módulo *gateway* con el que trabajamos permite tener conectividad vía Wi-Fi. Por lo tanto, para realizar la subida de datos a la nube deberemos vincular el dispositivo a la red Wi-Fi de nuestro entorno, para posteriormente, suscribirnos al broker abierto y online, Hive MQ.

La implementación completa del código para el *gateway* se encuentra en el **anexo 4**, como ya hemos mencionado en el anterior párrafo, en este código encontramos la programación de la recepción de los paquetes LoRa, junto a la subida de los paquetes al broker Hive MQ. Para facilitar la comprensión de este último aspecto, voy a comentar las líneas de código más relevantes que nos han permitido poder subir los datos al bróker.

Lo primero que tenemos que hacer es incluir al *sketch* las librerías *WiFi.h* y *PubSubClient.h*. Una vez incluidas, debemos vincularnos a la red Wi-Fi de nuestro entorno de trabajo y acceder al bróker online Hive MQ mediante su IP pública y el puerto TCP, en la página web del mismo bróker nos especifican estos datos.

You can access the broker at:

Broker: **broker.hivemq.com**

TCP Port: **1883**

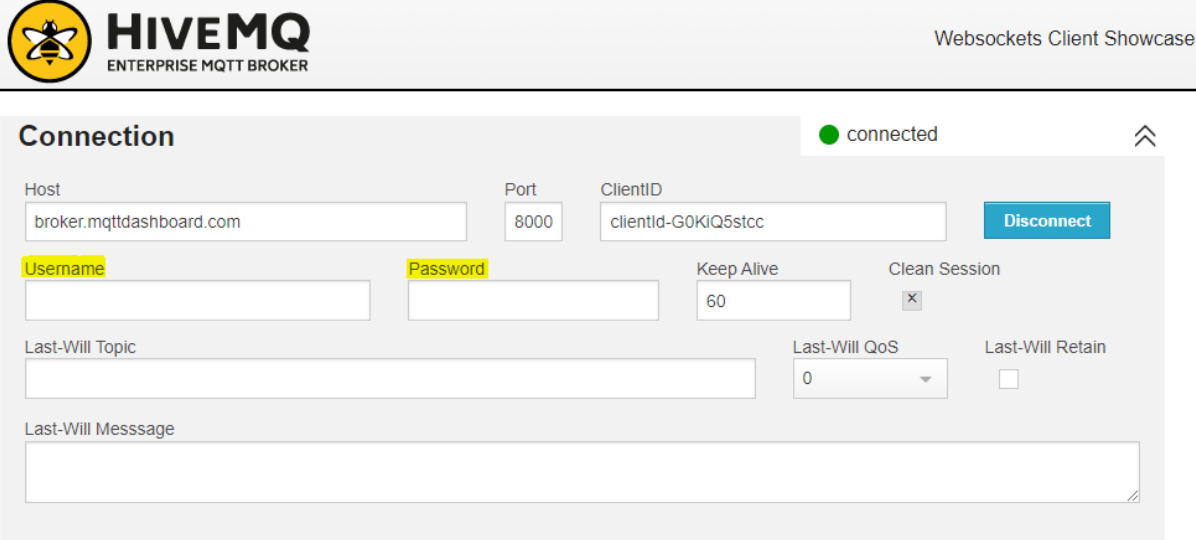
Websocket Port: **8000**

Figura 3.19. Datos para el acceso a HiveMQ

```
const char* ssid = "SercommFB50";
const char* password = "XXXXXXXXXXXX";
const char* mqttServer = "broker.hivemq.com"; // broker IP
const int mqttPort = 1883; // Puerto TCP de hiveMQ. Mensaje MQTT no encriptado
const char* mqttUser = "";
const char* mqttPassword = "";
```

Figura 3.20. Código de la IDE de Arduino para la vinculación al Wi-Fi y bróker HiveMQ

Como el tópic al que nos vamos a suscribir no requiere ni usuario ni contraseña, las dos últimas líneas que vemos en la *Figura 3.21* las dejamos vacías. En caso de querer proteger la información de ese tópic, sí sería necesario asignarle un usuario y una contraseña de seguridad, tanto en el bróker online como en las dos líneas de código de la imagen.



The screenshot shows the HiveMQ Websockets Client Showcase interface. At the top left is the HiveMQ logo (a bee) and the text "HIVEMQ ENTERPRISE MQTT BROKER". At the top right is the text "Websockets Client Showcase". Below this is a "Connection" section with a green "connected" status indicator and an up arrow icon. The connection settings are as follows:

- Host:
- Port:
- ClientID:
- Disconnect button:
- Username:
- Password:
- Keep Alive:
- Clean Session:
- Last-Will Topic:
- Last-Will QoS:
- Last-Will Retain:
- Last-Will Message:

Figura 3.21. Conexión a HiveMQ sin usuario ni contraseña

Finalmente, necesitamos crear el tópic Hive MQ en el que vamos a subir y leer nuestra información a/de la base de datos pública. Para este caso, hemos decidido asignarle un nombre único que sea difícil de coincidir con otros tópicos que otro usuario Hive pueda utilizar y alterar nuestra base de datos. Como podéis ver en la *Figura 3.22*, hemos decidido ponerle "ControlEV" para posteriormente, suscribirnos a este tópic en el bróker Hive MQ.

```

if (!client.connected()) {
  reconnect();
}
// Subida a HiveMQ mediante el topico ControlEV
int i = LoRaData.length() + 1; // one extra character for the null
char aux[i]; // buffer
LoRaData.toCharArray(aux, i); //conversion string to chararray
client.publish("ControlEV", aux); //envio el dato al topic en formato chararray
Serial.println("");
Serial.println("Paquete subido a HiveMQ: ");
Serial.println(LoRaData);
Serial.println("-----");
client.loop();

```

Figura 3.22. Suscripción al tópico HiveMQ desde la IDE de Arduino para subir la información

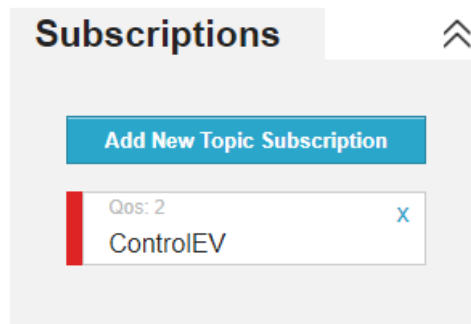


Figura 3.23. Suscripción al tópico HiveMQ para visualizar la información

Otra consideración sobre el código de la *Figura 3.22*, es que los datos que subamos a Hive tienen que ser de carácter “*charrarray*”, es decir, deben subirse como una cadena de caracteres. Por ello, como vemos en el código hacemos la conversión del carácter “*String*” recibido por LoRa a “*chararray*”, para así poder subir la información al bróker Hive MQ.

Prueba final del prototipo demostrador

Realizado este último apartado de programación, damos por concluida la programación del prototipo demostrador y observamos su correcto funcionamiento. Como podemos ver en la *Figura 3.24*, desde el “monitor serie” de la IDE de Arduino, observamos cómo llegan los paquetes al *gateway*, se conecta al bróker mediante el protocolo MQTT una vez tiene conectividad Wi-Fi, y como cabía esperar, los sube a Hive MQ. La *Figura 3.25* nos verifica la correcta subida de los paquetes a la base de datos con el tópico “*ControlEV*”.

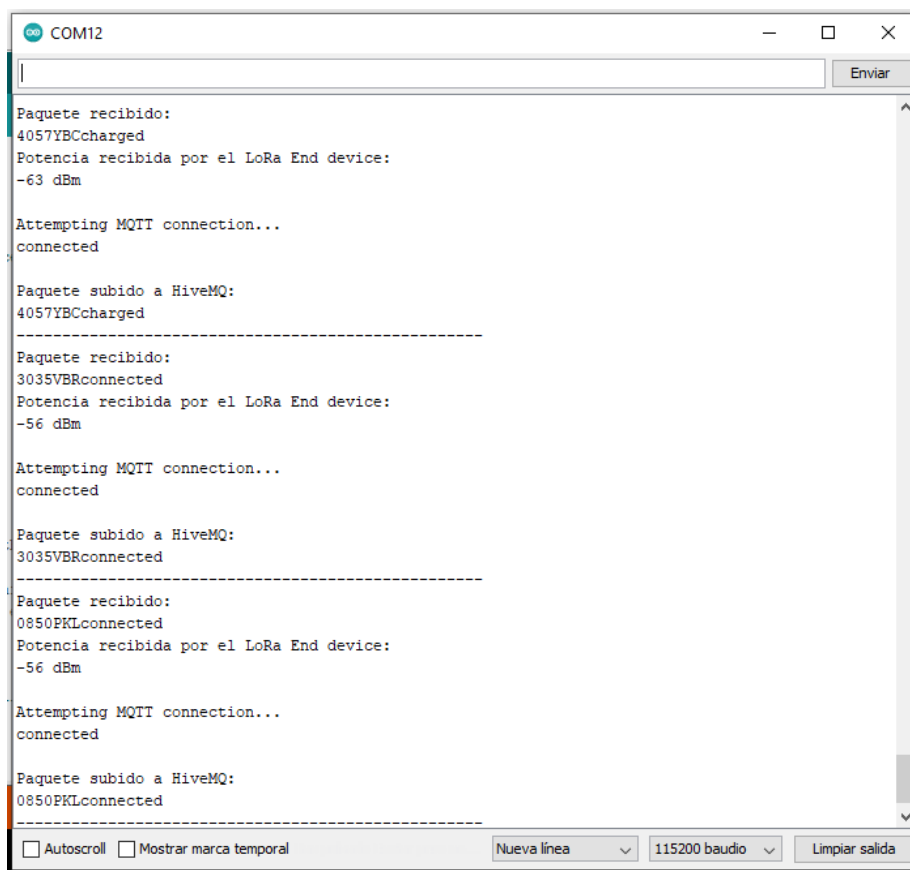


Figura 3.24. Recepción de paquetes LoRa del end device y subida de estos a HiveMQ

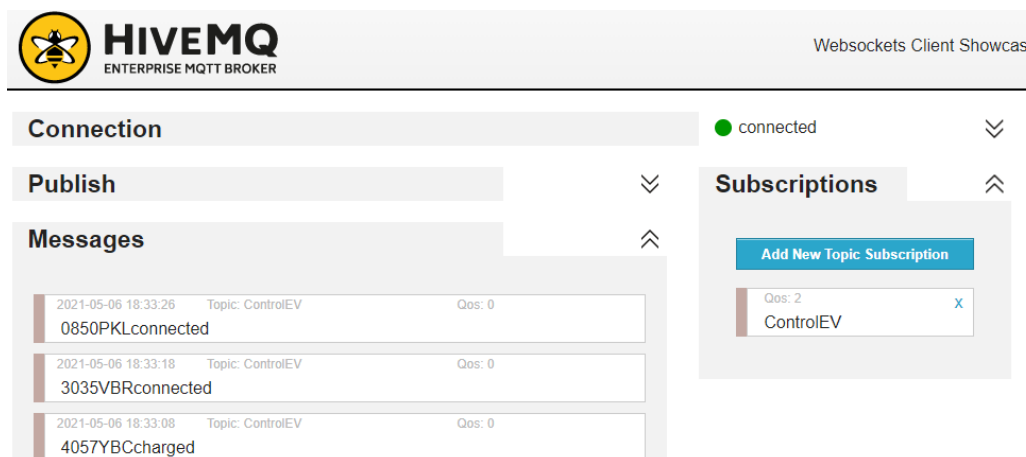


Figura 3.25. Lectura de los paquetes LoRa en la base de datos HiveMQ

Resumen de la programación

En este apartado de programación se ha propuesto y desarrollado una variante a menor escala del prototipo inicial. Dado que por tiempo, recursos y presupuesto no se ha podido desplegar el desarrollo del primer prototipo, hemos podido adaptar un prototipo demostrador que pueda servir como base y ejemplificación para un desarrollo futuro de servicio IoT.

Hemos conseguido realizar el objetivo principal de la simulación, conseguir que una información recibida en el *end device* se consiga comunicar, pasando por un *gateway*, a una

base de datos online. Haciendo uso de los dos protocolos comunicativos mencionados a lo largo del proyecto, LoRaWAN, para la comunicación a larga distancia entre los dos dispositivos, y Wi-Fi, para subir los mensajes recibidos a la nube.

En la *Figura 3.26* se muestra la instalación de los módulos según la función que deban realizar, ambos se encuentran en espacios distintos. A la izquierda tenemos el *end device*, alimentado por una batería portátil para poder generar paquetes LoRa y enviarlos al *gateway*. A la derecha podemos ver el *gateway*, conectado al ordenador para poder observar la llegada de los paquetes LoRa y posteriormente, subir esos paquetes vía Wi-Fi y MQTT a la base de datos.

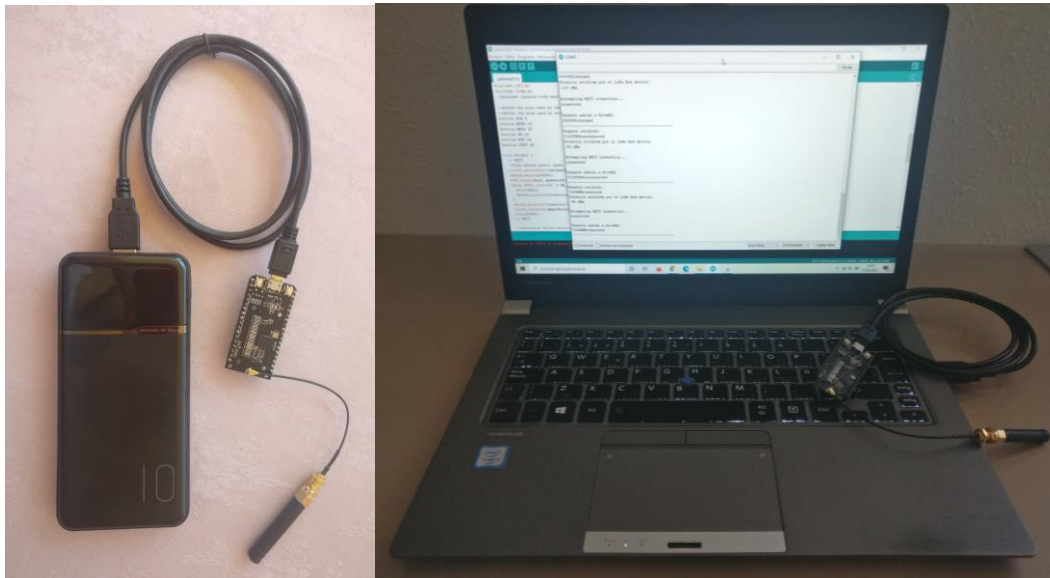


Figura 3.26. Instalación y funcionamiento de los módulos end device (izquierda) y gateway (derecha) LoRa

Conseguida la implementación del prototipo demostrador, podemos dar por concluido este apartado y proseguir con el final del prototipo basado en LoRaWAN. Los detalles de la programación se encuentran explicados y desarrollados en los anexos y a lo largo de los apartados anteriores, en los cuales ponemos de manifiesto su programación y el perfecto funcionamiento de los dos módulos LoRa.

3.3.6. Seguridad del prototipo demostrador

A pesar de que en las bases del estudio hemos mencionado que LoRa presenta un comportamiento muy seguro a la hora de crear sus enlaces, en el prototipo demostrador no la tenemos en cuenta de acuerdo con lo explicado en el estudio, de manera que:

- No hacemos uso de la *AppKey*. Al programar un envío broadcast, cualquier dispositivo que esté operando en la misma frecuencia de trabajo (868 MHz) y se encuentre dentro del rango de alcance, puede recibir los datos que se están transmitiendo entre los dos módulos LoRa.

- No encripta la informació con el uso de la *App Session Key*. Por ello, si se requiere su encriptado se debe programar en el *sketch* antes de enviarlo al módem de radiofrecuencia y una vez recibida en el receptor, desencriptarla.
- No utilizamos la *Network Session Key* garantizando seguridad a nivel de red. Al subir los datos a través del protocolo MQTT, no se aplica la seguridad IP completa entre el *gateway* y servidor. Con un *hackeo* simple de la IP es posible acceder a los datos de transmisión en la red.

Estos aspectos no son críticos en el sistema del prototipo demostrador debido a que los datos no suponen ningún activo de valor para nadie, no son susceptibles de robo, ya que se trata de un proyecto educativo y de simulación. Por ello, en principio no se procede emplear ningún sistema de seguridad que impida el acceso a los datos.

En todo caso, podría tenerse en cuenta de cara a mejoras futuras a implementar en el sistema si se quisiera realizar el prototipo considerado inicialmente en el parking de PCTCAN, donde este sí nos llevaría a trabajar con información confidencial. Por lo que sí sería interesante programar/verificar la seguridad LoRa para no comprometer datos relevantes de dichos sensores de cargadores de coches eléctricos.

3.3.7. Diagrama de Gantt

Para representar las actividades del WBS y planificar su realización en función del tiempo, decidimos elaborar un diagrama de Gantt. Gracias a esta herramienta podemos saber el tiempo total invertido en el prototipo con LoRaWAN y la secuencia de los trabajos para su realización.

Se pretende remarcar que para este diagrama hemos decidido desarrollar los trabajos de la fase uno del WBS, ya que como hemos mencionado anteriormente, la fase de instalación no ha sido posible llevar a cabo. De todas maneras, con este diagrama podremos observar cuánto tiempo nos ha llevado realizar la fase principal del prototipo y tener una idea de cuánto tiempo se tardaría en elaborar el proyecto completo.

Toda la fase de diseño, programación y asignación de recursos me supuso tuvo una duración total de 220 horas, que es equivalente aproximadamente a 28 días de trabajo. El diagrama se encuentra en el anexo 5 por su gran extensión.

3.3.8. Presupuestos

A continuación, en la *Tabla 3.7* se muestra el presupuesto del **prototipo demostrador**. En la tabla no se incluye el tiempo de diseño del sistema, ni construcción ni instalación del prototipo. Tampoco se tiene en cuenta el gasto de los recursos humanos para su desarrollo. Como se puede ver, el presupuesto, sin contar pequeño material adicional de instalación, es en total **19,00 €**.

Cantidad	Descripción	P. Unitario	Subtotal
2	TTGO LoRa32 V1.0	9,5 € / unidad	19,00 €
TOTAL	19,00 €		

Tabla 3.7. Presupuesto del prototipo demostrador

Además del presupuesto del prototipo demostrador, queremos mostrar el coste que supondría desarrollar el **prototipo inicial**, teniendo en cuenta los mismos aspectos que en el presupuesto anterior. Observando la *Tabla 3.8*, vemos que el precio total ascendería a **2024,76 €**, sin tener en cuenta el material para la vinculación de los dispositivos a los sensores de cargadores de EV, tampoco se tiene en cuenta el coste que supondría la instalación, ni el precio de suscripción al servicio que ofrecido para la conexión a la red Wi-Fi ni la mano de obra para su instalación.

Cantidad	Descripción	P. Unitario	Subtotal
185	TTGO LoRa32 V1.0	10,56 € / unidad ¹	1953,60 € ²
1	Dragino LG01-P-868-EU	61,53 € / unidad	61,53 €
TOTAL	2015,13 €		

1. A pesar del mismo módulo es más caro porque la compra anterior la realizamos en un momento de descuento.

2. Presupuesto basado en el precio unitario

Tabla 3.8. Presupuesto del prototipo inicial

Comparando los presupuestos, podemos afirmar que el factor económico ha sido crucial a la hora de diseñar el prototipo demostrador que simula el propósito de nuestro estudio.

3.3.9. Conclusiones sobre prototipo LoRaWAN

Si observamos el WBS expuesto al inicio de la redacción de nuestro prototipo con LoRaWAN, podemos dar por concluida la realización al completo de la primera fase del prototipo, perteneciente al estudio de viabilidad de redes inalámbricas en un parking subterráneo.

Posteriormente, lo ideal sería proceder con la realización de la segunda fase cómo se detalla en el WBS, pero por limitación de tiempo, falta de recursos y adaptación del prototipo, dejamos esta fase como una línea futura para la realización del prototipo inicial. El prototipo demostrador carece de instalación compleja, que es en lo que fundamentalmente se basa la fase dos, solamente es necesario conectar por cable USB los dos módulos LoRa a dos ordenadores independientes. Por lo tanto, no tiene sentido proponer esta “instalación” como segunda fase en nuestro prototipo final.

Si se quisiera desplegar el prototipo inicial, sí que sería necesaria la elaboración de esta segunda fase para la perfecta previsión del desarrollo del prototipo en un área subterránea. Después de la programación y verificación del funcionamiento de los dispositivos, entraría en escena la

instalación de dichos dispositivos dentro del entorno de trabajo, tal como se explica en la estructuración del WBS. Para realizar esta instalación sería necesario acordar plazos y tiempo de obra con el responsable de la edificación para la mejor organización, contratar mano de obra en caso de necesidad, valorar costes, instalar la red en toda la zona dentro de los plazos acordados y comprobar que la red funciona como se estima en la redacción del informe.

Como se puede observar, la realización de esta fase supondría un aliciente fuera de nuestro alcance en estos momentos, queda la mención y previsión de las bases fundamentales para una línea futura que en caso de querer actualizar el proyecto sería perfectamente recomendable seguirla. Con estas aclaraciones dejamos concluidas las fases de nuestra estructura WBS, recomendables para un proyecto de ingeniería, y damos por finalizado la explicación del apartado del prototipo con LoRaWAN.

4. Conclusiones y Líneas futuras

4.1. Conclusiones

Llegados al apartado final del trabajo, podemos afirmar que se han realizado con éxito los dos objetivos principales expuestos en la introducción del proyecto. El primer objetivo, basado en la búsqueda y estudio de las principales redes WSN enfocadas para trabajar en un parking, nos ha servido de base para poder afrontar y realizar el segundo objetivo, un prototipo final realizado a través del estándar WSN más adecuado según una comparación previa entre protocolos.

El estudio de viabilidad, encontrado en el punto 2 del trabajo, nos ha brindado un aprendizaje y unos conocimientos necesarios para poder proponer una solución al servicio demandado por la empresa. Sin estos cimientos básicos, fomentados en una exhaustiva investigación técnica, nos hubiéramos encontrado con muchas dificultades a la hora de proponer un prototipo adecuado. Por lo tanto, corroboramos que este objetivo ha sido el foco principal de este trabajo. Hemos conseguido sacar un diseño funcional a partir del estudio técnico y adquirido conocimientos valiosos sobre las redes WSN y sus entornos de trabajo.

El segundo objetivo, con menos peso dentro del trabajo, pero sí relevante, nos ha proporcionado un conocimiento más profundo sobre las redes LoRaWAN, a pesar de que ya tenía referencias sobre esta tecnología porque es un estándar con el que hemos trabajado en la carrera y muy conocido en el ámbito de las telecomunicaciones e IoT. He conseguido ampliar mis conocimientos utilizando dispositivos que incorporen esta tecnología, a partir de ellos he podido crear un prototipo demostrador que ejemplifique la idea de mi prototipo inicial, y además corrobore las especificaciones del estudio. Pienso que es una solución muy adecuada para espacios similares a un parking subterráneo, como por ejemplo metros, parkings de aeropuertos, de supermercados y un largo etcétera.

Cabe destacar que para realizar el prototipo he dispuesto de un mes, ya que el trabajo lo he focalizado en hacer un estudio sobre las redes WSN. Si dispusiera de más tiempo seguramente le podría haber añadido alguna prestación más elaborada. Por ejemplo, que la comunicación pudiera ser bidireccional en vez de que solo fuera desde el sensor de coche EV a la nube. Aun así, creo que el prototipo ya que cumple con las ideas principales del trabajo y con lo demandado por la empresa, ya que en un entorno similar a un parking la comunicación se establecería con éxito.

También, me gustaría concluir que la entrevista con el Sr. Carlos Sánchez fue la mejor puesta en escena que podía tener. Al ser un especialista experimentado en las redes móviles, a la hora de enfocar el trabajo me dio un punto de vista adicional al que yo tenía y aprendí con detalle cómo elabora una gran empresa de telecomunicaciones los despliegues de red inalámbrica en superficies similares a la que me propuso la empresa.

4.2. Líneas futuras

Toda primera versión de una infraestructura IoT siempre es mejorable, y el prototipo presentado en este trabajo no es una excepción. A continuación, se presentan nuevas ideas con tareas que se podrían desarrollar en un futuro cercano, orientadas al segundo objetivo del trabajo, el prototipo con LoRaWAN.

Para empezar, en caso de disponer de más tiempo se podría haber programado un prototipo demostrador con más prestaciones. Como ya he dicho en el anterior apartado, a pesar de que cumple con la idea principal expuesta al inicio del trabajo, creo que dotar de alguna funcionalidad más en el prototipo hubiera podido ampliar mis conocimientos con la tecnología LoRa y la IDE de Arduino.

En este aspecto se podría investigar más sobre los dispositivos LoRa, con objeto de ampliar la distancia de comunicación entre ellos, conocer otros entornos de programación a parte del de Arduino y adentrarse más en la tecnología LoRa, para así, poder crear un prototipo menos educativo y más orientado a la comercialización y desarrollo de redes WSN extensas.

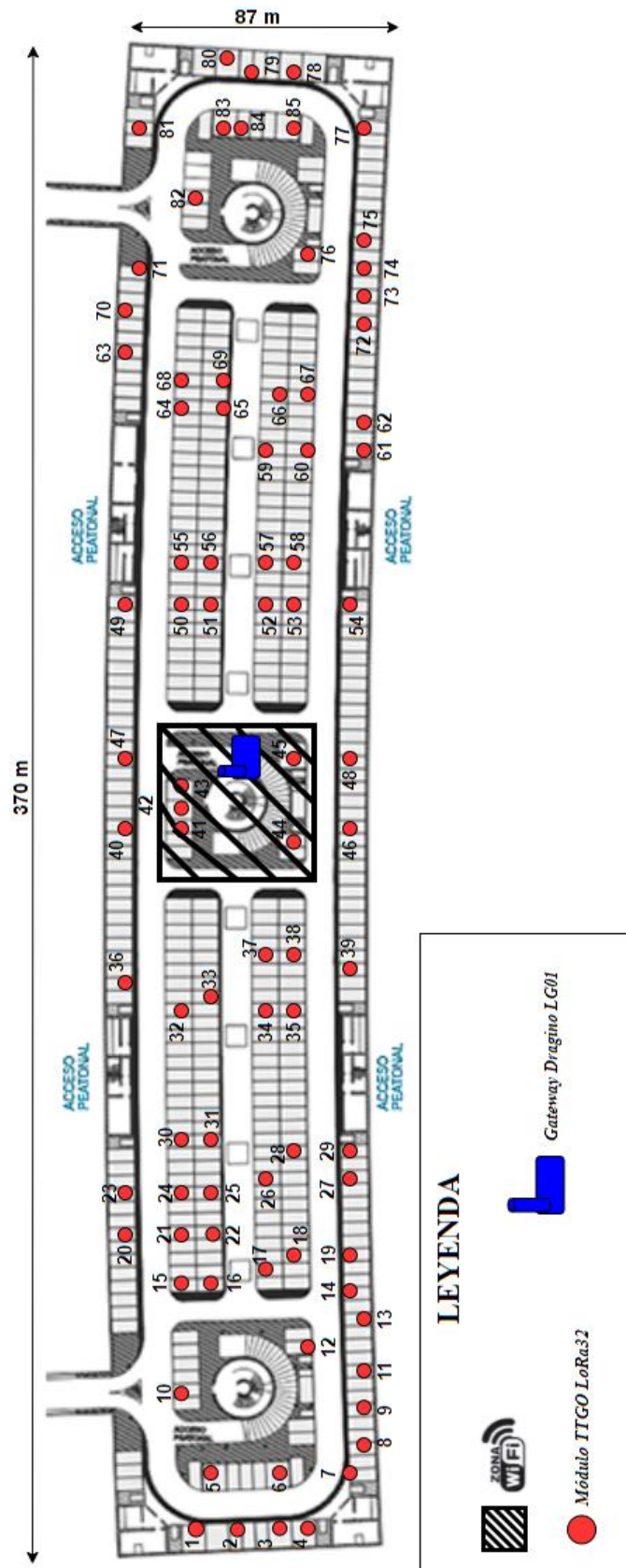
Por otra parte, la falta de acceso a los sensores de los cargadores de coches eléctricos y la base de datos del sistema de dichos cargadores nos ha limitado en la búsqueda de dispositivos LoRaWAN. Al no poder trabajar con estos sensores que tienen que ir conectados a los terminadores LoRa, ni poder acceder a la base de datos donde se almacena la información que debe comunicarse con la pasarela LoRa, no hemos podido tener la certeza de que los dispositivos expuestos en nuestro proyecto sean compatibles con los sensores y la nube. Creo que con un acceso a ellos y una buena investigación previa, se encontraría la tecnología LoRa adecuada para poder realizar la comunicación completa y escalar el prototipo demostrador.

Asimismo, con más presupuesto se podría haber desarrollado un prototipo demostrador con un *gateway* LoRa, en vez de utilizar tan solo dos *end devices*. También, podríamos haber añadido algunos dispositivos más a la red para así, tener una comunicación más íntegra, auténtica y variada. Un presupuesto más elevado nos hubiese brindado más posibilidades para poder ejemplificar nuestro objetivo en el trabajo final de grado.

Por lo tanto, con más tiempo, fondos y recursos, en caso de querer continuar con el desarrollo del prototipo, se nos ofrece una mejora del trabajo bastante evidente para poder realizar estos aspectos en un futuro y poder realizar la segunda parte de la estructura WBS, que queda mencionada pero no desarrollada por las limitaciones mencionadas a lo largo del apartado. Todos estos aspectos los dejamos mencionados como una línea futura para la mejora de este trabajo y no se descarta, en un futuro próximo, continuar con este prototipo para llevarlo a cabo como una solución comercial de IoT y que de valor a la comunidad de coches eléctricos.

5. Planos

5.1. Plano general de la instal·lació en la planta -1



6. Referencias y Webgrafía

- [1] Tekniker. “REDES DE SENSORES”. [Acceso: 22/02/21]. URL: <https://www.tekniker.es/es/redes-de-sensores>
- [2] Wikipedia. *Red ad hoc inalámbrica*. [Acceso: 22/02/21]. URL: https://es.wikipedia.org/wiki/Red_ad_hoc_inal%C3%A1mbrica
- [3] David Girbau; Ramón Villarino. “Tema 3: *Wireless Sensor Networks (WSN)*”. Teoría sobre las WSN de la asignatura Tecnologías para las Redes de Sensores, las IoT y las Smart Cities.
- [4] Priyanka Rawat; Kamal Deep Singh; Hakima Chaouchi; Jean Marie Bonnin. “Lectura 4: *Wireless sensor networks: a survey on recent developments and potential synergies*”. Teoría sobre las WSN de la asignatura Tecnologías para las Redes de Sensores, las IoT y las Smart Cities.
- [5] Cristian Campo Vos. “*Uso de una red inalámbrica de sensores para estimar riesgo de incendio*”. Proyecto fin de grado. Universidad Politécnica de Madrid.
- [6] Bulmaro Noguera. “*Topología de red: malla, estrella, árbol, bus y anillo*”. [Acceso: 26/02/21]. URL: <https://culturacion.com/topologia-de-red-malla-estrella-arbol-bus-y-anillo/>
- [7] Wikipedia. *Modos de operación de una unidad de cifrado por bloques*. [Acceso: 28/02/21]. URL: https://es.wikipedia.org/wiki/Modos_de_operaci%C3%B3n_de_unidad_de_cifrado_por_bloques
- [8] Wikipedia. *Red de sensores*. [Acceso: 03/03/21]. URL: https://es.wikipedia.org/wiki/Red_de_sensores
- [9] Toni Lázaro. “*Theme 4. Bluetooth technology*”. Teoría sobre la asignatura de Laboratorio de Sensores y Tecnologías Móviles.
- [10] “*CAPÍTULO 2: Bluetooth*”. [Acceso: 05/03/21]. URL: <http://bibing.us.es/proyectos/abreproy/11972/fichero/Cap%C3%ADtulo+2+--+Bluetooth.pdf>
- [11] Wikipedia. *Wifi*. [Acceso: 09/03/21]. URL: <https://es.wikipedia.org/wiki/Wifi#:~:text=Est%C3%AIndares%20que%20certifica%20la%20Alianza%20Wi%2DFi,Art%C3%ADculo%20principal%3A%20IEEE&text=Son%20los%20siguientes%3A,300%20Mbit%2Fs%2C%20respectivamente>
- [12] Xataka. “*Qué es Wi-Fi 6 y qué ventajas tiene con respecto a la versión anterior*”. [Acceso: 09/03/21]. URL: <https://www.xataka.com/basics/que-wi-fi-6-que-ventajas-tiene-respecto-a-version-anterior>
- [13] Adsl Zone. “*Qué es el WiFi y cómo funciona para conectar todo a Internet*”. [Acceso: 09/03/21]. URL: <https://www.adslzone.net/reportajes/tecnologia/que-es-wifi-como-funciona/>
- [14] Wikipedia. *IEEE 802.11ac*. [Acceso: 10/03/21]. URL: https://es.wikipedia.org/wiki/IEEE_802.11ac
- [15] Jordi Salazar. “*Redes inalámbricas*”. TechPedia. European Virtual Learning Platform for Electrical and Information Engineering. [Acceso: 11/03/21]. URL: https://upcommons.upc.edu/bitstream/handle/2117/100918/LM01_R_ES.pdf
- [16] Incibe. “*WiFi 6: características de seguridad y otras mejoras*”. [Acceso: 09/03/21]. URL: <https://www.incibe.es/protege-tu-empresa/blog/wifi-6-caracteristicas-seguridad-y-otras-mejoras>
- [17] “*Ventajas y desventajas de Zigbee*”. [Acceso: 14/03/21]. URL: <https://sites.google.com/site/ingsoftwares5a/ventajas-y-desventajas-de-zigbee>
- [18] Semtech SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver. [Acceso: 20/03/21]. URL: https://semtech.my.salesforce.com/sfc/p/E0000000JeIG/a/2R0000001OKs/Bs97dmPXeatnbdoJNVMIIDaKDIQz8q1N_gxDc_gqi7g2o
- [19] Marc Lázaro; Antonio Lázaro; Ramón Villarino. “*Feasibility of backscatter communication using LoRAWAN signals for deep implanted devices and wearable applications*”. Artículo. Universitat Rovira i Virgili.
- [20] LoRawan. “*¿Qué es LoRa y LoRaWAN?*”. [Acceso: 20/03/21]. URL: <https://lorawan.es/>
- [21] Wikipedia. *LoRaWAN*. [Acceso: 20/03/21]. URL: <https://es.wikipedia.org/wiki/LoRaWAN>
- [22] Medium. “*Haciendo IoT con LoRa: Capítulo 1. ¿Qué es LoRa y LoRaWAN?*” [Acceso: 22/03/21]. URL: <https://medium.com/beelan/haciendo-iot-con-lora-cap%C3%ADtulo-1-qu%C3%A9-es-lora-y-lorawan-8c08d44208e8>

- [23] Alberto Andreu Dólera. “*MODELADO DE REDES LORAWAN APLICADAS A LA CONSERVACIÓN PREVENTIVA DE PATRIMONIO CULTURAL*”. Proyecto fin de máster. Universidad Politécnica de Valencia. 2019. [Acceso: 22/03/21]. URL: <https://riunet.upv.es/bitstream/handle/10251/127816/Andreu%20-%20Modelado%20de%20redes%20LoRaWAN%20aplicadas%20a%20la%20conservaci%C3%B3n%20preventiva%20de%20patrimonio%20cultural.pdf?sequence=1>
- [24] PDAControl. “*Iniciando, consideraciones y conceptos LoRaWAN #1*”. [Acceso: 23/03/21]. URL: <http://pdacontroles.com/iniciando-consideraciones-y-conceptos-lorawan-1/>
- [25] Tecnología Humanizada. “*Seguridad en redes LoRaWAN*”. [Acceso: 23/03/21]. URL: <https://humanizationoftechnology.com/seguridad-en-redes-lorawan/revista/iot/01/2019/>
- [26] Panama Hitek. “*¿Qué es LoRaWAN y para qué se utiliza?*”. [Acceso: 22/03/21]. URL: <http://panamahitek.com/que-es-lorawan-y-para-que-se-utiliza/>
- [27] Aprendiendo Arduino. “*Estructura sketch en Arduino*”. [Acceso: 01/04/21]. URL: <https://aprendiendoarduino.wordpress.com/2016/06/26/estructura-sketch-en-arduino/>
- [28] Aprendiendo Arduino. “*Entornos de Aplicación Arduino*”. [Acceso: 09/03/19]. URL: <https://aprendiendoarduino.wordpress.com/2016/06/26/entornos-de-aplicacion-arduino/>
- [29] Arduino en Español. “*Wiki Comunitaria*”. [Acceso: 22/04/21]. URL: <https://playground.arduino.cc/Es/Es>
- [30] Aprendiendo Arduino. “*MQTT*”. [Acceso: 24/04/21]. URL: <https://aprendiendoarduino.wordpress.com/tag/hivemq/>
- [31] Hive MQ. “*HiveMQ MQTT Broker*”. [Acceso: 24/04/21]. URL: <https://www.hivemq.com/hivemq/mqtt-broker/>
- [32] Aprendiendo Arduino. “*Arduino y LoRaWAN*”. [Acceso: 12/05/21]. URL: <https://www.aprendiendoarduino.com/tag/lora-gateway/>
- [33] PCTCAN. “*Parking subterráneo PCTCAN*”. [Acceso: 14/05/21]. URL: <https://www.pctcan.es/mapa-del-parking-subterraneo/>
- [34] AliExpress. “*TTGO 2 unids/lotte ESP32 SX1276 LoRa, 868 / 915MHz Bluetooth Wi-Fi Lora Internet antena Placa de desarrollo para Arduino*”. [Acceso: 01/05/21]. URL: https://es.aliexpress.com/item/4000059708434.html?spm=a2g0o.productlist.0.0.684d12a1FZzTm6&algo_pvid=5a0a78a5-5084-4c75-bc8c-439340e9f305&algo_expid=5a0a78a5-5084-4c75-bc8c-439340e9f305-5&btsid=2100bddb16182328936242583e60a1&ws_ab_test=searchweb0_0.searchweb201602_.searchweb201603
- [35] Robot Shop. “*LG01-P Puerta de Enlace LoRa 868 MHz (Europa)*”. [Acceso: 01/05/21]. URL: https://www.robotshop.com/es/es/lg01-p-puerta-enlace-lora-868-mhz-europa.html?gclid=Cj0KCOjw38-DBhDpARIsADJ3kjnev8h5gEUCQRM4ZdHSY8QCGAR6qMcx0Q9eM9TBq1mmfwTmw3ZEI98aAi4kEALw_wcB

7. Anexos

7.1. Código *end device* del envío de paquetes LoRa

```

#include <SPI.h>
#include <LoRa.h>
//define the pins used by the transceiver module
//define the pins used by the LoRa transceiver module
#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 14
#define DIO0 26

int counter = 0;

void setup() {
  //initialize Serial Monitor
  Serial.begin(115200);
  while (!Serial);
  Serial.println("LoRa End device");

  //setup LoRa transceiver module
  SPI.begin(SCK, MISO, MOSI, SS);
  //setup LoRa transceiver module
  LoRa.setPins(SS, RST, DIO0);

  //replace the LoRa.begin(---E-) argument with your location's frequency
  //433E6 for Asia
  //866E6 for Europe
  //915E6 for North America
  while (!LoRa.begin(866E6)) {
    Serial.println(".");
    delay(500);
  }
  // Change sync word (0xF3) to match the receiver
  // The sync word assures you don't get LoRa messages from other LoRa
transceivers
  // ranges from 0-0xFF
  LoRa.setSyncWord(0xF3);
  Serial.println("LoRa Initializing OK!");
  Serial.println(" ");
}

void loop() {
  //Begin LoRa packet to receiver
  LoRa.beginPacket();
  String cont = String(counter);
  String packet = String("Prueba de potencia " + cont);
  LoRa.print(packet);
  Serial.println(packet);
  Serial.println(" ");
}

```

```

LoRa.endPacket();
counter++;
delay(10000);
}

```

7.2. Código gateway de recepción LoRa junto a su potencia recibida

```

#include <SPI.h>
#include <LoRa.h>
//define the pins used by the transceiver module
#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 14
#define DIO0 26

void setup() {
  Serial.begin(115200);
  while (!Serial);
  Serial.println("LoRa Gateway");

  //SPI LoRa pins
  SPI.begin(SCK, MISO, MOSI, SS);
  //setup LoRa transceiver module
  LoRa.setPins(SS, RST, DIO0);

  //replace the LoRa.begin(---E-) argument with your location's frequency
  //433E6 for Asia
  //866E6 for Europe
  //915E6 for North America
  while (!LoRa.begin(866E6)) {
    Serial.println(".");
    delay(500);
  }
  // Change sync word (0xF3) to match the receiver
  // The sync word assures you don't get LoRa messages from other LoRa
transceivers
  // ranges from 0-0xFF
  LoRa.setSyncWord(0xF3);
  Serial.println("LoRa Initializing OK!");
  Serial.println(" ");
}

void loop() {
  // try to parse packet
  String LoRaData = "";
  int packetSize = LoRa.parsePacket();
  while (!packetSize) {
    packetSize = LoRa.parsePacket();
    //Serial.print( packetSize);
    if (packetSize) {
      // received a packet
      Serial.println("Paquete recibido: ");
    }
  }
}

```

```

    // read packet
    while (LoRa.available()) {
        LoRaData = LoRa.readString();
        Serial.print(LoRaData);
    }
    // print RSSI of packet
    Serial.println(" ");
    Serial.println("Potencia recibida por el LoRa End device: ");
    Serial.print(LoRa.packetRssi());
    Serial.print(" dBm");
    Serial.println(" ");
    Serial.println(" ");
}
}
}

```

7.3. Creación y envío de la información de los sensores para cargadores de EV

```

#include <SPI.h>
#include <LoRa.h>
//#include <arduino-LoRa-master.h>

//define the pins used by the transceiver module
#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 14
#define DIO0 26

int counter = 0;

void setup() {
    //initialize Serial Monitor
    Serial.begin(115200);
    while (!Serial);
    Serial.println("LoRa End device");

    //setup LoRa transceiver module
    SPI.begin(SCK, MISO, MOSI, SS);
    //setup LoRa transceiver module
    LoRa.setPins(SS, RST, DIO0);

    //replace the LoRa.begin(---E-) argument with your location's frequency
    //433E6 for Asia
    //866E6 for Europe
    //915E6 for North America
    while (!LoRa.begin(866E6)) {
        Serial.println(".");
        delay(500);
    }
    // Change sync word (0xF3) to match the receiver

```

```
// The sync word assures you don't get LoRa messages from other LoRa
transceivers
// ranges from 0-0xFF
LoRa.setSyncWord(0xF3);
Serial.println("LoRa Initializing OK!");
Serial.println(" ");
}

void loop() {
  //Begin LoRa packet to receiver
  LoRa.beginPacket();
  String LoRapacket, nums, letters, connection;
  long randomNumber = random(10000);
  nums = String(randomNumber);
  if (0 <= randomNumber && randomNumber < 10) {
    nums = String("000" + nums);
  }
  if (10 <= randomNumber && randomNumber < 100) {
    nums = String("00" + nums);
  }
  if (100 <= randomNumber && randomNumber < 1000) {
    nums = String("0" + nums);
  }

  randomNumber = random(0, 7);
  switch (randomNumber) {

    case 0:
      letters = String("YBC");
      break;

    case 1:
      letters = String("FGH");
      break;

    case 2:
      letters = String("VBR");
      break;

    case 3:
      letters = String("PKL");
      break;

    case 4:
      letters = String("WRN");
      break;

    case 5:
      letters = String("QWZ");
      break;

    case 6:
      letters = String("XML");
```

```

    break;
}

randomNumber = random(0, 3);
switch (randomNumber) {

    case 0: //CONNECTED
        connection = String("connected");
        break;

    case 1: //DISCONNECTED
        connection = String("disconnected");
        break;

    case 2: //CHARGED
        connection = String("charged");
        break;
}

LoRapacket = String(nums + letters + connection);
LoRa.print(LoRapacket);
Serial.println("Matrícula recibida: " + nums + letters);
Serial.println("Estado del EV recibido: " + connection);
Serial.println("");
Serial.println("Mensaje enviado por LoRa: " + LoRapacket);
Serial.println("");
LoRa.endPacket();
delay(10000);
}

```

7.4. Recepción LoRa al gateway y subida de la información a HiveMQ

```

// MQTT
#include <WiFi.h>
#include <PubSubClient.h>

const char* ssid = "SercommFB50";
const char* password = "XXXXXXXXXXXX";
const char* mqttServer = "broker.hivemq.com"; // broker IP
const int mqttPort = 1883; // Puerto TCP de hiveMQ. Mensaje MQTT no encriptado
const char* mqttUser = "";
const char* mqttPassword = "";

WiFiClient espClient;
PubSubClient client(espClient);

//Rutina de gestion de datos cuando recibe un topico
void callback(char* topic, byte* payload, unsigned int length) {
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
    }
}

```

```

    }
    Serial.println();
}

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.println("Attempting MQTT connection...");
    // Attempt to connect
    if (client.connect("ESP32Client", mqttUser, mqttPassword)) {
      Serial.println("connected");
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}
}
// MQTT

#include <SPI.h>
#include <LoRa.h>
//#include <arduino-LoRa-master.h>

//define the pins used by the transceiver module
//define the pins used by the LoRa transceiver module
#define SCK 5
#define MISO 19
#define MOSI 27
#define SS 18
#define RST 14
#define DIO0 26

void setup() {
  // MQTT
  //Fixo rutina gestio dades rebudes
  client.setCallback(callback);
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.println("Connecting to WiFi..");
  }
  Serial.println("Connected to the WiFi network");
  client.setServer(mqttServer, mqttPort);
  delay(1500);
  // MQTT

  //initialize Serial Monitor
  while (!Serial);
  Serial.println("LoRa Gateway");
}

```

```

//SPI LoRa pins
SPI.begin(SCK, MISO, MOSI, SS);
//setup LoRa transceiver module
LoRa.setPins(SS, RST, DIO0);

//replace the LoRa.begin(---E-) argument with your location's frequency
//433E6 for Asia
//866E6 for Europe
//915E6 for North America
while (!LoRa.begin(866E6)) {
  Serial.println(".");
  delay(500);
}
// Change sync word (0xF3) to match the receiver
// The sync word assures you don't get LoRa messages from other LoRa
transceivers
// ranges from 0-0xFF
LoRa.setSyncWord(0xF3);
Serial.println("LoRa Initializing OK!");
Serial.println(" ");
}

void loop() {
  // try to parse packet
  String LoRaData = "";
  int packetSize = LoRa.parsePacket();
  while (!packetSize) {
    packetSize = LoRa.parsePacket();
    //Serial.print( packetSize);
    if (packetSize) {
      // received a packet
      Serial.println("Paquete recibido: ");

      // read packet
      while (LoRa.available()) {
        LoRaData = LoRa.readString();
        Serial.print(LoRaData);
      }
      // print RSSI of packet
      Serial.println(" ");
      Serial.println("Potencia recibida por el LoRa End device: ");
      Serial.print(LoRa.packetRssi());
      Serial.print(" dBm");
      Serial.println(" ");
      Serial.println(" ");
    }
  }

  //MQTT
  if (!client.connected()) {
    reconnect();
  }
}

```

```
/// Subida a HiveMQ mediante el topico ControlEV
int i = LoRaData.length() + 1; // one extra character for the null
char aux[i]; // buffer
LoRaData.toCharArray(aux, i); //conversion string to chararray
client.publish("ControlEV", aux); //envio el dato al topic en formato
chararray
Serial.println("");
Serial.println("Paquete subido a HiveMQ: ");
Serial.println(LoRaData);
Serial.println("-----");
client.loop();
//MQTT
}
```

7.5. Diagrama Gantt de la fase 1

