

# IMPLEMENTACIÓN DE UN SISTEMA INALÁMBRICO DE BAJO COSTE PARA APLICACIONES IoT

AUTOR: CRISTIAN ANTÓN RODRÍGUEZ

TUTOR: NARCÍS CARDONA MARCET

DIRECTOR: GERARDO MARTÍNEZ PINZÓN

Trabajo Fin de Grado presentado en la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universitat Politècnica de València, para la obtención del Título de Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación  
Curso 2017-18

Valencia, 10 de septiembre de 2018



## Agradecimientos

A mis padres por darme todo su apoyo siempre y de manera incondicional.

A mi hermana, por ayudarme en todo lo que siempre he necesitado.

A mi pareja, por estar ahí siempre y haberme apoyado en todo momento.

A mis tíos, por sus consejos, cariño y afecto.

A mi tutor de proyecto, Narcís Cardona, por ser tan amable y haberme ofrecido esta oportunidad.

A mi director de proyecto en el iTEAM, Gerardo Martínez, por su tiempo y dedicación.

A mis amigos, que tanto me han ayudado. Especialmente a Jose Mor.

A mi amigo y compañero de trabajo, Saul Fuster, por su apoyo y disponibilidad.

A todos aquellos profesores que me han ayudado a lo largo de mi carrera a ser quien soy.

A todos ellos, muchas gracias.

## Resumen

En este Trabajo Fin de Grado, se ha desarrollado una solución de bajo coste para Internet de las Cosas, haciendo uso de la tecnología LoRa y del hardware Arduino. El sistema se ha desarrollado como una solución genérica para cualquier tipo de aplicación, cuyas capacidades sean, la captación de los datos a través de sensores, la transmisión de los mismos a un nodo central, el análisis de esos datos en una central y el almacenamiento de toda esa información dentro de una base de datos para que el usuario final pueda acceder a ellos fácilmente. En cuanto al desarrollo, este sistema ha sido validado en un laboratorio y posteriormente desplegado en distintas zonas. En concreto, se realizaron medidas de campo en un entorno urbano (*indoor to outdoor*) y en un entorno rural (*outdoor to outdoor*), donde se pudo comprobar la robustez, fiabilidad y alcance del sistema.

## Resum

A aquest Treball de Fi de Grau, s'ha desenvolupat una sol·lució de baix presupost per a l'Internet de les Coses, util·litzant la tecnologia LoRa i el programari d'Arduino. El sistema s'ha desenvolupat amb una sol·lució genèrica per a qualsevol tipus d'aplicació, de les que les seues capacitats siguen, la captació de les dades per mitjà del sensors, la transmissió de les mateixes a un node central, l'anàlisi d'aquestes dades a una central i l'emmagatzematge dins d'una base de dades per a que l'usuari final pugua accedir a elles fàcilment. Segons el desenvolupament, aquest sistema ha sigut validat a un laboratori i posteriorment desplegar a distints zones. Concretament, es varen realitzar de camp a un entorn urbà (*indoor to outdoor*) i a un entorn rural (*outdoor to outdoor*) on se va poder comprovar la robustesa, fiabilitat i abast del sistema.

## Abstract

In this Final Degree Project, it has been developed a low cost solution for Internet of Things, by using LoRa technology and Arduino hardware. The system has been developed as a generic solution for any type of application, whose skills are, the data sensing through sensors, the transmission of those to a central node, the analysis of that data in a central and the storage of all that information within a database with the purpose that the final user could take it easily. About the development, the system has been validated in a laboratory and then deployed in different zones. Specifically, measurements have been done in an urban environment (*indoor to outdoor*) and in a rural environment (*outdoor to outdoor*) where could be checked the sturdiness, reliability and system range.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Objetivos	6
1.2. Estructura de la memoria	7
<b>2. Estado del arte</b>	<b>8</b>
2.1. Historia y evolución	8
2.2. Capacidades de un sistema IoT	9
2.3. Arquitectura	11
2.3.1. Arquitectura de cinco capas	12
2.4. Tecnología LoRa	17
2.4.1. Clases de dispositivos	18
2.4.2. Estructura de trama	19
2.4.3. Parámetros de transmisión	20
2.5. Hardware IoT	21
2.5.1. Arduino	22
<b>3. Diseño de un sistema IoT</b>	<b>24</b>
3.1. Arquitectura del sistema	24
3.2. Sistema del nodo cliente	25
3.2.1. Modo de operación general	25

3.3.	Sistema del nodo central . . . . .	28
3.3.1.	Modo de operación general . . . . .	28
3.4.	Sistema administrador . . . . .	30
3.4.1.	Modo de operación . . . . .	31
<b>4.</b>	<b>Validación del sistema y análisis de resultados</b>	<b>32</b>
4.1.	Captura de datos de los sensores . . . . .	32
4.1.1.	Almacenamiento de la información . . . . .	34
4.2.	Cobertura del sistema en un entorno real . . . . .	35
4.2.1.	Medidas <i>indoor - outdoor</i> (I2O) en Campus UPV - Entorno urbano . . . . .	35
4.2.2.	Medidas <i>outdoor - outdoor</i> (O2O) en Huerta de Valencia - Entorno urbano . . . . .	39
4.3.	Comparación con la plataforma de Libelium . . . . .	42
4.3.1.	Estudio de cobertura . . . . .	43
4.3.2.	Estudio económico . . . . .	44
<b>5.</b>	<b>Conclusiones y líneas futuras de trabajo</b>	<b>45</b>
5.1.	Conclusiones . . . . .	45
5.2.	Líneas de trabajo futuras . . . . .	47
<b>A.</b>	<b>Instrumentación nodo cliente</b>	<b>50</b>
<b>B.</b>	<b>Instrumentación nodo central</b>	<b>53</b>
<b>C.</b>	<b>Parametrización inicial</b>	<b>55</b>
C.1.	Canales y potencia de transmisión . . . . .	55
C.2.	Modos de transmisión . . . . .	56
C.3.	Tablas de encaminamiento e ID de los paquetes . . . . .	60
<b>D.</b>	<b>Estimación del canal radio</b>	<b>62</b>

# Índice de figuras

1.1. Representación técnica de una red IoT . . . . .	2
1.2. Tipos de compañías interesadas en IoT . . . . .	4
2.1. Representación de posibles escenarios IoT . . . . .	9
2.2. Bloques sistema IoT . . . . .	9
2.3. Tipos de arquitecturas IoT . . . . .	11
2.4. Arquitectura IoT de cinco capas . . . . .	12
2.5. Clases de dispositivos de tecnología LoRa . . . . .	19
2.6. Estructura física de un mensaje de <i>uplink</i> . . . . .	20
2.7. Estructura física de un mensaje de <i>downlink</i> . . . . .	20
2.8. Modulación <i>Chirp Spread Spectrum</i> en LoRa . . . . .	21
2.9. Hardware Arduino UNO . . . . .	22
3.1. Arquitectura de un sistema inalámbrico de bajo coste para aplicaciones de IoT . . . . .	24
3.2. Configuración de los elementos para el nodo cliente . . . . .	25
3.3. Diagrama de flujo general del cliente . . . . .	26
3.4. Configuración de los elementos para el nodo central . . . . .	28
3.5. Diagrama de flujo general del servidor . . . . .	29
3.6. Administrador central del sistema . . . . .	30
4.1. Dispositivo del nodo cliente . . . . .	33
4.2. Gráfica de medidas de los sensores . . . . .	33

4.3. Base de datos para la información de temperatura . . . . .	34
4.4. Mapa de medidas en Campus UPV, caso A . . . . .	36
4.5. Gráfica de las medidas captadas usando el modo por defecto . . . . .	36
4.6. Mapa de medidas en Campus UPV, caso B . . . . .	38
4.7. Gráfica de las medidas captadas usando el modo del Caso B . . . . .	38
4.8. Mapa de medidas entorno rural . . . . .	40
4.9. Unidad móvil para las mediciones en un entorno real . . . . .	40
4.10. Gráfica de las medidas captadas en el entorno rural . . . . .	41
4.11. Mapa de medidas en Campus UPV, sistema Libelium . . . . .	43
A.1. Arduino MEGA 2560 . . . . .	50
A.2. LoRa Shield - 868 MHz . . . . .	51
A.3. Sensor de temperatura . . . . .	51
A.4. Sensor de humedad . . . . .	52
A.5. Sistema completo del nodo cliente . . . . .	52
B.1. Arduino UNO . . . . .	53
B.2. LoRa Shield - 868 MHz . . . . .	54
B.3. Sistema completo del nodo cliente . . . . .	54
C.1. Interfaz LoRa Calculator . . . . .	56
D.1. Función estimación del canal . . . . .	63

# Índice de cuadros

2.1. Tecnologías inalámbricas disponibles en entornos personales, locales y amplios . . . . .	15
2.2. Plataformas en la nube de IoT y sus características . . . . .	16
2.3. Tecnologías disponibles para un sistema IoT . . . . .	17
2.4. Tipos de Arduino y sus características . . . . .	23
4.1. Resultados de las medidas para el modo por defecto . . . . .	37
4.2. Resultado de las medidas para el modo Caso B . . . . .	39
4.3. Resultado de las medidas de campo en entorno rural . . . . .	42
4.4. Estudio económico comparativo de dos sistemas IoT . . . . .	44
C.1. Frecuencias centrales de transmisión para la tecnología LoRa en la banda de 868 MHz	55
C.2. Parámetros para configuración de los modos . . . . .	57
C.3. Especificaciones para el registro de configuración 0x1D . . . . .	58
C.4. Ejemplos programados para el registro 0x1D . . . . .	59
C.5. Especificaciones para el registro de configuración 0x1E . . . . .	59
C.6. Ejemplos programados para el registro 0x1D . . . . .	59
C.7. Especificaciones para el registro de configuración 0x26 . . . . .	60
C.8. Ejemplos programados para el registro 0x1D . . . . .	60
C.9. Registros de los modos de transmisión . . . . .	60
C.10.ID de los paquetes recibidos . . . . .	61



# Capítulo 1

## Introducción

Internet de las cosas (*IoT*, *Internet of Things*, en inglés), es uno de los temas más relevantes de investigación en los últimos años tanto para organismos de estandarización como para la industria tecnológica. Asimismo, estudios indican que su revolución comercial será a partir del año 2020, con alrededor de 25 mil millones de dispositivos conectados a Internet. En efecto, IoT es un concepto que se refiere a la interconexión digital de objetos cotidianos a Internet, donde cualquier objeto tanto físico como virtual puede conectarse a otros objetos y/o a Internet, creando una estructura de conexiones entre objetos o entre objetos y humanos. El Clúster Europeo de Investigación en IoT (*IERC*, *European Research Cluster on the Internet of Things*, en inglés) define IoT como: “una red global de infraestructura dinámica con capacidades de autoconfiguración, basada en estándares y protocolos de comunicación interoperables, donde las ‘cosas’ físicas y virtuales tienen identidad, atributos físicos y personalidad virtual; y que usa interfaces inteligentes y perfectamente integradas en la red de información” [1]. Una representación técnica de una red IoT se muestra en la figura 1.1

Como se puede observar en la figura 1.1, una red IoT está conformada por diferentes componentes, como dispositivos, pasarelas (*Gateway*, en inglés), objetos físicos y objetos virtuales [2]. Un objeto físico puede ser representado en el mundo de la información mediante uno o más objetos virtuales, sin embargo, un objeto virtual puede existir sin la necesidad de tener ningún objeto físico asociado. Un dispositivo es una pieza de equipo con capacidades obligatorias de comunicación y capacidades opcionales de detección, accionamiento, adquisición, almacenamiento y procesamiento de datos. Los dispositivos recaban diversos tipos de información y la suministran a las redes de información y comunicación para su posterior procesamiento. En concreto, los dispositivos se comunican con otros dispositivos principalmente a través de cuatro formas: (1) usando la red de comunicaciones por medio de un Gateway, (2) por medio de la red sin necesidad de pasar por el Gateway, (3) los dispositivos se comunican directamente entre ellos y (4) como una combinación de las opciones anteriores. Por último, las redes de comunicaciones transfieren datos adquiridos por los dispositivos a aplicaciones y otros dispositivos, así como instrucciones de las aplicaciones a los dispositivos. Las redes de comunicación ofrecen capacidades

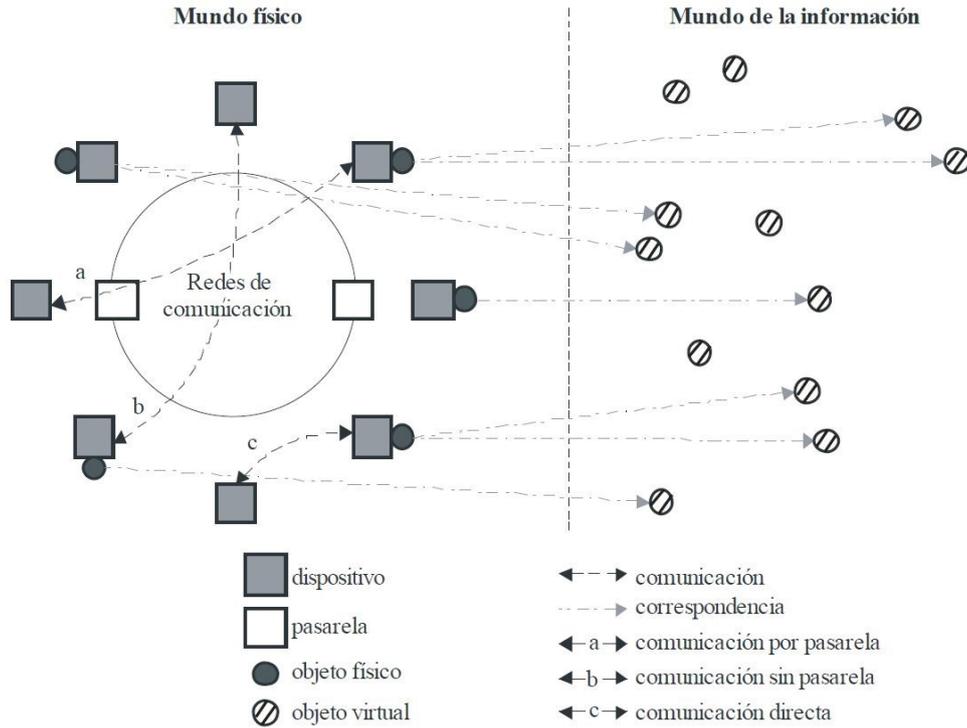


Figura 1.1: Representación técnica de una red IoT

de transferencia de datos fiables y eficientes. La infraestructura de una red IoT puede crearse mediante redes existentes, como las redes convencionales basadas TCP/IP, y/o redes evolutivas, tales como las redes de la próxima generación [2].

Una vez descrita la representación técnica de una red, es importante mencionar que IoT comprenderá diversidad de aplicaciones y servicios, tales como: salud inteligente, logística, ciudades inteligentes, agricultura y ganadería inteligentes, entre otras. De la misma forma, IoT generará una gran cantidad de datos de información derivados de estas aplicaciones, que mediante una gestión eficiente y segura permitirá crear nuevos modelos de mercado tanto horizontales como verticales. Asimismo, se están proponiendo diferentes soluciones tecnológicas inalámbricas que podrían soportar estas aplicaciones y servicios. Por un lado, se han propuesto tecnologías de campos específicos, tales como, Wi-Fi (*Wireless Fidelity*, en inglés), Bluetooth, ZigBee y LpWA (*Low Power Wide Area Networks*, en inglés). Por otro lado, el 3GPP (*3rd Generation Partnership Project*, en inglés) estandarizó tres tecnologías de banda estrecha que podrían hacer frente a las demandas de los servicios de IoT, Narrow Band for Internet of Things (NB-IoT), Long Term Evolution Machine Type Communications Category M1 (eMTC) y Extended Coverage GSM IoT (EC-GSM-IoT). Pero independientemente del tipo de solución tecnológica, se deberán cumplir los siguientes requerimientos para IoT: eficiencia energética, redes densas que puedan dar soporte a un número masivo de dispositivos, interoperabilidad, heterogeneidad, escalabilidad, fiabilidad, movilidad, seguridad... [3]. Respecto a las capacidades que deberá cumplir toda red IoT podemos citar [4]:

- Identificación del producto a través de una nomenclatura o identificación única a los objetos. Para IoT la identificación se realiza mediante EPC (*Electronic Product Code*, en inglés) o uCode system.
- Direccionamiento para la interconexión de redes mediante IPv4 o IPv6 (*Internet Protocol versión 4 or 6*, en inglés).
- Captación de datos mediante sensores inteligentes o detección de presencia tanto animada como inanimada a través de tecnologías como RFID (*Radio Frequency IDentification*, en inglés).
- Comunicación de los datos captados mediante tecnologías inalámbricas, como las mencionadas en el párrafo anterior. La función de dichas tecnologías es transmitir los datos captados desde los sensores hasta la parte de procesado, de forma rápida y segura.
- Computación a nivel de hardware o software. A nivel de hardware la computación se realiza mediante placas de desarrollo de bajo coste, las cuales son también conocidas como computadores de placa reducida. En la actualidad, existe una diversidad de mercado principalmente liderado por compañías como Arduino, Raspberry Pi, Intel Galileo, Gadgeteer. A nivel de software los datos captados se procesan y analizan de forma más exhaustiva para una gestión más eficiente de estos. Este procesado se lleva a cabo mediante sistemas operativos como: Contiki, TinyOS, LiteOS, Riot OS... o mediante herramientas disponibles en el Cloud (Nimbits, Hadoop...).
- Traducción de datos obtenidos en la etapa anterior a lenguaje de aplicación para servicios demandados por el usuario final. Para ello, se pueden utilizar modelos estándares para el intercambio de datos en la web, tales como, RDL (*Resource Description Framework*, en inglés), OWL (*Ontology Web Language*, en inglés), etc.

Todas esas capacidades que debe cumplir un sistema son articuladas dentro de las arquitecturas propuestas para IoT. Principalmente existen tres tipos de arquitecturas: las de 3, 4 y 5 capas; cada una de ellas útil para un determinado tipo de función, aunque actualmente la más utilizada, por ser la más polivalente y flexible, es la arquitectura de 5 capas [2][4]. Esta arquitectura está compuesta por la capa de percepción, la capa de red, la capa de procesado, la capa de aplicación y la capa de negocio. La capa de percepción representa los sensores físicos IoT, cuya función principal es medir y recopilar información, la cual es enviada por la capa de red a la capa de procesado. Esta capa es conocida también como el middleware del sistema IoT, cuya función es almacenar, analizar y procesar grandes cantidades de información. La siguiente capa es la de aplicación que es la responsable de proporcionar los servicios específicos requeridos por los usuarios finales. Por último, la capa de negocio es la encargada de determinar las acciones futuras y estrategias de negocio sobre el sistema IoT.

El ecosistema de IoT generarán un gran volumen de mercado que se puede clasificar en mercados horizontal y vertical. Por un lado, el mercado horizontal está conformado por la computación ubicua

y las funcionalidades analíticas pertenecientes a todos aquellos servicios que son independientes del dominio de aplicación. En este tipo de mercado se proporciona tanto bienes como servicios comunes a toda un área de negocio. Aquí tendrían cabida los servicios MROs (Mantenimiento, Reparaciones y Operaciones) u otros suministros indirectos que no forman parte de los procesos productivos. Por otro lado, el vertical es un tipo de mercado que engloba las soluciones específicas que los fabricantes aportan al sector de IoT. Dentro de este mercado, podemos citar soluciones de productos hardware, software, aplicaciones finales de usuario o servicios. Los 5 principales mercados verticales que están teniendo un importante crecimiento en el último año son: Smart Home, eHealth, Sport, Connected Car and Smart Cities [5]. La figura 1.2 muestra como está integrado el mercado IoT.

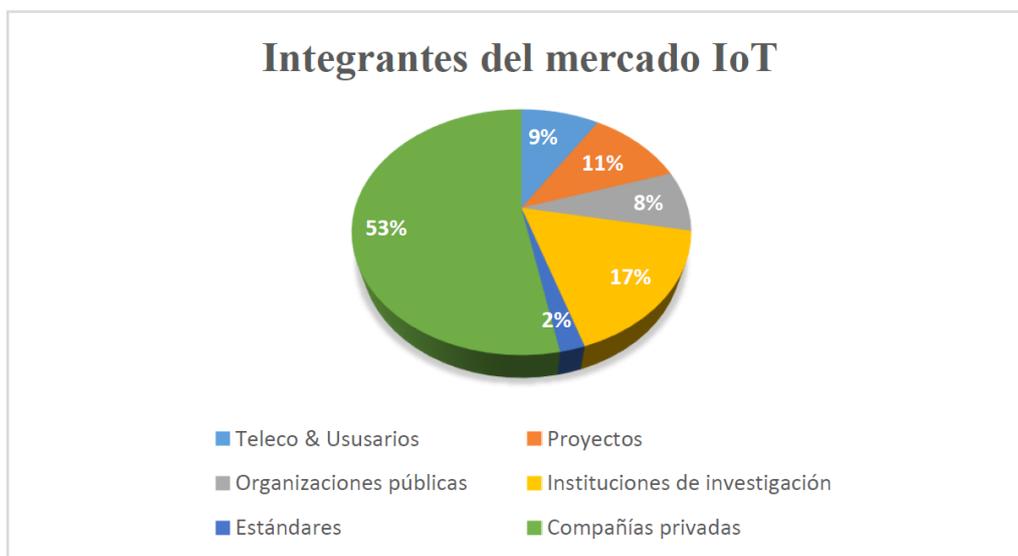


Figura 1.2: Tipos de compañías interesadas en IoT

Como se muestra en la figura anterior, hay principalmente seis integrantes del mercado IoT, obteniendo la mayor y menor participación las compañías privadas con un 53 % y las organizaciones de estandarización con un 2 %, respectivamente [6]. Respecto al mayor integrante del mercado IoT, se está presentando una segmentación del mismo, teniendo en cuenta que cada fabricante y/o compañía están diseñando y ofreciendo sus propias plataformas IoT, y, por ende, una solución tecnológica diferente en función de la aplicación o requerimientos de los usuarios finales. El principal interés de estas grandes compañías es tener una semántica común sólo en el nivel middleware<sup>1</sup> para poder soportar diferentes tecnologías de código abierto como FIWARE o SOFIA2 que sean capaces de hacer interoperables unas capas con otras. En concreto, por una parte, FIWARE es una plataforma promovida e impulsada por la Unión Europea para el despliegue y desarrollo de un Internet del Futuro totalmente compatible a

<sup>1</sup>Internet of Things **middleware** (también denominado “plumbing”) es un software que sirve como interfaz entre las capas de red y de aplicación de IoT, cuya función es hacer posibles las comunicaciones entre los elementos incapaces de comunicarse por sí solos. Básicamente, funciona como una capa de traducción oculta para permitir la comunicación y la administración de datos en aplicaciones distribuidas.

todos los niveles. Por otra parte, la plataforma Sofia2 se trata de una iniciativa llevada a cabo por la empresa española Indra, que permite la interoperabilidad de múltiples sistemas y dispositivos, ofreciendo una plataforma semántica que permita poner información del mundo real a disposición de las aplicaciones inteligentes de IoT, es decir, se trata de un middleware multilenguaje, multiprotocolo, multiplataforma y multidispositivo [6]. Sin embargo, las iniciativas anteriores no son lo suficientemente potentes como para llevar a cabo una línea de mejora global en el concepto de interoperabilidad. Por ese motivo la Unión Europea impulsó INTER-IoT, un proyecto que da origen a la implementación y la experimentación de una red de trabajo abierta que permita la interoperabilidad a distintos niveles, una metodología asociada y herramientas para permitir la interoperabilidad voluntaria entre plataformas heterogéneas de Internet of Things [6]. El enfoque INTER-IoT se puede aplicar a cualquier dominio de aplicación por lo que se puede dividir en tres grandes bloques:

- INTER-LAYER, que conforma los métodos y herramientas para proporcionar interoperabilidad entre cada capa de las plataformas IoT.
- INTER-FW (*interoperability framework*), que hace referencia a una estructura/arquitectura global.
- INTER-METH (*interoperability methodology*), que se refiere a la metodología basada en las herramientas CASE (acrónimo de *Computer Aided Software Engineering*) que son las diversas aplicaciones informáticas o programas informáticos destinados a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Este tipo de herramientas ayudarán a automatizar cada fase del proceso de integración, proporcionando pautas de trabajo, facilidades gráficas modelos ingenieriles y repositorios de datos.

Respecto a los órganos de estandarización de IoT, existe principalmente dos organizaciones: ETSI (European Telecommunications Standards Institute) y AIOTI (Alliance for Internet of Things Innovation). Por una parte, la ETSI desarrolló los siguientes estándares que comprenden los sectores de Smart Grids y Smart Metering: “SmartM2M; Smart Appliances; Reference Ontology and oneM2M Mapping”, “Open Smart Grid Protocol (OSGP); Smart Metering/Smart Grid Communication Protocol” [7][8]. Por otra parte, AIOTI, aunque no es propiamente un organismo de estandarización ha desarrollado importantes recomendaciones que involucran diferentes sectores de IoT. Dentro de sus principales recomendaciones podemos citar: “Position on the Cybersecurity”, “AIOTI WG11 Report on Smart Manufacturing”, “AIOTI WG08 Report on Smart Cities”, entre otros, [9]. Teniendo en cuenta que los estándares desarrollados hasta el momento no garantizan aún la interoperabilidad de redes heterogéneas, se espera por lo menos, promover puntos en común (entre fabricantes) que garanticen la interoperabilidad de las diferentes redes IoT en capas superiores de la arquitectura.

Por último, el coste asociado al desplegar un sistema IoT dependerá de los requerimientos de los usuarios finales (número de nodos, números de usuarios, sensores, etc.), del hardware utilizado y el

valor agregado al usuario final. Por lo tanto, se podrá implementar una solución de bajo coste para sistemas IoT (usando hardware de bajo coste como Arduino o Raspberry Pi, con una programación y un fin específico), o se podrá contratar una solución propietaria y cerrada de alguna empresa y/o fabricante (Libelium, Prosys de Bosh, DGlogik). El hecho de solicitar una solución propietaria tiene un coste económico mayor, el cual dependerá del valor añadido que represente para el usuario final. Tomando esto en consideración, este trabajo final de carrera, se focaliza en proponer una arquitectura de bajo coste para aplicaciones de IoT con capacidades similares a soluciones desarrolladas por fabricantes de IoT ya existentes.

## 1.1. Objetivos

El objetivo principal de este proyecto es diseñar e implementar un prototipo de bajo coste para aplicaciones punto a punto dentro del ecosistema IoT, mediante la comunicación entre un usuario final y un nodo central a través de la tecnología LoRa. Para cumplir este objetivo nos hemos planteado los siguientes objetivos específicos:

- Proponer una arquitectura para sistema IoT que minimice los costes de implementación, donde involucre una red de sensores, un transceptor como usuario final, un nodo central como estación base y un centro de control para el procesado de la información.
- Implementar para el usuario final un transceptor usando un microcontrolador de bajo coste del fabricante Arduino cuya función principal sea la captación de los datos obtenidos de los sensores y su posterior transmisión a un nodo central.
- Programar un nodo central de bajo coste que actúe como pasarela con la unidad de procesado y el usuario IoT.
- Garantizar las capacidades de captación de datos, comunicación, y gestión eficiente de los recursos radio tanto para el usuario como el nodo central IoT.
- Desarrollar e implementar un sistema local centralizado para la captación y procesado de la información obtenida de los sensores IoT, el cual estará compuesto por la unidad de procesamiento a través de la herramienta matemática MATLAB, y una base de datos creada y gestionada por el sistema MySQL.
- Validar el sistema IoT mediante medidas de campo en un entorno controlado para una correcta verificación de sus funcionalidades.
- Comparar la solución propuesta de nuestro sistema IoT, respecto a la solución de un fabricante, en función de los costes de implementación.

## 1.2. Estructura de la memoria

El contenido de este documento está organizado y distribuido como se detalla a continuación en los siguientes puntos:

- Capítulo 2: presenta un estado del estado de arte de IoT donde se analizan sus principales arquitecturas y tecnologías. Además, se detalla sobre la tecnología LoRa, y el hardware de bajo coste del fabricante Arduino, los cuales son la base en la que se sustenta la implementación del sistema IoT de este proyecto.
- Capítulo 3: se aborda el diseño del sistema IoT donde en primer lugar se detalla la arquitectura planteada; describiendo detalladamente los componentes que la conforman, (transceptores del usuario final y del nodo central, unidad de procesado y almacenamiento de la información y especificaciones de los sensores utilizados). En segundo lugar, se detallan todas las funciones que se han desarrollado para la implementación de los transceptores que forman el nodo del usuario final y el nodo central. Asimismo, se detalla el proceso llevado a cabo para el almacenamiento de los datos captados por los sensores conectados al usuario.
- Capítulo 4: se presenta el proceso de validación del sistema en un entorno real, dividiendo este proceso de pruebas en dos partes: por un lado, la correcta captación de los datos por el nodo del usuario final y por otro lado, el correcto funcionamiento de las prestaciones radio del equipo, tales como estimación de canal, verificación de una correcta canalización dentro de la banda espectral en la que se transmite, una correcta optimización del nivel de potencia recibida y un buen nivel de señal a ruido en recepción que permita una correcta decodificación de los datos transmitidos. Además también se realiza una comparación del sistema IoT de bajo coste propuesto en este proyecto con otro desarrollado en un TFG anterior en el que se implementó un sistema IoT con un equipo de la empresa Libelium. También se presenta un estudio económico que compara detalladamente los presupuestos necesarios para desarrollar uno u otro.
- Capítulo 5: se presentan las conclusiones extraídas para conseguir diseñar e implementar un sistema IoT de bajo coste que funcione correctamente en la transmisión y análisis de los datos. Además, también se proponen y se analizan posibles líneas de trabajo futuro.

## Capítulo 2

# Estado del arte

### 2.1. Historia y evolución

Internet of Things es un término ya extendido en nuestra sociedad, que ha sufrido una gran evolución en la última década. Para hablar de sus inicios es necesario remontarse al año 1999, cuando el británico Kevin Ashton habla por primera vez de IoT, concretamente utilizó dicho término para referirse a un entorno en el que los objetos de la vida cotidiana se conectarían a Internet haciendo uso de la tecnología RFID (*Radio Frequency IDentier*). No obstante, por aquel entonces no pudo tener un crecimiento potencial debido a las limitaciones tecnológicas. Años más tarde las redes de comunicación de datos y las tecnologías existentes fueron mejorando y la idea de IoT continuó creciendo con fuerza y generando una gran expectación. IoT cobra una gran importancia porque es la primera evolución real de Internet; se trata de un salto que dará lugar a aplicaciones revolucionarias que tienen el potencial de mejorar significativamente la manera de vivir, dado que se trata de un ecosistema que está y estará más aún, presente en nuestra vida cotidiana. Según se pronostica, IoT tendrá su máxima extensión comercial en 2020, y es que, según las predicciones, que en un futuro no muy lejano el ecosistema IoT abarcará la gestión eficiente de recursos que están presentes en nuestra sociedad como: un coche, una casa, una fábrica o incluso una ciudad. Todos ellos estarán conectados transmitiendo los datos que puedan ser de interés para el usuario, formando así, redes de trabajo en nuestro entorno, tal y como se muestra la figura 2.1.



Figura 2.1: Representación de posibles escenarios IoT

## 2.2. Capacidades de un sistema IoT

Todo sistema IoT debe comprender ciertas capacidades, las cuales se ven reflejadas en los bloques que componen el sistema de la figura 2.2. Estos bloques no son más que una representación de las tareas que debe realizar una plataforma IoT para llevar a cabo un correcto funcionamiento [4].

Bloques sistema IoT



Figura 2.2: Bloques sistema IoT

La primera tarea que se debe realizar es la **identificación**, una labor crucial dentro de un sistema IoT. Su finalidad es identificar mediante una etiqueta única a cada dispositivo dentro de la red. Un correcto direccionamiento es muy importante para no confundir la ID de los objetos con sus direcciones<sup>1</sup>. Por tanto, para direccionar los objetos se utilizan los protocolos IPv4 e IPv6. No obstante,

<sup>1</sup>Con la ID de los objetos se está haciendo referencia al nombre que se le ha puesto a un dispositivo en concreto, por

actualmente otras tecnologías proponen alternativas de direccionamiento como es el caso de 6LoWPAN [10]. Esta proporciona un mecanismo de compresión sobre las cabeceras de IPv6, lo que hace posible un direccionamiento apropiado para las redes de baja potencia.

Una vez identificado cada objeto, se debe proceder a la tarea de **captación** de datos. En IoT dicha tarea empieza en los sensores y acaba cuando la información es depositada en una base de datos o en la nube; aunque el trabajo de medición o captación es principalmente realizado por sensores inteligentes, actuadores o dispositivos “*wearables*” que en la mayoría de las veces están integrados en ordenadores de placa reducida (en inglés *Single Board Computers -SBCs-*) como Arduino Yun, Raspberry PI, BeagleBone Black, etc. Dichos dispositivos analizan las órdenes recibidas de capas superiores y las ejecutan sobre aquellos objetos encargados de medir.

Seguidamente a la captación de datos se procede al envío de los mismos; para ello se emplean tecnologías de **comunicación**, encargadas de proporcionar una conexión inalámbrica entre el nodo del usuario y el nodo central o entre nodos de usuario, entre otros. Entre las tecnologías inalámbricas más importantes propuestas para IoT se pueden destacar: LoRa/LoRaWAN, tecnologías móviles 3GPP (NB-IoT, eMTC, EC-GSM), WiFi, Bluetooth, RFID y *Near Field Communication*(NFC).

Tras la recepción de los datos el bloque de **computación** procesa la información recibida. Los principales motores que llevan a cabo esta tarea son hardware específicos, bases de datos, cloud computing, y módulos encargados del procesado de big data; aunque también existen plataformas, únicamente software, como los sistemas operativos de tiempo real (RTOS, *Real Time Operating Systems*), Android o plataformas en la nube como Nimbits que complementan el trabajo de computación de los primeramente mencionados.

Luego del procesado de los datos, se encuentra el bloque de **servicios**, que se encarga de gestionar dichos datos de tal forma que puedan ofrecer un determinado servicio para el usuario. Los servicios que puede abarcar un sistema IoT se podrían subdividir en cuatro grupos: servicios de identidad vinculada, servicios de agrupación de información, servicios de cooperación y servicios ubicuos.

- El servicio de identidad vinculada es el más básico e importante; simplemente consiste en identificar y relacionar un objeto del mundo virtual con su correspondiente exportado del mundo real.
- El servicio de agrupación de información se encarga de coleccionar y organizar los datos previamente medidos y procesados que serán necesarios para la aplicación IoT.
- El servicio de cooperación trabaja sobre los servicios encargados de recopilar información, de los que obtiene los datos y toma una decisión acorde a ellos.

---

ejemplo, un sensor que mide la humedad se le asigna la ID “SENSOR1”. Con la dirección del objeto se hace referencia a la ruta de direccionamiento dentro de la red de comunicaciones.

- Los servicios ubicuos se encargan de controlar el sistema al completo y observar su comportamiento.

Por último, el bloque **semántico**, se encarga de extraer la información procesada y analizada para proporcionar los servicios requeridos para la aplicación del usuario final. Esta tarea puede realizarse mediante tecnologías semánticas Web, como RDF (*Resource Description Framework*), OWL (*Web Ontology Language*) o EXI (*Efficient XML Interchange*).

### 2.3. Arquitectura

Principalmente existen 3 tipos de arquitecturas para IoT: la arquitectura más básica es la que está compuesta por tres capas, seguida por la que recomienda la ITU-T (una arquitectura de cuatro capas), aunque la mayoría de fabricantes siguen un modelo de cinco capas que varía según la aplicación. La figura 2.3 muestra un esquemático de las arquitecturas más relevantes para IoT.

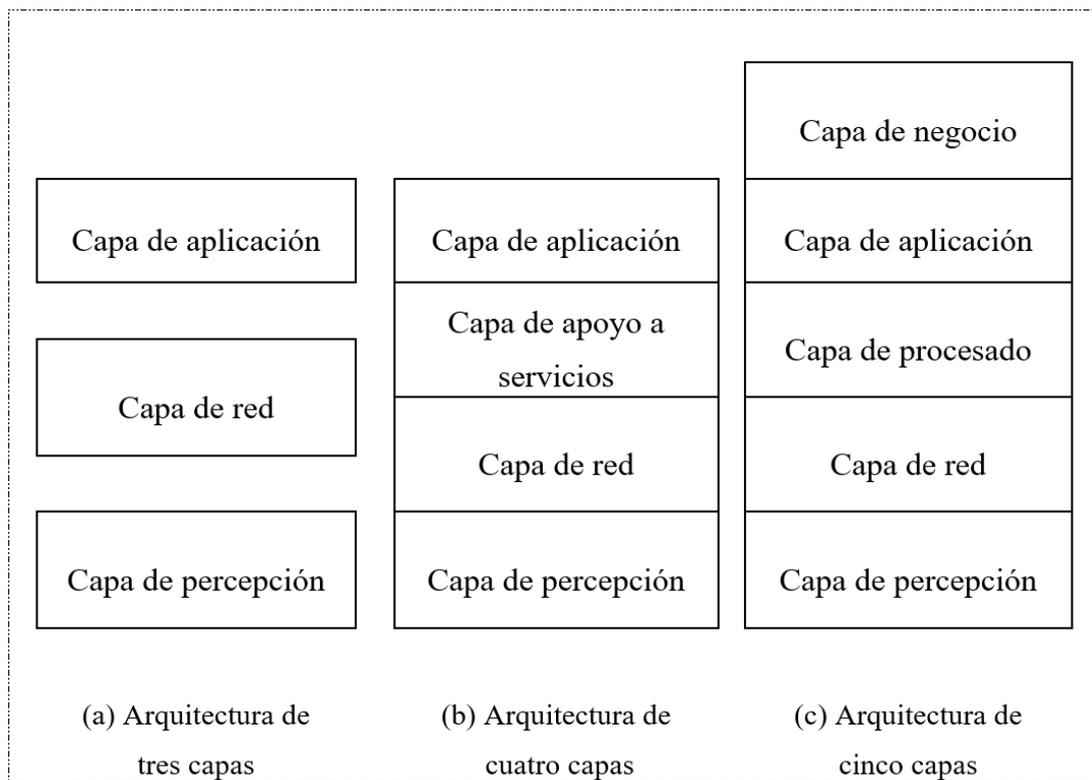


Figura 2.3: Tipos de arquitecturas IoT

Como se muestra en la figura anterior, la arquitectura de tres capas es la más sencilla de las tres propuestas, la cual se utiliza en aplicaciones simples que no requieren una posterior ampliación de las capacidades de la red [11]. Respecto las arquitecturas de cuatro y cinco capas, su principal diferencia es que de la cinco tiene una capa extra denominada “capa de negocio”, la cual gestiona las actividades,

los servicios y los resultados, que posteriormente son estudiados para llevar a cabo posibles acciones de optimización y mejora dentro del sistema. A continuación, se detalla la arquitectura IoT de cinco capas.

### 2.3.1. Arquitectura de cinco capas

La arquitectura de cinco capas está compuesta por: la capa de objetos o de percepción, la capa de abstracción o de red, la capa de procesamiento o Middleware, la capa de aplicación y la capa de negocio, tal y como se puede observar en la figura 2.4 [12].

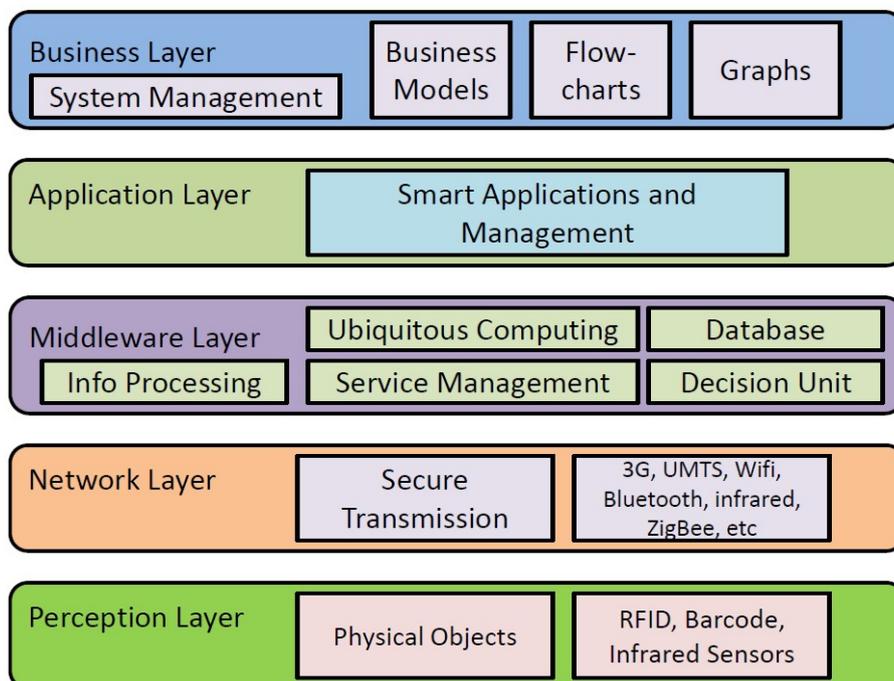


Figura 2.4: Arquitectura IoT de cinco capas

La **capa de percepción** representa los sensores físicos de IoT, cuyo objetivo es el medir y recopilar información como, localización, temperatura, peso, velocidad, aceleración, humedad, etc. Después de la medición de dichos parámetros, los datos captados son digitalizados y transferidos a la siguiente capa (la capa de red).

La **capa de red** es la encargada de transferir los datos previamente captados en formato digital, de manera segura, desde la capa de captación hasta la capa de procesado y vice-versa. Para ello se debe realizar una correcta modulación de la información en transmisión y una correcta demodulación de la información en recepción.

En la capa de red pueden emplearse diversas tecnologías según las necesidades requeridas. A continuación, se va a realizar una clasificación completa de las principales, especificando sus datos técnicos.

Posteriormente se realiza una agrupación en dos grandes conjuntos en función de su área de cobertura/servicio; WPAN (*Wireless Personal Area Network*) y WWAN (*Wireless Wide Area Network*), donde a su vez en WWAN se hace una distinción entre las tecnologías estandarizadas por el 3GPP (*3rd Generation Partnership Project*) y las que se consideran de bajo consumo (LP - *Low Power*).

### WPAN

- La tecnología RFID trabaja mediante un chip o una etiqueta adherida a un objeto, que proporciona la identidad de dicha “cosa”. El funcionamiento se basa en el envío de una señal electromagnética desde la fuente a una etiqueta/chip, la/el cual provoca una corriente de inducción reflejada; posteriormente la señal reflejada se detecta por la fuente y es transmitida a la base de datos para verificar la identidad dicho objeto. Esta tecnología se usa sobre todo para la transmisión M2M y su rango de trabajo oscila desde 10cm hasta 200m dependiendo de si estamos ante etiquetas receptoras pasivas o activas [13].
- El protocolo NFC trabaja en la banda de frecuencias de 13.56MHz y soporta una tasa de aproximadamente 424 kbps. Las comunicaciones se pueden dar entre sensores activos y “etiquetas” pasivas o entre sensores y “etiquetas” activas. Su rango de trabajo se encuentra entre los 10 cm, aunque puede variar notablemente dependiendo del tipo de etiqueta [14].
- Bluetooth presenta una tecnología capaz de intercambiar datos entre dispositivos que se encuentran a escasos metros (entre 10 y 100, dependiendo de la potencia de transmisión). Cabe destacar que, desde la salida de Bluetooth 4.1, esta tecnología permite conectividad mediante IP, lo que le relaciona directamente con IoT [15].

ZigBee es un sistema de comunicación inalámbrica que tiene una baja tasa de transmisión y un bajo consumo energético. ZigBee esta compuesto por tres categorías de nodos: el coordinador, que se encarga de controlar toda la red y los caminos de su comunicación; el router, que interconecta los nodos ofreciendo un nivel de aplicación; y el dispositivo final, que únicamente recopila información y lo transmite al nodo padre, siendo capaz de operar en modo de espera o en marcha para alargar así la duración de la batería.

### WLAN

- La tecnología WiFi permite intercambiar información mediante comunicación vía radio entre objetos que estén separados hasta un máximo de 100 metros, por lo que abarca un área de cobertura de WPAN y WLAN. Puede trabajar tanto en la banda de 2.4 GHz como en la banda de 5 GHz. Esta última, ha sido recientemente habilitada y, al no existir otras tecnologías (Bluetooth, ZigBee, WUSB) que la utilicen, se producen muy pocas interferencias. En esa banda su alcance es algo menor que en la banda de 2,4 GHz (aproximadamente un 10 % menor) [15].

## WWAN

- Narrowband IoT (NB-IoT) es un estándar de comunicaciones radio desarrollado por 3GPP que también pertenece al grupo de tecnologías de LPWAN (en inglés, *Low Power Wide Area Network*). Este estándar permite la conexión de una amplia gama de dispositivos y servicios celulares. Cabe añadir que esta red utiliza la ya existente red móvil para conectar todos esos objetos. Es una de las tecnologías competidoras de Sig-Fox, LoRaWAN... que están explicadas a continuación.
- EC-GSM-IoT es una tecnología estandarizada que está basada en eGPRS y está diseñada como un sistema celular de alta capacidad, largo alcance, baja energía y baja complejidad para comunicaciones IoT. La batería de los dispositivos que usan esta tecnología puede alcanzar una duración de hasta 10 años, permitiendo así, numerosas posibilidades.
- LoRa es una tecnología que se puede clasificar dentro de conjunto de WWAN, ya que puede llegar en condiciones óptimas a alcanzar los 11km de cobertura. Las frecuencias de trabajo están las bandas no licenciadas del espectro (ISM), 433MHz, 470MHz, 868MHz y 915MHz, dependiendo de la zona geográfica de trabajo y puede alcanzar un *throughput* en la transmisión de datos de hasta 50kbps. Además, se ha clasificado como una tecnología de bajo coste energético por lo que aporta una larga vida a las baterías de los transmisores (en caso de no utilizar la red eléctrica como suministradora). Asimismo, con LoRaWAN (la evolución de LoRa), se pueden crear redes de trabajo multipunto fácilmente sin que la red sufra pérdidas de paquetes [16].
- Sigfox es una tecnología destinada principalmente a la transmisión de datos en las redes IoT. En la red Sigfox se transmiten mensajes de 12 bytes, pudiendo enviar hasta 140 mensajes al día. Esta tecnología aporta una conectividad bidireccional y su frecuencia de transmisión se ubica en bandas ISM, por lo que, dado que tiene cobertura internacional, dependiendo del lugar geográfico de trabajo, la transmisión deberá realizarse en una o en otra banda. Su rango de trabajo tiene un promedio de unos 30-50 km en las zonas rurales y de entre 3 y 10 km en las zonas urbanas, donde hay más obstáculos y el ruido es mayor.

A modo de resumen la siguiente tabla presenta las principales tecnologías según su alcance y sus parámetros principales:

WPAN				
Tecnologías	Estandarización 3GPP	Alcance	Banda de frecuencias	Velocidad de datos
Bluetooth	Estandarizado	1-100 metros	2,4 GHz	1 Mb/s
NFC	Estandarizado	Rango muy corto	13.56 MHz	106, 212 o 424 kbps
RFID	Estandarizado	Etiquetas pasivas 3-5 m Las activas <100 m	868,9 MHz	-
WiFi	Estandarizado	10-100 metros	2,4 GHz / 5 GHz	54 Mb/s
ZigBee	Estandarizado	10-100 metros	868 MHz Europa, 915 MHz EEUU y 2,4 GHz resto	250 kbit/s
WLAN				
Tecnologías	Estandarización 3GPP	Alcance	Banda de frecuencias	Velocidad de datos
WiFi	Estandarizado	10-100 metros	2,4 GHz / 5 GHz	54 Mb/s
WWAN				
3GPP Estándares				
Tecnologías	Estandarización 3GPP	Alcance	Banda de frecuencias	Velocidad de datos
NB-IOT	Estandarizado	<15 km	Licenciada 700-900 MHz	<150 kbps
EC-GSM	Estandarizado	<15 km	Licenciada 800-900 MHz	10 kbps
LPWANs				
LoRaWAN	No estandarizado	<11 km	No licenciada 433 MHz / 868 MHz	<10 kbps
Sigfox	No estandarizado	<13 km	No licenciada 868 / 915 MHz	<100 kbps

Cuadro 2.1: Tecnologías inalámbricas disponibles en entornos personales, locales y amplios

Sigfox fue la primera tecnología que introdujo una limitación en el modo de transmisión, para cumplir con las regulaciones actuales, ya que los reglamentos europeo y español<sup>2</sup> que regulan las bandas ISM limitan el ciclo de trabajo de transmisión a un 1% con la finalidad de mitigar las interferencias. Actualmente tecnologías como LoRaWAN (cuyo banda de transmisión se ubica en ISM) también siguen dichas limitaciones.

La **capa de procesamiento** es también conocida como la capa middleware. Su misión es almacenar, analizar y procesar grandes cantidades de información provenientes de la capa de red. Para realizar esas tareas se utilizan diversas tecnologías como hardware específicos, bases de datos, *cloud computing*, y módulos encargados del procesamiento de *big data*.

Tal y como se puede ver en figura 2.2 donde se muestra un esquemático de las plataformas software disponibles encargadas de dar soporte a la capa de procesamiento [4]; cada plataforma tiene unas características apropiadas para un determinado tipo de aplicación. De entre las plataformas previamente expuestas destaca Xively, que pertenece a la empresa Google la cual también posee una plataforma para servicios de IoT inteligentes. Por su parte Xively es una plataforma IoT empresarial que abarca muchos campos y cuya función es la de ayudar a acelerar un producto o un servicio conectado. Sobresale también por ofrecer “*Cloud Services*”, “*Business Services*” y “*Partner Network*”. En contraposición a la anterior destaca Nimbits, la cual es una plataforma que ha apostado por focalizarse en el uso

<sup>2</sup>Disponible en la Orden IET/787/2013, de 25 de abril, por la que se aprueba el cuadro nacional de atribución de frecuencias, dentro del Boletín Oficial del Estado.

aplicaciones de control como REST o XMPP. Nimbits puede ser usado para diversidad de acciones, desde *logging time* y geo codificación de los datos sensados hasta para responder a eventos en tiempo real.

Plataforma	Gateway	Provisión	Seguridad	Facturación	Protocolo de aplicación			
					REST	CoAP	XMPP	MQTT
Arkessa	-	+	+	-	+	-	-	+
Axeda	+	+	+	+	+	-	-	-
Etherios	+	+	+	-	+	-	-	-
LittleBits	-	-	-	-	+	-	-	-
NanoService	+	+	+	-	+	+	-	-
Nimbits	-	-	-	-	+	-	+	-
NinjaBlocks	+	-	-	-	+	-	-	-
OnePlatform	+	+	+	-	+	+	+	-
RealTime.io	+	+	-	-	+	-	-	-
SensorCloud	+	+	-	-	+	-	-	-
SmartThings	+	+	-	-	+	-	-	-
TempoDB	-	-	-	-	+	-	-	-
Thingworx	-	+	+	-	+	-	-	+
Xively	+	+	+	+	+	-	-	+

Cuadro 2.2: Plataformas en la nube de IoT y sus características

La **capa de aplicación** es la responsable de ejecutar servicios específicos para el usuario, siendo la interfaz por la cual los usuarios finales pueden interactuar con uno o varios dispositivos y realizar consultas sobre datos de interés o incluso realizar algún tipo de configuración en remoto. En la mayoría de despliegues IoT es necesario una interacción con el usuario, de ahí la importancia de esta capa.

La **capa de negocio** se encarga del sistema IoT al completo que también engloba las cuatro capas inferiores, incluyendo las aplicaciones, servicios, modelos de utilidad y privacidad de usuarios. Esta capa puede ayudar a determinar las futuras acciones y estrategias de negocio que se deben seguir para sacarle un mayor provecho, si cabe, a la aplicación en su conjunto. Además, en este nivel se comparan los datos salientes de cada capa con los datos esperados, para mejorar así, los servicios y mantener la privacidad de los usuarios. También puede proporcionar una interfaz donde se pueda visualizar análisis y gráficas que permitan plasmar una oportunidad de optimización en el sistema. Esto obliga a alojar los componentes pertenecientes a esta capa en dispositivos con suficiente potencia como para cubrir sus necesidades de computación.

Por último, la tabla 2.3 relaciona las capacidades que debe cumplir una red IoT dentro de la arquitectura de cinco capas detallada anteriormente. Es importante resaltar, que la capa de negocio sería la única que no estaría completamente relacionada al completo con un bloque de tareas, pero sí relacio-

nada parcialmente con el bloque semántico, el cual le aporta toda la información y todos los datos en el formato ideal para que dicha capa pueda realizar las acciones pertinentes.

Arquitectura	Partes del proceso de un sistema de IOT		Tecnologías, sistemas o herramientas empleadas
Capa de percepción	Identificación	Nomenclatura	EPC, uCode
		Direccionamiento	IPv4, IPv6
	Sensado		Sensores inteligentes, dispositivos de sensado "wearable", sensores integrados, actuadores, RDIF
Capa de red	Comunicación		RFID, NFC, UWB, Bluetooth, Sigfox, LoRa, Z-Ware, WiFi, WiFiDirect, NB-IoT, EC-GSM...
Capa de procesado	Procesado	Hardware	Arduino, Phidgets, Intel Galileo, Raspberry Pi, Gadgeteer, BeagleBone, Cubieboard, Smart Phones
		Software	OS (Contiki, TinyOS, LiteOS, Riot OS, Android); Cloud (Nimbits, Hadoop, etc)
Capa de aplicación	Servicios		Identidad vinculada (transporte), agrupación de información (red inteligente), servicio de cooperación (smart home), servicio ubicuo (smart city)
	Semántica		RDF, OWL, EXI
Capa de negocio			

Cuadro 2.3: Tecnologías disponibles para un sistema IoT

## 2.4. Tecnología LoRa

Para cubrir las necesidades de la red IoT que se va a proponer en este proyecto es necesario utilizar una tecnología de transmisión que pertenezca al conjunto de LPWAN (*Low Power Wireless Area Network*). Por ello, de entre las tecnologías de baja potencia expuestas anteriormente en la capa de red, se selecciona LoRa (tecnología desarrollada por Cycleo, empresa que actualmente pertenece al fabricante Semtech) cuyas características se adecuaban perfectamente a las de una red de IoT. Entre sus ventajas destacan:

- El alcance de transmisión, ya que permite cubrir un radio de cobertura de hasta 15km en campo abierto y de 3km de alcance entre transmisor y receptor en escenarios urbanos [17]. En ambos casos hay que tener en cuenta que la calidad, tanto de los módulos empleados, como de las antenas, influyen significativamente en el alcance.
- El bajo coste a la hora de la implementación, ya que estamos ante una tecnología que trabaja en bandas ISM, lo que significa que no hace falta adquirir una licencia para transmitir. Aunque esto trae consigo una desventaja, y es que se debe convivir con un mayor nivel de interferencias. Por

ese motivo, es necesario implementar funciones que refuercen al sistema ante actividad de otros equipos adyacentes a la red en cuestión.

- La seguridad que sigue (cifrado AES128) es una condición indispensable para lograr una protección fiable de los datos e información transmitidos. En la evolución al protocolo LoRaWAN se incluye dos niveles de seguridad uno para la red y otro para la aplicación.

En referencia a la topología que sigue el protocolo LoRaWAN, suele desplegarse en forma de estrella, donde los receptores o enrutadores interceptan los mensajes de los dispositivos finales y los redirigen hacia un servidor central. Aunque, las topologías punto a punto o mallado también son de un uso frecuente. En la topología punto a punto se establece una conexión directa entre los nodos, sin embargo, en una red con topología de mallado todos los nodos pueden estar conectados entre sí. Para poder optar por una u otra topología, es fundamental realizar un estudio previo en el que se analicen parámetros como: latencia, *throughput*, flexibilidad, escalabilidad, etc.; de ese modo se podrá implementar la tecnología LoRa de forma óptima.

#### 2.4.1. Clases de dispositivos

El protocolo LoRaWAN es un protocolo asíncrono basado en ALOHA, lo que significa que un dispositivo final se puede “despertar” en intervalos programados para verificar los mensajes del enlace descendente y de sincronización, reduciendo así la latencia de comunicación y el consumo, respectivamente. Para ello, LoRaWAN utiliza diferentes clases de dispositivo con una latencia en la comunicación del enlace descendente diferente, con el objetivo de optimizar la duración de la batería. Teniendo en cuenta las necesidades de la aplicación, LoRaWAN considera tres clases de dispositivos finales, las cuales se muestran en la figura 2.5 y se pueden describir brevemente de la siguiente manera:

- Clase A - Los dispositivos finales de Clase A permiten comunicaciones bidireccionales por lo que a cada transmisión de *uplink* que realiza un dispositivo le siguen dos ventanas de recepción de *downlink*. El *slot* de transmisión programado por el dispositivo final, se basa en sus propias necesidades de comunicación con una pequeña variación basada en un tiempo aleatorio (protocolo tipo ALOHA). Debido a su comunicación condicionada estos son los dispositivos de menor consumo energético.
- Clase B - Los dispositivos finales de Clase B permiten más *slots* de recepción. Además de las ventanas de recepción aleatorias de clase A, los dispositivos de clase B abren ventanas de recepción adicionales a horas programadas. Con la finalidad de que el dispositivo final sepa en qué momento debe abrir estas ventanas, este recibe una señal sincronizada en el tiempo por parte del receptor/enrutador, de ese modo el servidor sabe cuando el dispositivo está escuchando.

- Clase C - Los dispositivos finales de Clase C tienen ventanas de recepción abiertas casi continuamente, solo se cierran cuando se transmite. Los dispositivos finales de clase C usan más energía para operar que la clase A o la clase B, pero ofrecen una latencia más baja para la comunicación entre el servidor y el dispositivo final.

A continuación, en la siguiente figura se recoge un esquema que muestra las funcionalidades de cada una de las clases anteriormente definidas [18]:

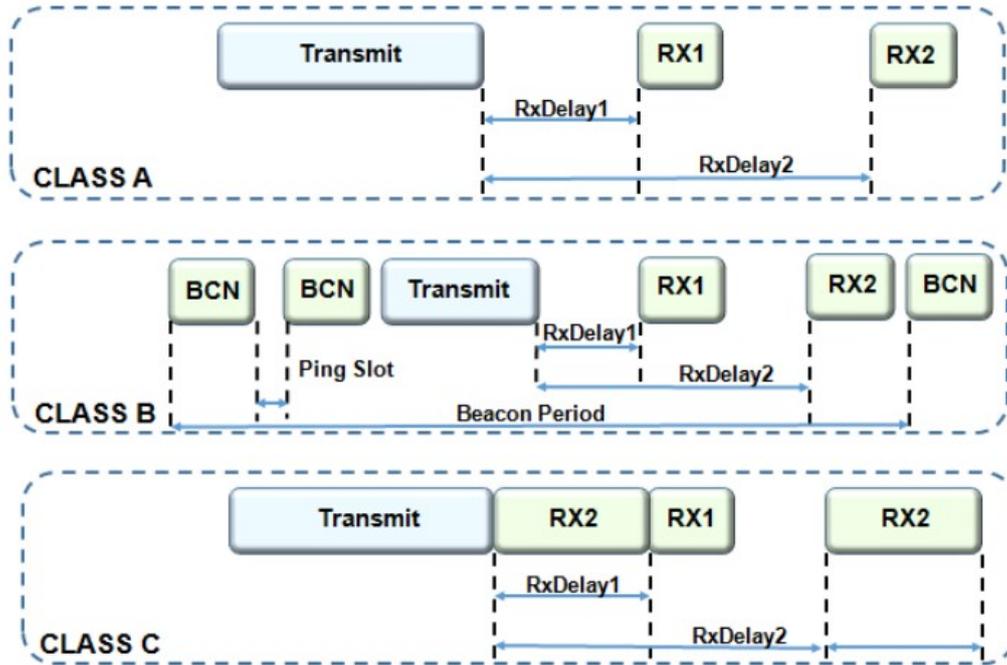


Figura 2.5: Clases de dispositivos de tecnología LoRa

De ese modo, a la hora de diseñar una red que sea dependiente de la tecnología LoRa, existirán tres tipos de clases para elegir según las necesidades que requiera el sistema.

### 2.4.2. Estructura de trama

Haciendo referencia al formato físico de un mensaje transmitido LoRaWAN, podremos discriminar en tramas de *uplink* y tramas de *downlink* [19].

En el **enlace ascendente** los mensajes son enviados por dispositivos finales al servidor de red haciendo uso de una o varias puertas de enlace. Estos mensajes utilizan el modo explícito de LoRa en el que se incluyen un preámbulo de 8 símbolos, un encabezado físico LoRa (PHDR) que ocupa 4 octetos, dejando 252 octetos para los datos, más un CRC (*Cyclic Redundancy Check*) de encabezado (PHDR\_CRC) y otro CRC al final de la trama, lo que asegura la integridad de la carga útil. Los campos

PHDR, PHDR\_CRC y CRC son insertados por el transceptor de radio. Con ello, la estructura de la trama que se envía en *uplink* se muestra en la figura 2.6:



Figura 2.6: Estructura física de un mensaje de *uplink*

En el **enlace descendente** los mensajes son enviados por el servidor de red a un solo dispositivo final y es retransmitido por un único *gateway*. Los mensajes de enlace descendente usan el modo explícito de paquete de radio en el que se incluyen un preámbulo de 8 símbolos, un encabezado físico LoRa (PHDR) que ocupa 4 octetos, un CRC de encabezado (PHDR\_CRC) y un campo de 252 octetos para los datos PHYPayload. De ese modo la estructura de un mensaje transmitido en *downlink* sería la que se muestra en la figura 2.7:

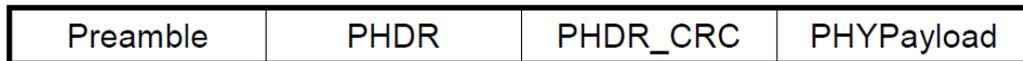
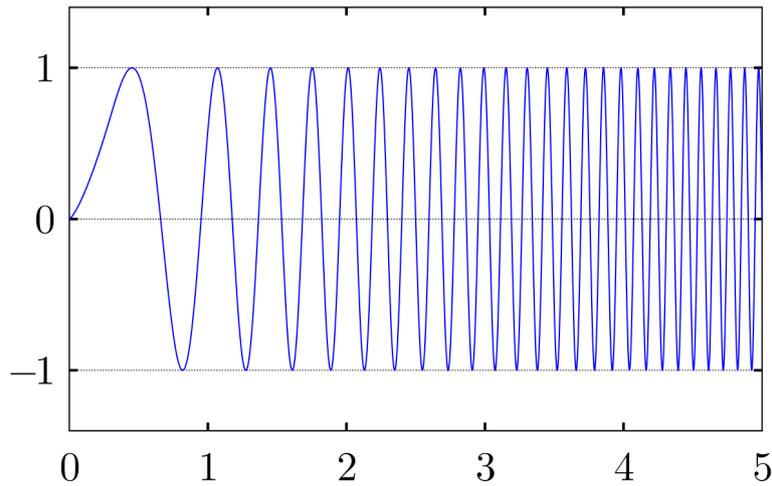


Figura 2.7: Estructura física de un mensaje de *downlink*

Con este tipo de mensajes se realiza la comunicación entre el nodo central y el nodo o los nodos de los usuarios.

### 2.4.3. Parámetros de transmisión

Las técnicas que se siguen en la transmisión se basan en espectro ensanchado y una variación de CSS (*Chirp Spread Spectrum*), que se caracteriza por modular los datos en diferentes canales y velocidades utilizando FEC (*Forward Error Correction*) para realizar una correcta corrección de errores. En la figura 2.8 se puede observar dicha técnica, en la que el efecto del chirp hace que la frecuencia aumente con el paso del tiempo [17].

Figura 2.8: Modulación *Chirp Spread Spectrum* en LoRa

LoRa utiliza toda la anchura de banda del canal para transmitir una señal haciéndola así más robusta al ruido del canal e insensible a las compensaciones de frecuencia. También utiliza una ganancia de codificación que aporta una mayor sensibilidad al receptor, lo que implica más eficacia a la hora de demodular los paquetes entrantes. De ese modo se puede llegar a trabajar en recepción con señales que alcancen hasta los 19.5 dB negativos de SNR.

## 2.5. Hardware IoT

Implementar un sistema de Internet of Things requiere un componente hardware que sea capaz de interactuar con el entorno físico que le rodea. También, es necesario un componente software adecuado para gestionar la información generada y actuar sobre el hardware anteriormente mencionado. La sencillez de instalación y desarrollo que presentan estos dispositivos, así como su gran potencial, capacidad y eficacia a la hora de realizar su labor son características que, junto con su reducido precio, convierten a las placas de desarrollo Hardware y Software en dispositivos idóneos para crear objetos y ambientes digitalizados en IoT. Los hardware más comunes son Arduino, Raspberry Pi, BeagleBone, Intel Galileo o Cubieboard entre otros. De entre los anteriormente mencionados cada uno destaca por tener unas características especiales:

- **Arduino** se trata de un ordenador de placa reducida que posee una baja capacidad de memoria pero es capaz de procesar las instrucciones de manera muy rápida. Destaca por su bajo precio y su sencillez a la hora de ser implementado.
- **Raspberry Pi** se trata de un pequeño ordenador que destaca por la capacidad que tiene a la hora de alojar grandes programas o incluso sistemas operativos.

- **BeagleBone** destaca por ser una de las principales alternativas hardware a Raspberry Pi. Se trata de una placa similar a Raspberry Pi pero dispone de más pines de conexión, aunque la memoria RAM es inferior.
- **Intel Galileo** fue el primer microcontrolador en seguir la línea de Arduino. Combina la tecnología Intel con el soporte para tarjetas de expansión de hardware de Arduino (llamadas “*shields*”), con su entorno de desarrollo de software y con sus librerías. Destaca por tener un procesador Intel con una arquitectura x86.
- **Cubieboard** se trata de un ordenador de placa reducida fabricado en China que tiene la capacidad de ejecutar sistemas operativos como Android 4 ICS, Ubuntu 12.04 desktop, Fedora 19 ARM Remix desktop, Arch Linux ARM, Debian-based Cubian distribution o OpenBSD.

### 2.5.1. Arduino

Estudiando las características de cada uno de los hardware existentes en el mercado, finalmente se estudia la posibilidad de que el soporte tanto para el transceptor LoRa como para los sensores sea el ordenador de placa reducida de la marca Arduino (figura 2.9).



Figura 2.9: Hardware Arduino UNO

Las placas de Arduino son capaces tanto de leer información (de sensores, por ejemplo) proveniente de los pines de entrada, como activar pines de salida (encendiendo, por ejemplo, un LED). Para ello es necesario transmitirle las instrucciones pertinentes al microcontrolador de la placa utilizando el lenguaje de programación de Arduino (que es una variación del lenguaje C++) y el Arduino Software (IDE), el cual trae ya instalado una lista de todas las placas con las que puede trabajar y un repositorio muy completo de librerías disponibles que se actualizan periódicamente. Los principales motivos por los que se ha elegido este hardware son las ventajas que ofrece sobre otros sistemas:

- Su coste - Arduino ofrece placas a un precio muy económico. La última versión y más actualizada

placa cuesta menos de 40€.

- Multiplataforma - El IDE de Arduino se puede ejecutar en Windows, Macintosh OSX, y en cualquier distribución de Linux. En contraposición de la mayoría de sistemas que están limitados a Windows
- Su entorno de programación es simple y claro - El software de Arduino es fácil de usar para principiantes y suficientemente flexible para usuarios avanzados que deseen realizar tareas más allá de lo convencional.
- Código abierto y software ampliable - El software es de código abierto y está disponible para ser complementado por programadores. Todas las librerías también son de código abierto, por lo que se pueden modificar o complementar sin ningún impedimento.
- Código abierto y hardware ampliable - Las especificaciones de las placas de Arduino están todas publicadas bajo la licencia de Creative Commons, por lo que, fabricantes, diseñadores o incluso los propios usuarios pueden construir sus propios módulos compatibles con Arduino, para mejorar así las posibilidades de la placa.

A continuación, se muestran las características principales de las placas Arduino más comunes actualmente en el mercado:

Característica de Arduino	UNO	Mega 2560	Leonardo	DUE
<b>Tipo de microcontrolador</b>	Atmega 328	Atmega 2560	Atmega 32U4	AT91SAM3X8E
<b>Velocidad de reloj</b>	16 MHz	16 MHz	16 MHz	84 MHz
<b>Pines digitales de E/S</b>	14	54	20	54
<b>Entradas analógicas</b>	6	16	12	12
<b>Salidas analógicas</b>	0	0	0	2 (DAC)
<b>Memoria de programa (Flash)</b>	32 Kb	256 Kb	32 Kb	512 Kb
<b>Memoria de datos (SRAM)</b>	2 Kb	8 Kb	2.5 Kb	96 Kb
<b>Memoria auxiliar (EEPROM)</b>	1 Kb	4 Kb	1 Kb	0 Kb

Cuadro 2.4: Tipos de Arduino y sus características

Dadas las ventajas que presentan los dispositivos de la marca Arduino y visto que sus especificaciones son suficientes para las dimensiones de este proyecto, se opta finalmente por trabajar con este tipo de placas y con su IDE, no solo por su flexibilidad, por su compatibilidad como numerosos módulos y por sus enormes repositorios de información sino que también por su precio tan económico en el mercado.

## Capítulo 3

# Diseño de un sistema IoT

### 3.1. Arquitectura del sistema

En este proyecto, se utiliza una arquitectura de cuatro capas, en la que la capa de percepción está formada por sensores que permiten monitorizar diferentes parámetros, tales como, humedad y temperatura; la capa de red encargada de la comunicación entre el nodo central y el nodo del cliente usa la tecnología LoRa; la capa de procesado es una solución dual entre un software para cálculo numérico Matlab y una base de datos para la gestión eficiente de la información y por último, la capa de aplicación permite ofrecer servicios de monitorización al usuario final. La red se ha diseñado de manera centralizada, de forma que los clientes transmiten toda la información captada por los sensores a un nodo central que hace de *gateway* con la unidad de procesado. La descripción de la arquitectura de la red muestra en la figura 3.1.

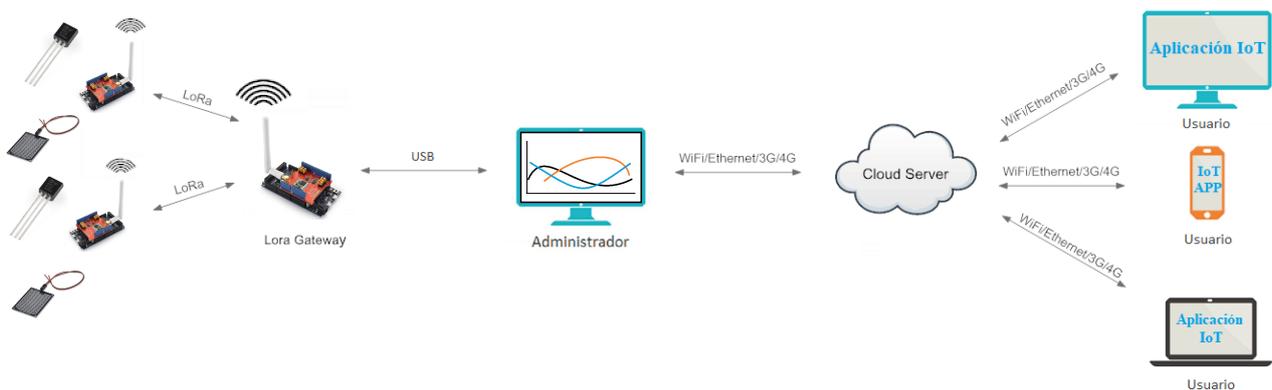


Figura 3.1: Arquitectura de un sistema inalámbrico de bajo coste para aplicaciones de IoT

A continuación, se detalla cada componente de la arquitectura IoT:

- Uno o varios clientes que transmiten al nodo central la información captada por los sensores. En

este caso se está utilizando un sensor de humedad y un sensor de temperatura.

- Un nodo central, gestiona los recursos radio y la calidad del enlace de los usuarios finales. Además, es el encargado de hacer de pasarela entre los clientes y el administrador de la red.
- Una central de gestión de la información, a la que está conectada el *gateway* vía usb, analiza y almacena los datos recibidos. Para ello usa las siguientes aplicaciones:
  - Matlab, es la herramienta matemática encargada de analizar y mostrar por pantalla de forma gráfica los datos recibidos.
  - Una base de datos MySQL, donde se almacena la información previamente analizada. Las aplicaciones de los usuarios finales deberán acceder a ella para obtener la información necesaria.

### 3.2. Sistema del nodo cliente

En la figura 3.2 se muestran los elementos hardware que componen al nodo cliente: un Arduino Mega, un *shield* para comunicaciones LoRa y dos sensores (uno de temperatura y otro de humedad).

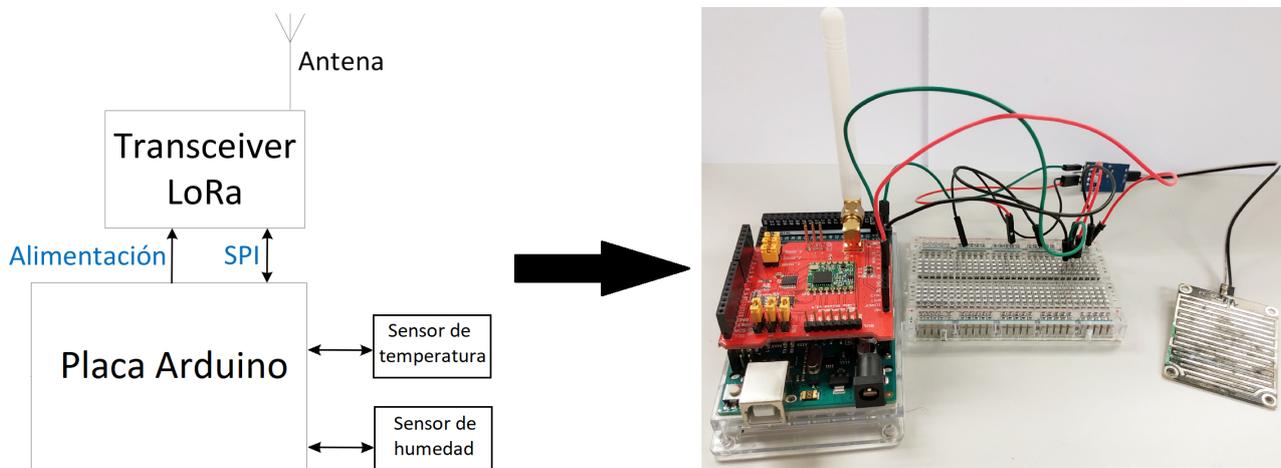


Figura 3.2: Configuración de los elementos para el nodo cliente

Las características físicas junto con las especificaciones técnicas de cada dispositivo se explicarán en el Apéndice A.

#### 3.2.1. Modo de operación general

Para explicar el modo de operación del nodo cliente, se expone en la figura 3.3 un diagrama de flujo en el que se pueden ver las acciones que este nodo sigue en todo momento.

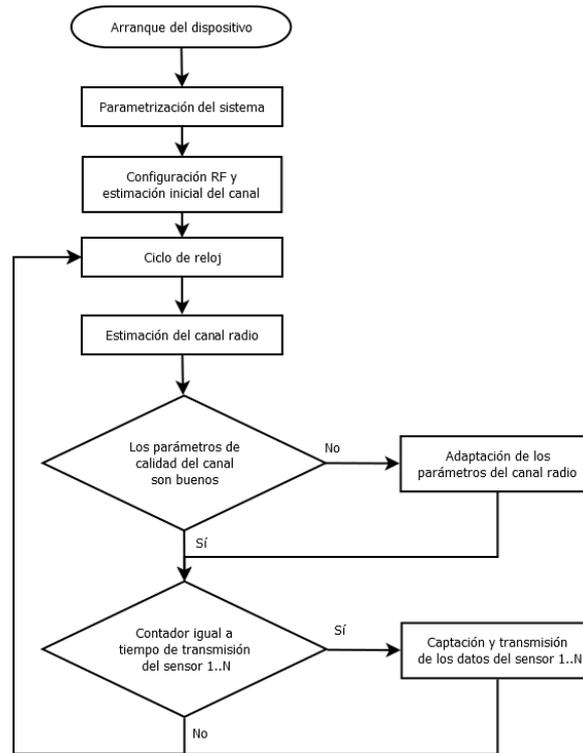


Figura 3.3: Diagrama de flujo general del cliente

Tal y como se puede observar en el flujograma, el **nodo del cliente** está compuesto por tres funciones principales:

- **La primera** función, está compuesta por las acciones encargadas de inicializar variables, crear los objetos necesarios y establecer los parámetros de transmisión y recepción iniciales. Para ello, inicialmente se deben incluir las siguientes librerías:
  - RH\_RF95 y RHEncryptedDriver, pertenecientes a RadioHead cuyo autor es Mike McCauley. Estas dos librerías se encargan de dar soporte a la transmisión mediante la tecnología LoRa. La diferencia entre ellas, es que la segunda que permite añadir una encriptación a los datos transmitidos.
  - Speck, es una librería perteneciente al paquete de Crypto creada por la entidad Southern Storm Software, Pty Ltd. Esta librería es complementaria a RHEncryptedDriver.
  - SPI, librería creada por Arduino que permite la comunicación entre la placa Arduino y el transceptor LoRa, mediante el Bus *Serial Peripheral Interface*.

Una vez incluidas las librerías necesarias, se inicializan las variables globales y los contadores que serán empleados durante todo el programa. Seguidamente se realiza el *setup* de los parámetros radio, que consta de las siguientes partes:

- Configuración de la frecuencia del canal de transmisión en la banda de 868 MHz.
- Configuración del modo de transmisión según las necesidades de alcance o capacidad.
- Configuración de la tabla de encaminamiento de la red.
- Configuración de las cabeceras, indicando el *From* y el *To*.

Para más información sobre las opciones de configuración, consultar el Apéndice C.

- **La segunda** función, es la encargada de estimar el canal a partir de la potencia que recibe el *gateway*. Para una correcta estimación se siguen los siguientes pasos:
  1. Establecer la potencia de transmisión al máximo permitido, 14 dBm. De ese modo el cliente se asegura de que el nodo central recibirá el paquete de estimación, si está dentro de su alcance.
  2. Establecer el ID del mensaje, de tal forma que, en el servidor al leer la cabecera del paquete se reconozca como un paquete de estimación de canal. El contenido en este caso es indiferente, aunque por defecto se envía “Hola, paquete de estimación”.
  3. Esperar a la respuesta del servidor, la cual debe incluir el dato del RSSI medido en la recepción de dicho paquete en el nodo central.
  4. Calcular el *path loss* según la potencia con la que el servidor ha recibido ese paquete. De ese modo, se podrá configurar la potencia de transmisión óptima para la comunicación de los datos. Para ello, es necesario tener en cuenta el modo de transmisión empleado, ya que la sensibilidad en recepción depende principalmente de este.
- **La tercera**, se encarga de medir y transmitir de forma segura los datos tanto de temperatura como de humedad. Para ello, se sigue un patrón muy similar en ambos casos, ya que en el momento en el que se transmite cualquiera de esos datos sucede lo siguiente:
  1. Se lee la información, bien de temperatura o bien de humedad, del sensor pertinente. Después ese dato se convierte al formato indicado para la transmisión. En este caso el objeto de transmisión exige un formato *uint8\_t*.
  2. Se termina de completar la cabecera, estableciendo el ID del mensaje. Este ID cambiará dependiendo de si el paquete contiene información sobre temperatura, sobre humedad o sobre el canal radio (en cuyo caso sería un paquete de estimación).
  3. Tras captar los datos y rellenar todos los campos de la cabecera, el programa ejecuta la función encargada del envío.
  4. El programa accede a una función encargada de recibir la confirmación de recepción (ACK) por parte del servidor. En el caso de no recibir ninguna confirmación, el cliente vuelve a transmitir dichos datos hasta recibir la confirmación.

Es necesario tener en cuenta que el cliente se ha implementado de tal forma que, tiene varios niveles de potencia con los que puede transmitir; estos son 0dBm, 5dBm, 9dBm, 12dBm y 14dBm. La razón por la que se han seleccionado esos niveles y no otros, ha sido principalmente por el parámetro de consumo energético.

### 3.3. Sistema del nodo central

El sistema del nodo central está formado por placa Arduino UNO y un *shield* para comunicaciones LoRa, que puede actuar tanto de receptor como de transmisor. Este nodo hace de pasarela entre el nodo cliente y el ordenador administrador y se muestra en el esquema de la figura 3.4.

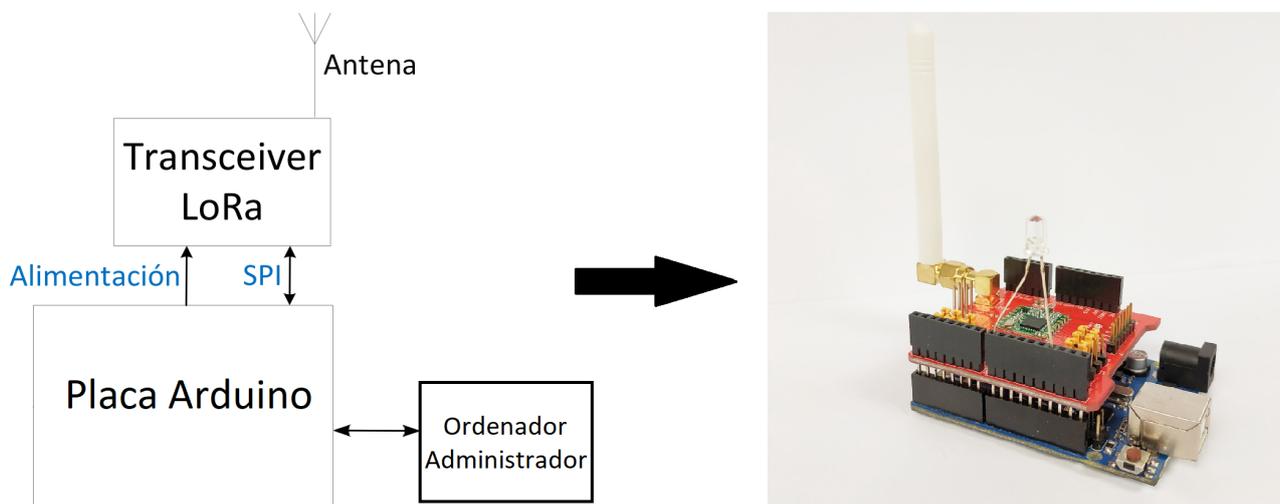


Figura 3.4: Configuración de los elementos para el nodo central

El nodo central es el encargado de configurar correctamente los parámetros de la red y de transmitirle la información recibida (de forma inalámbrica) a la central de gestión. Por lo que, es necesario utilizar una placa rápida y no necesariamente con mucha memoria disponible para datos (SRAM), ya que en este nodo no se almacena la información. Por esas razones, para el *gateway* se ha empleado un Arduino UNO. Para comprobar las características de dicho dispositivo, se puede consultar la tabla 2.4. En relación a las especificaciones técnicas de los demás dispositivos, se podrán consultar en el Apéndice B.

#### 3.3.1. Modo de operación general

A continuación, se estudiarán las funciones implementadas en el *gateway* para una correcta recepción y clasificación de los datos.

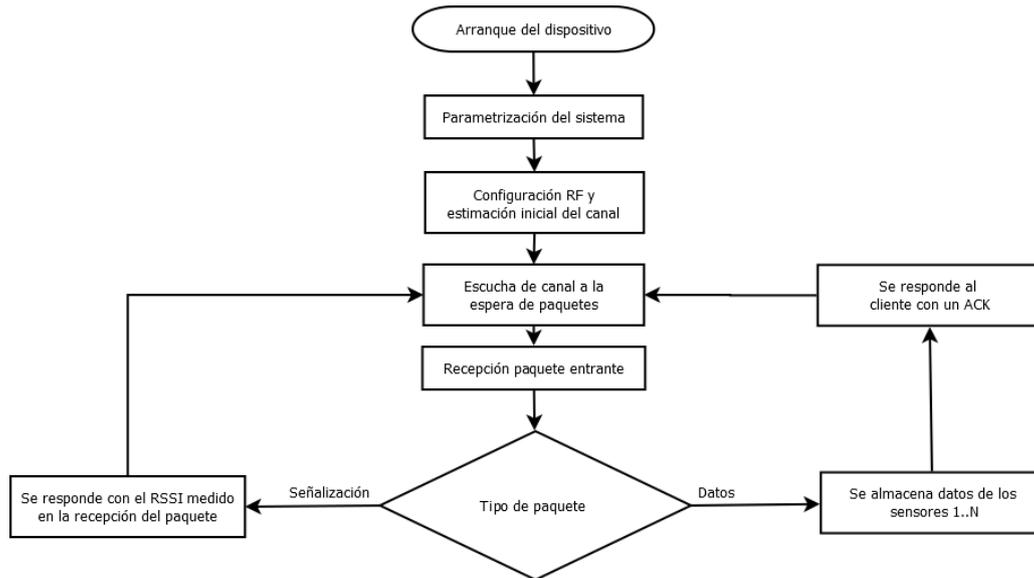


Figura 3.5: Diagrama de flujo general del servidor

Si se observa la figura 3.5, se puede percibir una visión general del funcionamiento del nodo central. A grandes rasgos, su funcionamiento se basa en la recepción, filtrado y clasificación de la información. Analizando en detalle el diagrama de flujo existen las siguientes funciones:

- **La primera**, está compuesta por las acciones encargadas de inicializar variables, crear los objetos necesarios y establecer los parámetros de transmisión y recepción iniciales. Para ello, al igual que en el nodo del cliente se añaden las librerías `RH_RF95`, `RHEncryptedDriver`, `Speck` y `SPI`; se inicializan las variables, los contadores globales y se realiza el *setup* del dispositivo que consta de las siguientes partes:
  - Configuración de la frecuencia del canal de transmisión y de recepción en la banda de 868MHz.
  - Configuración de la potencia de transmisión. Como es el nodo central y está conectado a la red eléctrica, transmitirá a la máxima potencia, para poder cubrir el mayor área posible.
  - Configuración del modo de transmisión.
  - Configuración de su dirección dentro de la red, *thisAddress* y del campo *From* que se pondrá posteriormente en la cabecera de cada paquete enviado. Esos dos parámetros deben coincidir.

Para más información sobre las opciones de configuración, consultar el Apéndice C.

- **La segunda**, se encarga de escuchar, filtrar y almacenar el dato, según el ID de la trama recibida. Para filtrar los paquetes entrantes se sigue un patrón muy similar. El nodo central dispone en su memoria de una tabla en la que se relaciona el número de ID con el tipo de paquete que se acaba de recibir. De ese modo se pueden dar las siguiente posibilidades:

- Que el paquete contenga un dato de temperatura, en cuyo caso se etiqueta y se deja a disposición del ordenador administrador, para que sea procesado y almacenado. Posteriormente se contesta con un ACK a la dirección del remitente (*From*) tal y como se ha explicado previamente.
  - Que el paquete contenga un dato de humedad, en cuyo caso se etiqueta y se deja a disposición del ordenador administrador, para que sea procesado y almacenado. A continuación se contesta con un ACK de igual modo que en el caso anterior.
  - Que se trate de un paquete de estimación de canal, en cuyo caso tan sólo hay que contestar a la dirección del remitente especificando en el cuerpo del mensaje el RSSI que se ha medido tras la recepción de dicho paquete.
- **La tercera** se encarga de responder a cada trama recibida. El proceso que sigue esta función comienza una vez el paquete está clasificado y ordenado. En este caso el nodo central contesta a la dirección *From* del paquete que había recibido; es decir, el *To* del paquete de respuesta será igual al *From* del paquete primeramente recibido.

Es preciso aclarar que, la configuración de los parámetros del *setup* tiene que ir en concordancia con la configuración del resto de nodos del sistema. Para obtener una información más detallada sobre la función de estimación de canal implementada en el nodo central del sistema, se recomienda consultar el Apéndice D.

### 3.4. Sistema administrador

La central de gestión de la información, es un ordenador que está ubicado entre el *gateway* y una base de datos, donde se gestiona de forma eficiente la información del usuario final. La figura 3.6 lo representa de forma gráfica.



Figura 3.6: Administrador central del sistema

### 3.4.1. Modo de operación

En el administrador central se han implementado una serie de tareas que tienen como finalidad proporcionar fiabilidad a la red y almacenar la información recibida. Para lograr ese objetivo se utilizan las siguientes herramientas: Matlab (analiza los datos matemáticamente), XAMPP (permite conectar con la base de datos MySQL) y Heidi (gestiona la base de datos). A continuación se detallan las acciones que se llevan a cabo en el administrador:

- Se ejecuta un programa en Matlab que permite escuchar todo lo que llega por el puerto de comunicaciones al que está conectado el *gateway*.
- Cuando se recibe un dato se clasifica según la etiqueta establecida previamente (según el tipo de sensor), y se almacena en un vector junto con la hora de recepción.
- Inmediatamente después de la recepción, los datos de temperatura o de humedad captados hasta ese momento, se representan en una gráfica con respecto al momento de captación. Será ahí cuando se pueda detectar algún tipo de anomalía en el comportamiento del sistema.
- Además, tanto si algún dato captado por los sensores está fuera del un intervalo considerado el óptimo, como si algún parámetro radio es tan bajo que pone en peligro la comunicación, saltará una alarma que indicará que el valor está fuera rango óptimo.
- Por último, una vez que se comprueba la fiabilidad de la información analizada, se sube a una base de datos y se clasifica según el tipo de sensor. Esta información es clasificada en una tabla según la fecha de captación.

Cabe añadir que, en este caso la base de datos se encuentra alojada en el servidor local del ordenador administrador (puerto 3306 de la dirección IP 127.0.0.1).

## Capítulo 4

# Validación del sistema y análisis de resultados

La validación del correcto funcionamiento de este sistema inalámbrico IoT se lleva a cabo principalmente en dos partes: En la primera de ellas, se verifica el funcionamiento del usuario final respecto a los datos captados por los sensores, y en la segunda parte, se verifica el funcionamiento del sistema mediante el alcance que puede ofrecer la tecnología LoRA.

### 4.1. Captura de datos de los sensores

En esta sección se van a analizar y verificar los datos de temperatura y humedad provenientes de los sensores, los cuales son útiles para monitorización de cultivos en aplicaciones de *Smart Farming*. A continuación se detalla la función de cada uno.

- La humedad de las hojas de la planta. La utilidad de esta medición reside en determinar en que momentos la planta está realizando la transpiración vegetal. Dicho parámetro es importante para conocer su ciclo vital. Este sensor devuelve el porcentaje de humedad, que debe estar entre 0 % y 100 %, aunque los valores típicos oscilan entre 5 % y 30 % de humedad. En el caso en que se mida algún valor fuera de dicho rango, desde la central de información se avisaría con un *warning* al usuario.
- La temperatura ambiental del entorno en el que se encuentra la planta. Esta medición es útil para verificar en todo momento que la planta se encuentre en un rango de temperatura óptimo, pudiendo considerarse éste entre 15 y 28 grados centígrados. Al igual que en el caso anterior, si se detecta algún dato de temperatura fuera de ese rango saltará un *warning* en el ordenador central.

Dichas mediciones se han realizado en un entorno cerrado, específicamente en el laboratorio del Grupo de Comunicaciones Móviles del instituto iTEAM, estableciendo diferentes ciclos de trabajo en función del tipo de sensor. A continuación, en la figura 4.1 se muestra el montaje del nodo cliente en el momento de la medición de los datos en dicho laboratorio:

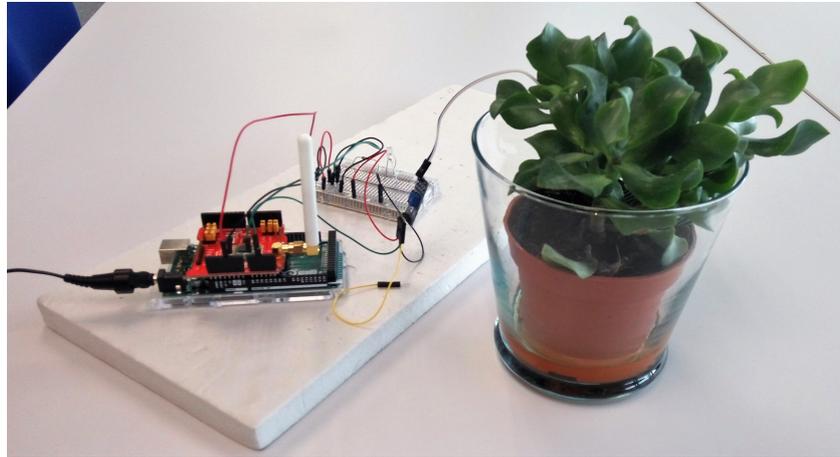


Figura 4.1: Dispositivo del nodo cliente

En referencia a las mediciones de validación, el sistema se dejó en funcionamiento durante todo un fin de semana y se recolectaron los datos, que se pueden visualizar en la figura 4.2.

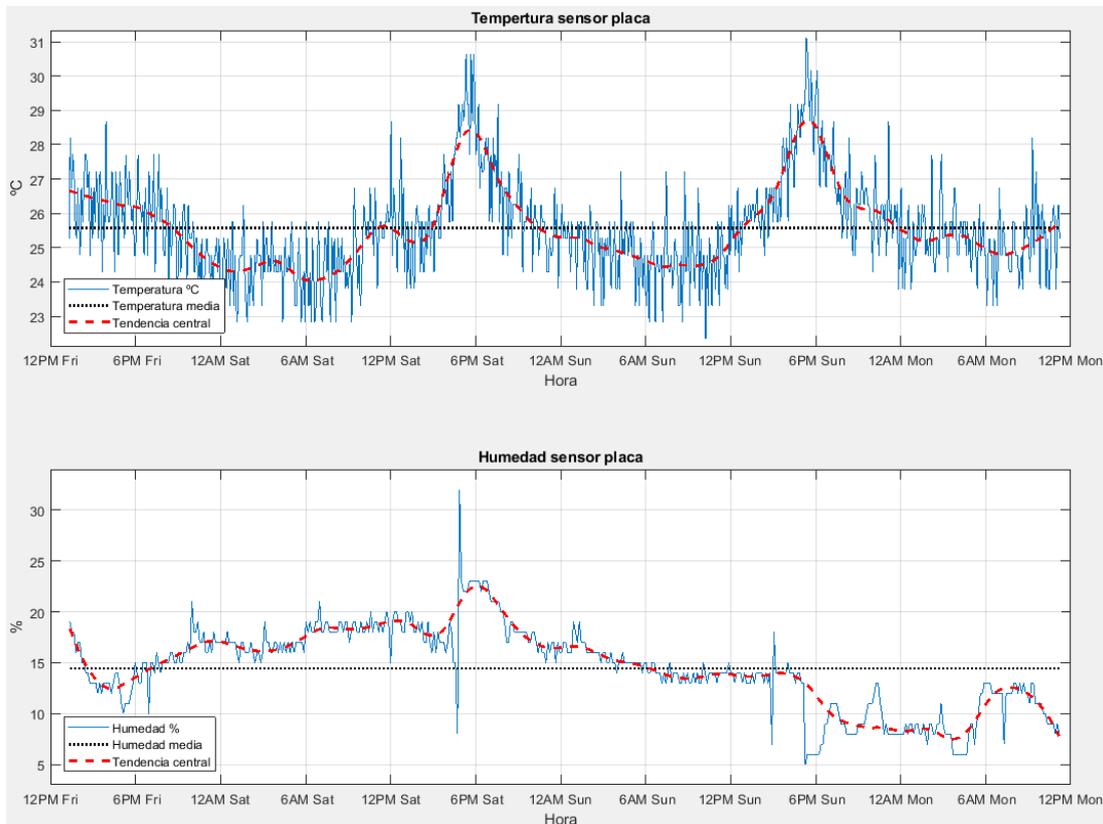


Figura 4.2: Gráfica de medidas de los sensores

Como se puede observar en la gráfica, se proporcionan tres tipos de medidas; la medida instantánea (en color azul), el valor medio (en color negro) y por último la tendencia central que siguen los datos captados (en color rojo). Respecto al valor medio, se obtuvo para la temperatura un valor de 25,6 °C y para la humedad un valor de 14,8 %. No obstante, se observan picos de temperatura que superan los 28 °C los días sábado y domingo a las 6 PM. Con respecto a la humedad, la planta presentaba algunos valores inferiores a los obtenidos en la media, debido a que ésta no se regó durante el fin de semana. Cabe añadir que, la temperatura se midió siguiendo un periodo de captación de 3 minutos, mientras que el de la humedad fue de 8 minutos. Por ese motivo se pueden observar un mayor número de muestras en la gráfica de la temperatura, (aunque el ciclo de trabajo de cada sensor se puede variar en función de los requerimientos del usuario). Por último, también se tuvo en cuenta durante las mediciones el *Packet Error Rate* (PER) del sistema, confirmando que no supera el valor propuesto por el estándar LoRa, (1 %), con lo que se demuestra que estamos ante un sistema fiable [20].

#### 4.1.1. Almacenamiento de la información

Para la gestión de la información de los sensores se ha empleado una base de datos bajo el lenguaje MySQL y gestionada por el programa Heidi. A continuación, en la figura 4.3 se muestra una captura de pantalla de la interfaz del programa.

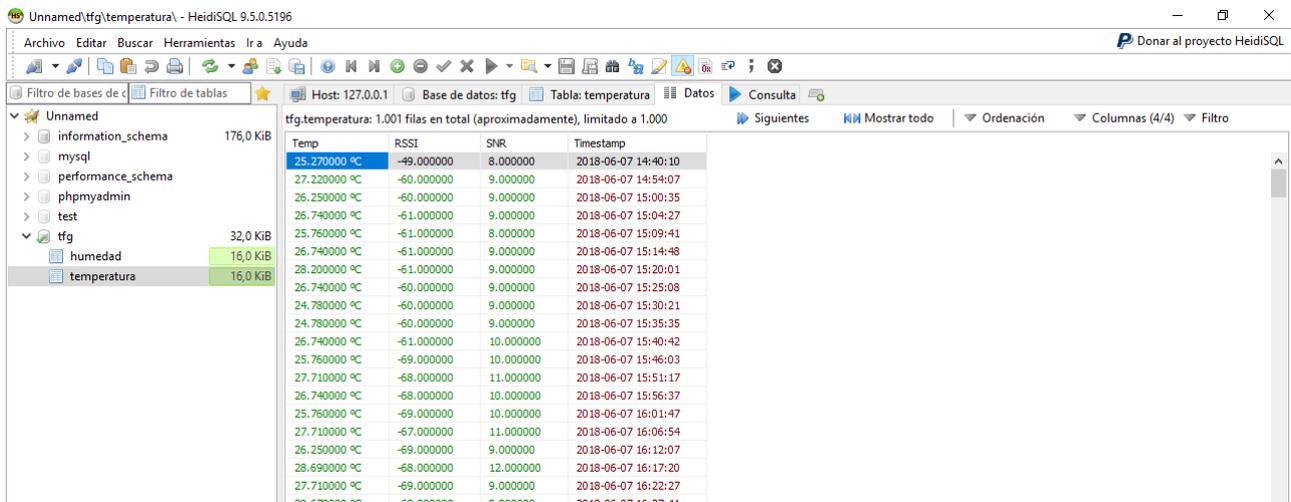


Figura 4.3: Base de datos para la información de temperatura

Como se puede observar en la figura anterior, en la base de datos se almacena concretamente la siguiente información:

- Los parámetros radio para la estimación de canal: RSSI y SNR.
- Información captada por los sensores: temperatura o humedad de las hojas.

- Registro temporal de los datos recibidos en el nodo central.

## 4.2. Cobertura del sistema en un entorno real

En esta sección se va a validar la cobertura y el alcance del sistema en dos tipos de entornos, un entorno urbano y un entorno rural. Para realizar este estudio se han seguido los siguientes pasos:

- Primeramente se establece en el mapa el recorrido en donde van a tener lugar las mediciones.
- En segundo lugar se establece el servidor en un punto fijo, bien sea interior o exterior, pero siempre conectado a un ordenador central, desde el que se pueden analizar los parámetros radio.
- En tercer lugar, para la realización de las medidas, el nodo del cliente se moverá por los puntos asignados previamente y en el servidor se visualizarán y se analizarán los parámetros radio, teniendo en cuenta el punto donde se encuentra el cliente.

Respecto a ubicación, tanto el nodo cliente como el nodo central se han situado aproximadamente a 1 metro de altura sobre el nivel del suelo.

### 4.2.1. Medidas *indoor - outdoor* (I2O) en Campus UPV - Entorno urbano

En este apartado se presentan las medidas de campo realizadas en el recinto de la UPV, ubicando el nodo central en el interior del laboratorio del Grupo de Comunicaciones Móviles (edificio 8G). Las campañas de medidas se han realizado teniendo en cuenta dos modos de transmisión. En el *Caso A*, se realizarán unas medidas utilizando un modo de transmisión que tiene un compromiso bastante equitativo entre el alcance y la capacidad. En el *Caso B* se realizará otra tanda de medidas utilizando otro modo de transmisión que prioriza el alcance y en el que por tanto, teóricamente, el sistema podrá tener una mayor cobertura. De esta manera se podrá comparar la diferencia que existe al transmitir con un modo u otro en lo referente a la cobertura.

#### Caso A

El mapa de la figura 4.4 muestra el recorrido que se ha seguido para hacer las pruebas de campo en entorno urbano, con un modo que tiene una sensibilidad mínima de -129 dBm y cuyo Bit Rate es de 1.0274,20 bps. Específicamente, este modo tiene las siguientes características: *bandwidth* 250 kHz, *Code Rate* 4/5 y *Spreading Factor* 2048 Chips/symbol.

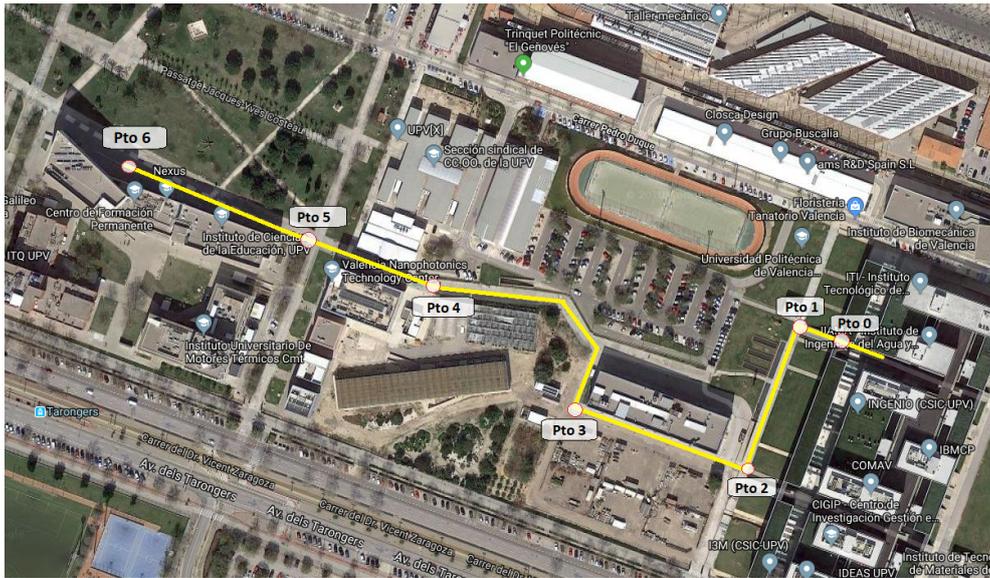


Figura 4.4: Mapa de medidas en Campus UPV, caso A

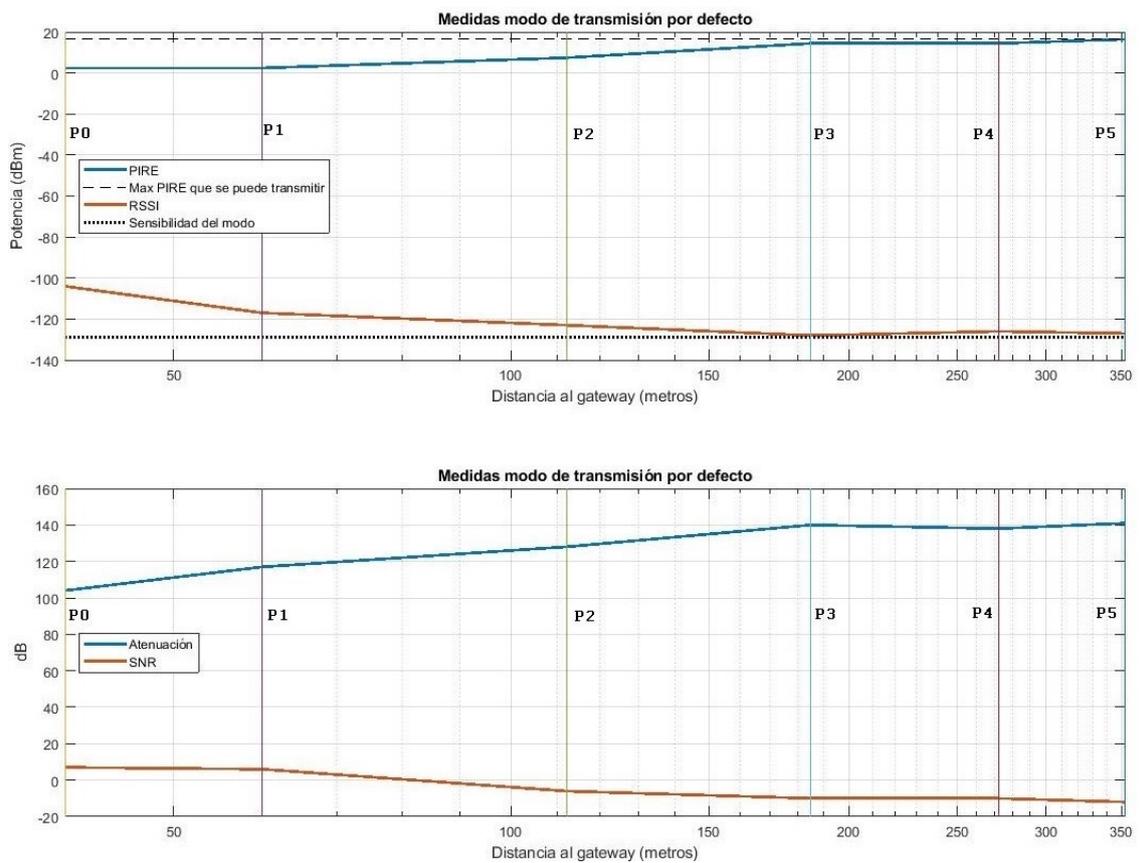


Figura 4.5: Gráfica de las medidas captadas usando el modo por defecto

En la figura 4.4 se muestra el mapa donde se realizó la campaña de medidas y a continuación en la figura 4.5 se pueden observar los resultados obtenidos en forma de gráfica. En cuanto a las medidas

obtenidas, se puede observar que la potencia máxima recibida son -104 dBm, la cual fue captada fuera del edificio donde se ubicaba nodo central, concretamente a 40 metros del mismo (siendo este el punto 0). En ese punto la relación señal a ruido con la que el *gateway* recibía la señal era de 7 dB. Haciendo referencia al punto más lejano desde el que se pudo establecer una conexión estable, este tuvo lugar a 353 metros de distancia del nodo central, siendo la potencia recibida de -127 dBm y la relación señal a ruido de -12 dB (punto 5). El punto 6 del mapa de la figura 4.4 es la primera zona donde la cobertura comienza a ser inestable, por lo que el sistema alcanza una conexión fiable hasta el punto 5.

Posición	Distancia al gateway	PIRE (dBm)	RSSI (dBm)	SNR (dB)	L (dB)
<b>Punto 0</b>	40 m	2.43	-104	7	106.43
<b>Punto 1</b>	60 m	2.43	-117	6	119.43
<b>Punto 2</b>	112 m	7.43	-123	-6	130.43
<b>Punto 3</b>	185 m	14.43	-126	-10	140.43
<b>Punto 4</b>	272 m	14.43	-126	-10	140.43
<b>Punto 5</b>	353 m	16.43	-127	-12	143.43

Cuadro 4.1: Resultados de las medidas para el modo por defecto

Por último, a modo de resumen se presentan las medidas captadas durante toda la campaña en la tabla 4.1. En ésta se muestra de izquierda a derecha los siguientes parámetros: el punto de localización del cliente, la distancia que hay en línea recta desde ese punto hasta el *gateway*, la potencia que el nodo cliente transmite en ese punto tras la estimación del canal, el RSSI (*Received Signal Strength Indicator*) recibido en el *gateway*, la relación señal a ruido en recepción y la atenuación en la transmisión de la señal.

En general, se puede afirmar que los resultados son coherentes, aunque destaca el cambio tan abrupto de PIRE en el punto 3. Esto es debido a que el cliente se encuentra justo detrás de un edificio que tiene una pared de hormigón, lo que provoca una atenuación considerablemente alta. Por ese motivo, tras la estimación de canal en ese punto, el cliente debe aumentar la potencia para poder transmitir sin riesgo a que la conexión se pierda.

## Caso B

El mapa de la figura 4.6 muestra el recorrido que se ha seguido para hacer las pruebas de campo en el mismo entorno que en el Caso A pero con un modo de transmisión más robusto, que permite alcanzar una distancia de cobertura mayor, dada su baja sensibilidad en recepción. En concreto, con este modo se pueden captar señales entrantes con una potencia de hasta -140 dBm y se puede realizar una transmisión de mensajes con un Bit Rate no superior a 122,1 bps. Se trata de una configuración que prioriza de manera notable la cobertura frente a la capacidad, aunque es suficiente para enviar datos

no muy complejos. Si se observan las especificaciones, este modo ofrece las siguientes características: *bandwidth* 62,5 kHz, *Code Rate* 4/6 y *Spreading Factor* 4096 Chips/symbol.



Figura 4.6: Mapa de medidas en Campus UPV, caso B

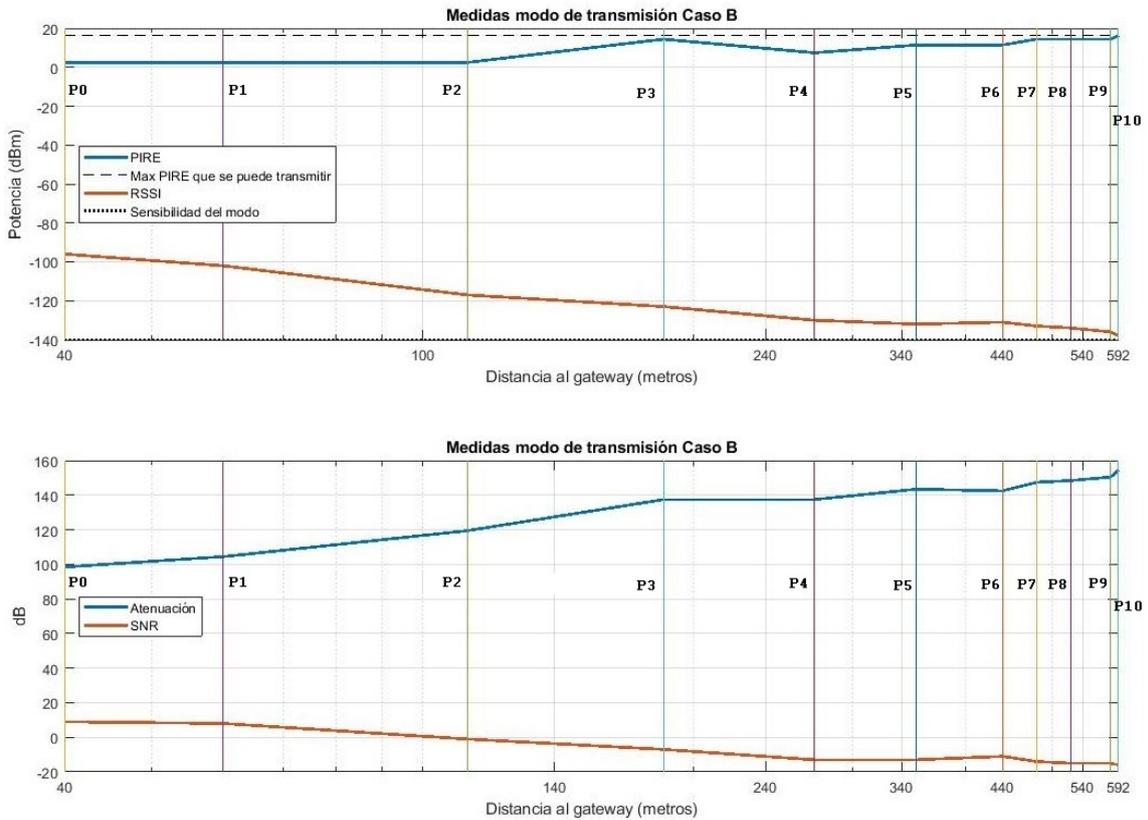


Figura 4.7: Gráfica de las medidas captadas usando el modo del Caso B

El recorrido seguido para realizar las medidas en este caso se puede visualizar en el mapa de la figura 4.6. Seguido de ese mapa, se muestra un análisis gráfico de las medidas captadas, donde se puede visualizar que, con este modo de transmisión se alcanzan casi los 600 metros en cobertura. En ese punto la potencia recibida es de -138 dBm junto con una relación señal a ruido de -16 dB. Con respecto al punto 0, que se trata del más cercano, se puede medir una señal de -96 dBm con una relación señal a ruido de 9 dB. Por lo que si comparamos estos resultados con los que se han obtenido en el Caso A, se confirma que con este modo, se pueden obtener unas mejores prestaciones en referencia a cobertura.

Posición	Distancia al gateway	PIRE (dBm)	RSSI (dBm)	SNR (dB)	L (dB)
<b>Punto 0</b>	40 m	2.43	-96	9	98.43
<b>Punto 1</b>	60 m	2.43	-102	8	104.43
<b>Punto 2</b>	112 m	2.43	-117	-1	119.43
<b>Punto 3</b>	185 m	14.43	-123	-7	137.43
<b>Punto 4</b>	272 m	7.43	-130	-13	137.43
<b>Punto 5</b>	353 m	11.43	-132	-13	143.43
<b>Punto 6</b>	440 m	11.43	-131	-11	142.43
<b>Punto 7</b>	480 m	14.43	-133	-14	147.43
<b>Punto 8</b>	524 m	14.43	-134	-15	148.43
<b>Punto 9</b>	580 m	14.43	-136	-15	150.43
<b>Punto 10</b>	592 m	16.43	-138	-16	154.43

Cuadro 4.2: Resultado de las medidas para el modo Caso B

A modo de resumen se presenta la tabla 4.2, en la que se ofrecen los datos obtenidos para esta tanda de medidas, en la que el *gateway* también estaba ubicado en el laboratorio de comunicaciones móviles del iTEAM dentro del edificio 8G.

Cabe añadir que, de entre los modos programados existe un modo que incluso puede ofrecer una mayor cobertura que el estudiado en este apartado, ya que con él, se pueden realizar detecciones de señales de hasta -142 dBm. El motivo por el que no se ha utilizado ese, se debe a que tan solo tiene una capacidad de 50,8 bps, lo que sería una tasa insuficiente para la mayoría de las aplicaciones IoT.

#### 4.2.2. Medidas *outdoor - outdoor* (O2O) en Huerta de Valencia - Entorno urbano

En este apartado se van a presentar las medidas de campo realizadas en un entorno rural, concretamente en una parte de la huerta al norte de Valencia. El mapa de medidas de muestra en la figura 4.8.



Figura 4.8: Mapa de medidas entorno rural



Figura 4.9: Unidad móvil para las mediciones en un entorno real

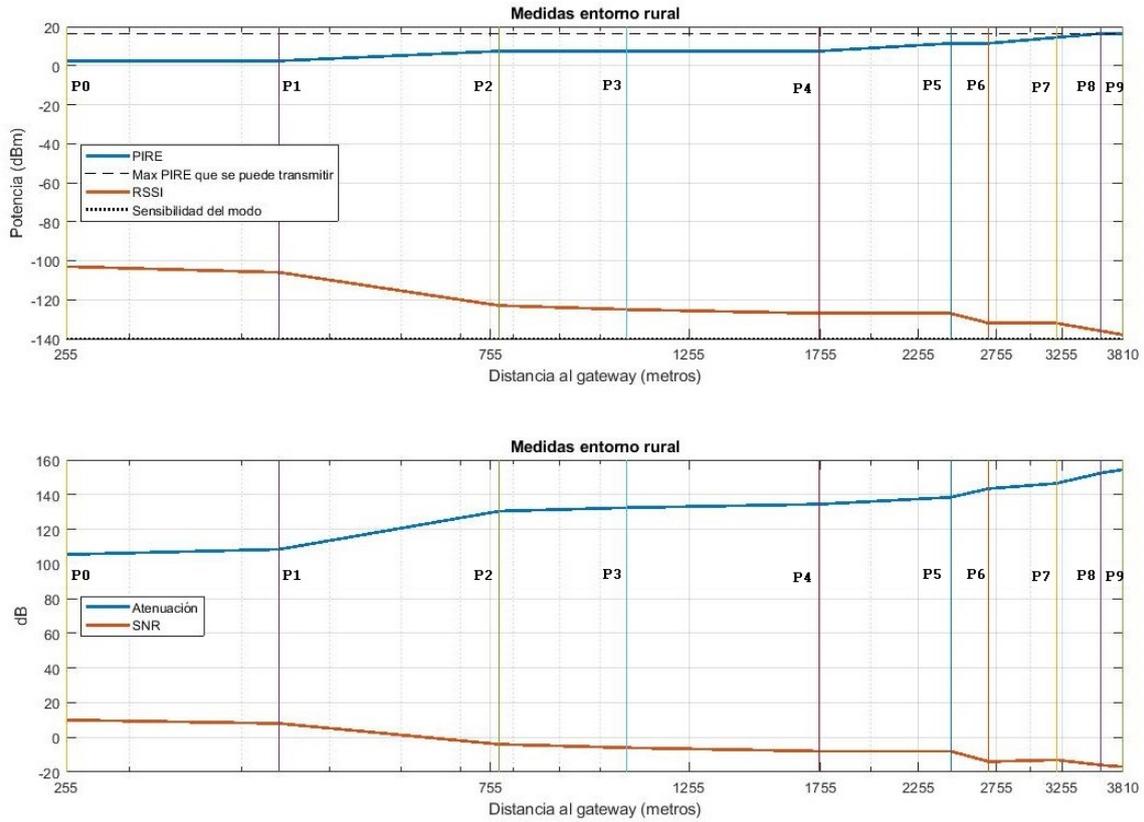


Figura 4.10: Gráfica de las medidas captadas en el entorno rural

Dado que se está realizando un estudio de cobertura, en esta tanda, se ha empleado el modo de transmisión menos limitante en cuestión de alcance, que es el mismo que se ha empleado para el Caso B del apartado anterior. Cabe añadir que al igual que en el resto de mediciones, la climatología ha sido soleada y sin viento, por lo que no habría que tener en cuenta ningún tipo de atenuación por lluvia, niebla... o cualquier otro fenómeno atmosférico.

Haciendo referencia a los resultados obtenidos en las medidas, se ha obtenido un alcance máximo de cobertura (habiendo establecido una conexión estable) de 3800 metros (punto 9). A esta distancia se recibía una señal de -138 dBm con una señal a ruido de -17 dB. Por contra en el punto más cercano, en el que se han realizado mediciones (punto 0 a 255 metros del nodo central), se ha registrado una potencia recibida de -103 dBm, con una señal a ruido de 10 dB. En la siguiente tabla, también se puede observar que aparece el descrito el punto 10, pero no se ha tenido en cuenta en los análisis dado que la conexión que existía era muy débil e inestable.

Posición	Distancia al gateway	PIRE (dBm)	RSSI (dBm)	SNR (dB)	L (dB)
<b>Punto 0</b>	255 m	2.43	-103	10	105.43
<b>Punto 1</b>	440 m	2.43	-106	8	108.43
<b>Punto 2</b>	772 m	7.43	-123	-4	130.43
<b>Punto 3</b>	1070 m	7.43	-125	-6	132.43
<b>Punto 4</b>	1750 m	7.43	-127	-8	134.43
<b>Punto 5</b>	2450 m	11.43	-127	-8	138.43
<b>Punto 6</b>	2700 m	11.43	-132	-14	143.43
<b>Punto 7</b>	3210 m	14.43	-132	-13	146.43
<b>Punto 8</b>	3600 m	16.43	-136	-16	152.43
<b>Punto 9</b>	3810 m	16.43	-138	-17	154.43
<b>Punto 10</b>	4130 m	-	-	-	-

Cuadro 4.3: Resultado de las medidas de campo en entorno rural

Tal y como se puede observar en la tabla 4.3 en un entorno rural el sistema se comporta de una forma completamente diferente a un entorno urbano. El alcance es mucho mayor y las pérdidas de propagación son mucho menores. En este tipo de entorno, el alcance del sistema, alcanza una distancia de 3.8 km, lo que aumenta en 3.3 km aproximadamente, el alcance obtenido en el entorno urbano. También es un dato a tener en cuenta el hecho de que el nodo central en el entorno urbano estaba en *indoor* y en el entorno rural se ha situado al aire libre, conectado tan solo a un ordenador portátil, donde se podían ver y analizar los datos captados.

Como apunte, se puede añadir que en todos los casos la máxima potencia transmitida por el cliente es de 14 dBm (lo cual está limitado por el reglamento de la Unión Europea), aunque la antena posee una ganancia de 3.43 dBi y unas pérdidas por conectores de 1 dB, por lo que la PIRE máxima con la que se puede transmitir es de 16.43 dBm.

### 4.3. Comparación con la plataforma de Libelium

Como ya se ha mencionado en capítulos anteriores, existen otras muchas plataformas propietarias que permiten construir sistemas de Internet of Things. A continuación, se va a presentar un estudio comparativo entre la plataforma diseñada en este proyecto y la plataforma de Libelium (desarrollada por Saul Fuster en un Trabajo Final de Grado realizado en el año 2017, en el mismo grupo de investigación en el que se ha desarrollado este proyecto, Grupo de Comunicaciones Móviles, iTEAM, [21]).

### 4.3.1. Estudio de cobertura

En el estudio de cobertura se van a analizar y comparar los resultados que se obtuvieron con el sistema de la marca Libelium y el sistema en cuestión de este proyecto. En la figura 4.11 las medidas de color azul corresponden con el *sistema low cost* en cuestión y las de color negro con el equipo de Libelium.



Figura 4.11: Mapa de medidas en Campus UPV, sistema Libelium

Las medidas del sistema Libelium de la figura anterior están realizadas con un modo transmisión muy similar al de las medidas de la figura 4.6. Para el estudio comparativo, el sistema propietario utiliza un modo con unas especificaciones de sensibilidad de -137 dBm y el sistema de bajo coste transmite con un modo que admite una sensibilidad de hasta -140 dBm. Por lo que, ambos sistemas transmiten con modos que tienen sensibilidades similares.

Dado que en el entorno de medidas, la ubicación del nodo central y los modos de transmisión, son prácticamente iguales en ambos casos, se puede concluir, en base a las medidas *indoor-outdoor* del entorno urbano que, a distancias iguales, ambos sistemas obtienen unos resultados parejos. Cabe añadir que, la ganancia de la antena tanto transmisora como receptora del sistema de Libelium es de 4.5 dBi y posee una alta eficiencia, mientras que la antena del sistema *low cost* tiene una ganancia de 3.43 dBi y una eficiencia del 51,76 %. Esto significa que las especificaciones de calidad en el sistema de bajo coste son ligeramente inferiores. Con eso y con todo, los resultados son muy similares.

Ambos sistemas se adaptan a la normativa europea, ya que sus potencias de transmisión se puede establecer como máximo a 14 dBm.

### 4.3.2. Estudio económico

Una vez, que se comprueba que ambos sistemas ofrecen las mismas prestaciones respecto a robustez y alcance, se procede a realizar un estudio económico que analiza el coste de los sensores, de las placas y de los transceptores, tanto del nodo central como del nodo cliente. En la tabla 4.4 se muestra el estudio económico comparativo de ambos sistemas:

Dispositivos	Arduino		Libelium	
<b>Nodo cliente</b>				
<b>Placa</b>	Arduino Mega 256	35 €	Waspote SX1272 LoRa module SMA 4.5 dBi - 868	178 €
<b>Transceptor LoRa</b>	LoRa Shield 868 MHz	20 €		
<b>Placa de sensores</b>	No es necesaria	-	Waspote Agriculture Sensor Board PRO	99 €
<b>Sensores</b>	Sensor de temperatura y sensor de humedad	20 €	Sensor de temperatura y sensor de humedad	20 €
<b>Nodo central</b>				
<b>Placa</b>	Arduino UNO	20 €	Waspote Gateway SX1272 LoRa module SMA 4.5 dBi - 868	168 €
<b>Transceptor LoRa</b>	LoRa Shield 868 MHz	20 €		
<b>Total</b>	<b>95 €</b>		<b>465 €</b>	

Cuadro 4.4: Estudio económico comparativo de dos sistemas IoT

Tal y como se puede observar en el anterior cuadro, la plataforma de Libelium necesita al menos una placa con un transceptor [22] y una placa supletoria donde se conectan los sensores [23]. Por su parte, Arduino no utiliza ninguna placa de sensores ya que estos se pueden conectar directamente a los pines analógicos/digitales. El precio de los sensores puede oscilar entre los 2 € y los 30 €, pero se han elegido unos sensores cuya suma alcanza un coste de 20 €. Los sensores que pueden ser perfectamente comunes a ambos sistemas. Haciendo referencia al nodo central, la marca Libelium usa una pasarela que se conecta directamente al ordenador central. Pero, en este proyecto, para el nodo central se utiliza un Arduino UNO (ya que el *gateway* requiere de poca memoria RAM y FLASH) y un transceptor LoRa como el que se utiliza en el nodo del cliente que va conectado físicamente con la placa Arduino [24]. El uso del espectro no encarecería el presupuesto de ningún de los dos proyectos, ya que en ambos casos se está transmitiendo en una banda ISM.

Por tanto, presentados los presupuestos de cada marca y a modo resumen, se puede deducir que el sistema de la marca Libelium cuesta un total de 465 € frente a los 95 € que han costado los elementos de este proyecto. Por lo que, el hecho de haber conseguido un sistema IoT *low cost* con las mismas características que el de la marca propietaria y con un coste de 370 € inferior, confirma la mejora obtenida en este TFG.

## Capítulo 5

# Conclusiones y líneas futuras de trabajo

### 5.1. Conclusiones

Actualmente, Internet de las Cosas es un tema relevante de investigación, para el cual se prevé un despliegue masivo de dispositivos y/o sistemas en el año 2020. Esto conllevará la gestión eficiente de recursos que están presentes en nuestra sociedad, tales como: un coche, una casa, una fábrica o incluso una ciudad, entre otros. En este Trabajo Final de Grado se ha **diseñado e implementado un prototipo de bajo coste** para aplicaciones punto a punto dentro del ecosistema IoT, mediante la comunicación entre un nodo usuario y un nodo central a través de la tecnología LoRa.

Se propone una **arquitectura de cuatro capas**, en la que la capa de percepción está formada por sensores que permiten monitorizar diferentes parámetros, tales como, humedad y temperatura; la capa de red encargada de la comunicación entre el nodo central y el nodo del cliente a través de la tecnología LoRa; la capa de procesado es una solución dual de gestión y almacenamiento de la información entre la herramienta matemática MATLAB y una base de datos MySQL. Por último, la capa de aplicación permite ofrecer servicios de monitorización al usuario final. La red se ha diseñado de manera centralizada, de forma que los clientes transmiten toda la información captada por los sensores a un nodo central que hace de pasarela.

Para la **validación del prototipo IoT**, se realizaron medidas de campo en un **entorno urbano y rural** que permitieron testear el correcto funcionamiento del sistema a través de la comprobación de las capacidades de detección, adquisición, comunicación, almacenamiento y procesamiento de la información del usuario final. En concreto, respecto a las capacidades de detección, adquisición y comunicación del sistema, se realizaron medidas para una aplicación de Smart Farming donde se monitorizaban parámetros importantes para el cultivo de las plantas, tales como: temperatura y humedad de las hojas. Dentro de la campaña de medidas, se pudo comprobar la fiabilidad del sistema, teniendo en cuenta que los datos captados por los sensores eran coherentes. Además, durante las mediciones, se comprobó que

el número de paquetes erróneos era inferior al 1 %, tal y como lo define la norma de estandarización de la tecnología LoRa. Por otro lado, respecto a la comprobación de las capacidades de alcance y robustez de la tecnología LoRa, se llevaron a cabo medidas de campo en un entorno urbano y un entorno rural con diferentes configuraciones.

En primer lugar se realizaron unas **pruebas de campo indoor - outdoor** en un **entorno urbano** con la configuración de un modo que permitía una tasa de transmisión de 1.074,20 bps, soportando una sensibilidad de -129 dBm en recepción. En este caso, con esa configuración y en ese entorno se alcanzó una **distancia máxima de 353 metros**. En segundo lugar, se realizaron otras pruebas de campo en el **mismo entorno (indoor - outdoor)** pero **con una configuración diferente**. En este caso, el modo de transmisión empleado (ancho de banda 62,5 kHz, tasa de codificación 4/6, y factor de ensanchamiento 4096 Chips/symbol), soportaba una sensibilidad de hasta -140 dBm en recepción y permitía una tasa de transmisión de 122,1 bps. Con esa parametrización se llegó a conseguir una **transmisión estable a 592 metros**, lo que ampliaba de forma considerable el alcance, reduciendo la capacidad del envío; aunque la capacidad de este modo es suficiente para enviar tasas de información bajas, como por ejemplo, datos obtenidos en sensores. En tercer y último lugar, se realizó una prueba de campo en un **entorno rural**, concretamente en la huerta situada al norte de Valencia. Dado que el modo utilizado en el caso anterior tenía unas muy buenas características de alcance y unas características de capacidad suficientes para este proyecto se decidió mantener esa **misma configuración**. Por lo que, en ese entorno y con esas características de transmisión se llegó a alcanzar una conexión estable con una separación de **3810 metros**, entre los transceptores.

Si comparamos las medidas realizadas en el entorno urbano (*I2O, indoor to outdoor*) con las llevadas a cabo en un entorno rural (*O2O, outdoor to outdoor*), el alcance del sistema aumenta en un factor 6, respecto a las medidas en urbano, es decir, se lograron alcanzar 3.8 Km.

Con la finalidad de reducir el consumo y de no introducir interferencias innecesarias en la red, en el sistema se **programó una función que estimaba el canal** y era capaz de autorregular la potencia de transmisión, aumentándola o reduciéndola según fuese necesario. Es importante resaltar, que estos resultados se alcanzaron para una altura tanto del nodo usuario como del nodo central de aproximadamente 1 metro sobre el nivel del suelo. Por lo tanto, si consideramos otras configuraciones (nodo central a una altura superior a 1 metro), podríamos tener un mayor alcance del sistema con la tecnología LoRa para aplicaciones de IoT.

En referencia a **los modos programados**, es necesario añadir que el modo con el que más distancia se podría transmitir, puede soportar una sensibilidad de -142 dBm en recepción, pero tan solo permite una tasa de envío de 50,8 bps. Por contra, el modo con más capacidad, permite un envío con una tasa de bit de hasta 37.500,00 bps aunque tan solo soporta una sensibilidad de -112 dBm en recepción. Estos dos últimos, son modos con unas características específicas y no se han tenido en cuenta para las pruebas de campo, ya que no valdrían para una solución IoT genérica.

Este proyecto surgió como continuidad de un trabajo que se desarrolló en el Grupo de Comunicaciones Móviles del iTEAM donde se había programado **un sistema IoT con una solución de la empresa Libelium con un coste aproximado de 465 €** [21]. En este Trabajo Final de Grado se logró desarrollar un sistema que costaba 370 € menos, pudiendo alcanzar unos resultados similares a los obtenidos con el sistema de Libelium. Por lo que, lograr implementar **un sistema IoT con hardware de bajo coste, con un valor aproximado de 95 €**, permite seguir avanzando en el desarrollo de una solución IoT que sea competitiva en el mercado.

## 5.2. Líneas de trabajo futuras

En este proyecto se han creado unas bases muy sólidas para poder construir sobre ellas la totalidad de un sistema IoT. Por ello existen unas líneas de trabajo en las que se podrá seguir avanzando, en el caso en el se consideren viables:

- Instalar un mayor número de nodos de clientes que puedan realizar mediciones en diversos puntos. De ese modo se podría crear un sistema Punto-Multipunto
- Finalizar la aplicación en Android, la cual está en desarrollo. Con esta aplicación se podrá obtener la información de la base de datos y mostrársela al usuario. En este proyecto se ha desarrollado la capa de captación, la capa de red y la capa de procesado, de ahí que uno de los pilares fundamentales de este sistema, sea poder tener una aplicación, con la que el usuario pueda interactuar para conocer los datos captados en el sistema.
- Alojarse la base de datos en un servidor externo para que la aplicación pueda acceder a ella desde cualquier ubicación, siempre y cuando tenga acceso a Internet.
- Conectar de forma directa el nodo central con la Base de Datos del servidor, haciendo uso de una pasarela, la cual podría ser por ejemplo, el Arduino Ethernet o el Arduino Ethernet Shield conectando al actual Arduino UNO. De ese modo, al eliminar un dispositivo en el camino de la información desde los sensores hasta la base de datos, se reducirían las posibilidades de error, la pérdida de datos y los retardos. Con este montaje, tan solo haría falta una central administradora que se hiciera cargo de la capa de negocio, donde se pudieran visualizar las gráficas correspondientes a la optimización del sistema.

# Bibliografía

- [1] UIT-T. *Internet of Things Global Standards Initiative*. Available: <https://www.itu.int/en/ITU-T>.
- [2] UIT-T. “SERIE Y: INFRAESTRUCTURA MUNDIAL DE LA INFORMACION, ASPECTOS DEL PROTOCOLO INTERNET Y REDES DE LA PROXIMA GENERACION”. *UIT-T Y.2060*. 2014. Available at: <https://www.itu.int/rec/T-REC-Y.2060-201206-I>.
- [3] MA Burhanuddin, Ali Abdul-Jabbar Mohammed, Ronizam Ismail, and Halizah Basiron. “Internet of Things Architecture: Current Challenges and Future Direction of Research”. *International Journal of Applied Engineering Research*, 12(21):11055–11061, 2017.
- [4] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. “Internet of Things: A Survey on Enabling Technologies, Protocols and Applications”. *IEEE Communications Surveys & Tutorials*, 17(4):2347–2376, 2015.
- [5] Alvaro Cárdenas. “Top 5 de los Mercados Verticales IoT”. *Secmotiic*, Septiembre 2017. Available at: <https://secmotiic.com/blog/top-5-de-los-mercados-verticales-iot/>.
- [6] ValenciaPort Foundation AFT. “Interoperability of Heterogeneous IoT Platforms”. *Stakeholders and market analysis report*, Mar 2016.
- [7] ETSI. “Internet of Things, Smart Grids”. *Past European Mandate on Smart Grids (M490)*. Available at: <http://www.etsi.org/technologies-clusters/technologies/internet-of-things/smart-grids>.
- [8] ETSI. “Internet of Things, Smart Metering”. *Mandate M/441*. Available at: <http://www.etsi.org/technologies-clusters/technologies/internet-of-things/smart-metering>.
- [9] Alliance for Internet of Things Innovation (AIOTI). *Resources*. Available at: <https://aioti.eu/resources-new/>.
- [10] Nandakishore Kushalnagar, Gabriel Montenegro, and Christian Schumacher. “IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals”. Technical report, 2007.
- [11] Pallavi Sethi and Smruti R Sarangi. “Internet of things: architectures, protocols, and applications”. *Journal of Electrical and Computer Engineering*, 2017, 2017.

- [12] Rafiullah Khan, Sarmad Ullah Khan, Rifaqat Zaheer, and Shahid Khan. “Future internet: the internet of things architecture, possible applications and key challenges”. In *Frontiers of Information Technology (FIT), 2012 10th International Conference on*, pages 257–260. IEEE, 2012.
- [13] Roy Want. “An introduction to RFID technology”. *IEEE pervasive computing*, 5(1):25–33, 2006.
- [14] Roy Want. “Near field communication”. *IEEE Pervasive Computing*, 10(3):4–7, 2011.
- [15] Erina Ferro and Francesco Potorti. “Bluetooth and Wi-Fi wireless protocols: a survey and a comparison”. *IEEE Wireless Communications*, 12(1):12–26, 2005.
- [16] Dae-Young Kim and Seokhoon Kim. “LoRaWAN Technology for Internet of Things”. *Journal of platform technology*, 3(1):3–8, 2015.
- [17] Eduard Ruano Lin. “LoRa protocol. Evaluations, limitations and practical test”. *Universitat Politècnica de Catalunya*, 2016.
- [18] Jonathan de Carvalho Silva, Joel JPC Rodrigues, Antonio M Alberti, Petar Solic, and Andre LL Aquino. “LoRaWAN-A low power WAN protocol for Internet of Things: A review and opportunities”. In *Computer and Energy Science (SpliTech), 2017 2nd International Multidisciplinary Conference on*, pages 1–6. IEEE, 2017.
- [19] LoRa Alliance Technical Committee. “LoRaWAN 1.1 Specification”. *N.SORNIN (Semtech)*, October 11, 2017. Final release.
- [20] Nicolas Ducrot, Dominique Ray, Ahmed Saadani, Olivier Hersent, Gabor Pop, and Guillaume Remond. “LoRa Device Developer Guide”. *Orange Connected Objects & Partnerships in collaboration with Actility*, 2016.
- [21] Saul Fuster Navarro. “Evaluacion de sistemas inalambricos IoT para aplicaciones de Smart Farming”. *Trabajo Final de Grado en Universitat Politècnica de València*, 2017.
- [22] Libelium. *Waspote SX1272 LoRa module SMA 4.5 dBi - 868 MHz*. Available at: <https://www.cooking-hacks.com/waspote-sx1272-lora-sma-4-5-dbi-868-mhz>.
- [23] Libelium. *Waspote Agriculture Sensor Board*. Available at: <https://www.cooking-hacks.com/waspote-agriculture-sensor-board-pro>.
- [24] Dragino. *LoRa Shield for Arduino - 868 Mhz*. Available at: <http://www.dragino.com/products/module/item/102-lora-shield.html>.

## Apéndice A

# Instrumentación nodo cliente

La instrumentación empleada en el nodo cliente consta de un módulo de comunicaciones que permite enviar la información captada por los sensores. Dicho módulo está compuesto por un Arduino MEGA 2560, que se muestra en la figura A.1 y un transceptor que se muestra en la figura A.2.

Este Arduino tiene una velocidad reloj de 16 MHz y destaca entre otras placas por su cantidad de pines disponibles para la conexión de periféricos (54 pines concretamente) y su amplia memoria FLASH, SRAM y EEPROM.

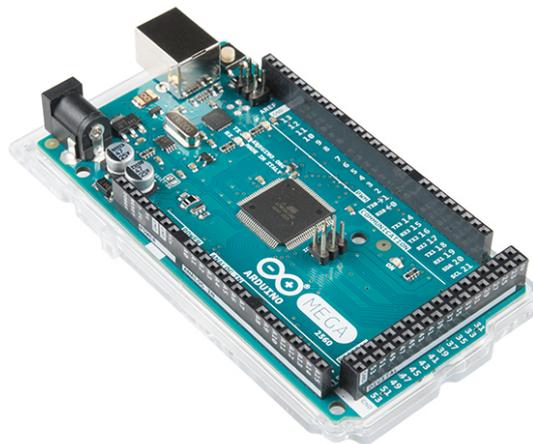


Figura A.1: Arduino MEGA 2560

El transceptor que se muestra en la siguiente figura, es un “escudo” que se coloca físicamente sobre la placa Arduino y permite transmitir con el protocolo LoRa en la banda de frecuencias de 868 MHz. Para ello es necesario una programación previa del Arduino MEGA (figura A.1), mediante el Arduino

IDE haciendo uso de las librerías compatibles con dicho escudo (figura A.2).

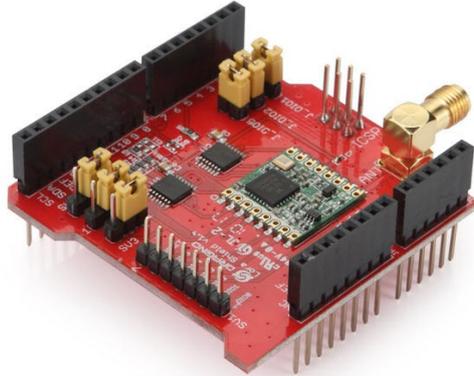


Figura A.2: LoRa Shield - 868 MHz

En cuanto a la captación de datos, se han empleado sensores capaces de medir la temperatura del ambiente y la humedad de las hojas de la planta. Dichos sensores se muestran en la figura B.3 y figura A.4 respectivamente.

Para medir el valor de temperatura de este sensor es necesario seguir los siguientes pasos.

- Primeramente se capta el valor dado por el sensor en la entrada analógica del Arduino y éste se convierte a voltaje mediante la siguiente ecuación:

$$voltage = analogRead(temperaturePin) * \frac{5,0}{1023,0} \quad (A.1)$$

- Después, se debe convertir el valor del voltaje en grados Celsius, haciendo uso de la siguiente ecuación.

$$TempC = (voltage - 0,5) * 100,0 \quad (A.2)$$



Figura A.3: Sensor de temperatura

En cuanto al valor de la humedad de las hojas, se emplea un sensor que se comporta como una resistencia. A mayor cantidad de agua en el sensor, menor voltaje se mide en el Arduino, por lo que

la humedad es inversa al voltaje recibido. Para obtener dicho en porcentaje es necesario emplear la siguiente fórmula:

$$Humedad = \frac{1023,0 * 100,0}{713,0} - \frac{100,0}{713,0} * medida\_analog \quad (A.3)$$

Siendo “medida\_analog” el dato captado por el Arduino en uno de sus pines analógicos al que está conectado el sensor.



Figura A.4: Sensor de humedad

Con toda esa instrumentación el nodo cliente queda de la siguiente manera:

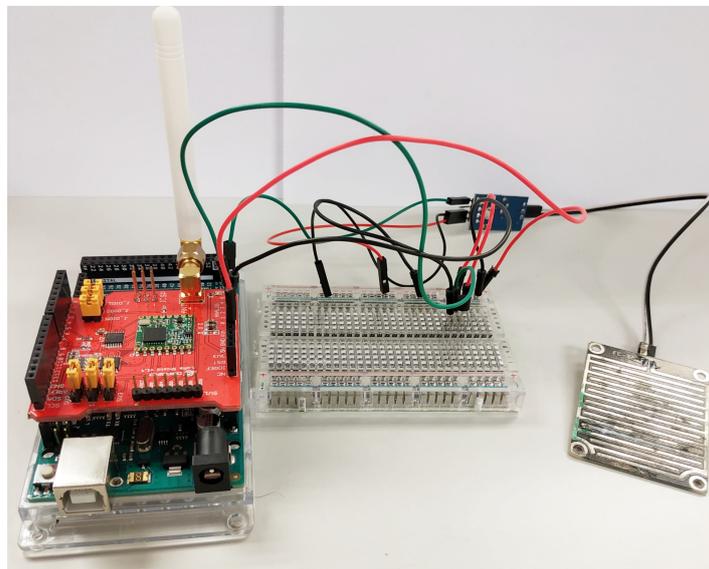


Figura A.5: Sistema completo del nodo cliente

Cabe añadir que en este proyecto se ha alimentado el cliente directamente a la corriente eléctrica, pero existen pilas o incluso mini placas solares que pueden alimentar todo el nodo, para poder instalar en nodo donde se desee sin tener que depender de una red eléctrica.

## Apéndice B

# Instrumentación nodo central

La instrumentación empleada en el nodo central consta de un módulo de comunicaciones, el cual está compuesto por una placa Arduino UNO y un transceptor, figura B.1 y figura B.2 respectivamente.

El Arduino UNO es la base sobre la que se sustenta el transceptor LoRa y donde se almacena todo el código de programación que hace que este nodo funcione como una pasarela con la central de administración.



Figura B.1: Arduino UNO

El transceptor que se muestra en la siguiente figura es el mismo que se utiliza para la comunicación del nodo cliente, por lo que tanto la colocación del mismo como sus especificaciones, son las mismas. Al igual que en el nodo cliente, el código que hace funcionar el transceptor se debe subir al Arduino UNO, mediante el Arduino IDE empleando las librerías compatibles con dicho escudo (figura B.2).

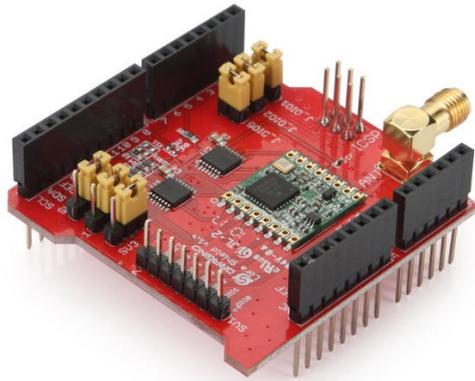


Figura B.2: LoRa Shield - 868 MHz

Con esa instrumentación el nodo central queda de la siguiente manera:

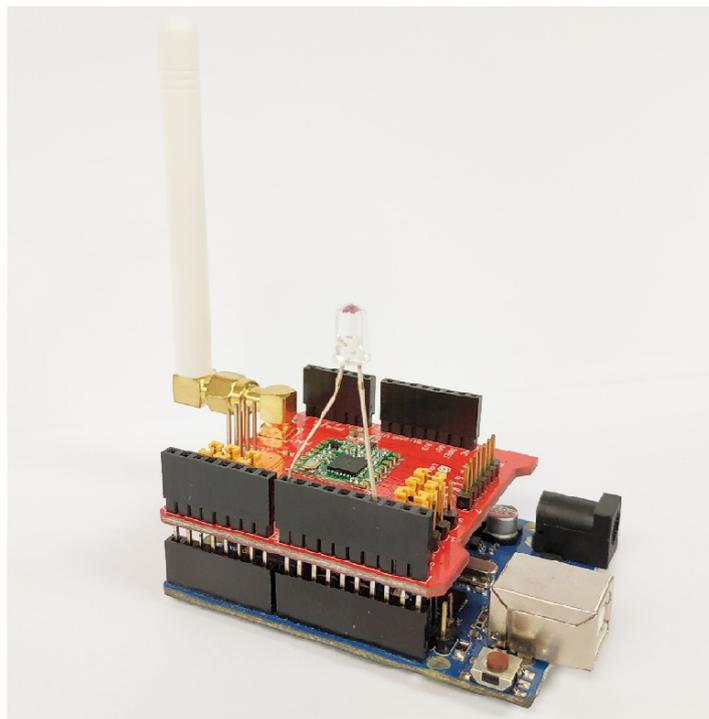


Figura B.3: Sistema completo del nodo cliente

Este nodo debe estar conectado mediante usb a una central que administre la información, por lo que, con ese cable usb también se estaría alimentando el nodo central. Por lo que, en este caso no es necesario emplear una fuente de alimentación externa. También se ha añadido un led que se ilumina cada vez que el nodo central recibe un mensaje.

## Apéndice C

# Parametrización inicial

Inicialmente, en el sistema se deben configurar ciertos parámetros para una correcta transmisión. Estos parámetros son: canales de transmisión, potencia de transmisión, modos de transmisión, tablas de encaminamiento e ID de los tipos de paquetes.

A continuación se procede a explicar en detalle la correcta configuración de cada uno de los parámetros anteriores.

### C.1. Canales y potencia de transmisión

LoRa es una tecnología que trabaja en bandas ISM (siendo la banda de transmisión de este sistema la de 868 MHz). Concretamente, en esta banda el estándar LoRa, fijó 8 canales para la transmisión, los cuales han sido implementados en el Arduino. Dichos canales son los que se muestran en la tabla C.1

<b>Frecuencia central del canal de transmisión (MHz)</b>
865.20
865.50
865.80
866.10
866.40
866.70
867
868

Cuadro C.1: Frecuencias centrales de transmisión para la tecnología LoRa en la banda de 868 MHz

Al ser las anteriores, frecuencias tan cercanas, las diferencias que existen entre sus características de atenuación y alcance son prácticamente inexistentes. El motivo por el que la tecnología ofrece esas 8 posibilidades, es principalmente para tener otras opciones en caso de que un canal este saturado o produzca muchas interferencias.

En cuanto a la potencia de transmisión, tal y como ya se ha mencionado en capítulos anteriores, los niveles programados, fueron 0 dBm, 5 dBm, 9 dBm, 12 dBm, 14dBm. Siendo 0 dBm y 14 dBm los niveles mínimo y máximo de potencia con los que el microchip puede transmitir, la base en la que se fundamentó la elección del resto de niveles fue principalmente el consumo eléctrico que el nodo debía utilizar para la transmisión. Para comprobar el consumo que cada nivel de potencia emplea, se debe acceder al programa LoRa Modem Calculator Tool (figura C.1) e introduciendo el nivel con el que se va a transmitir, se obtiene la electricidad que dicho nivel consume.

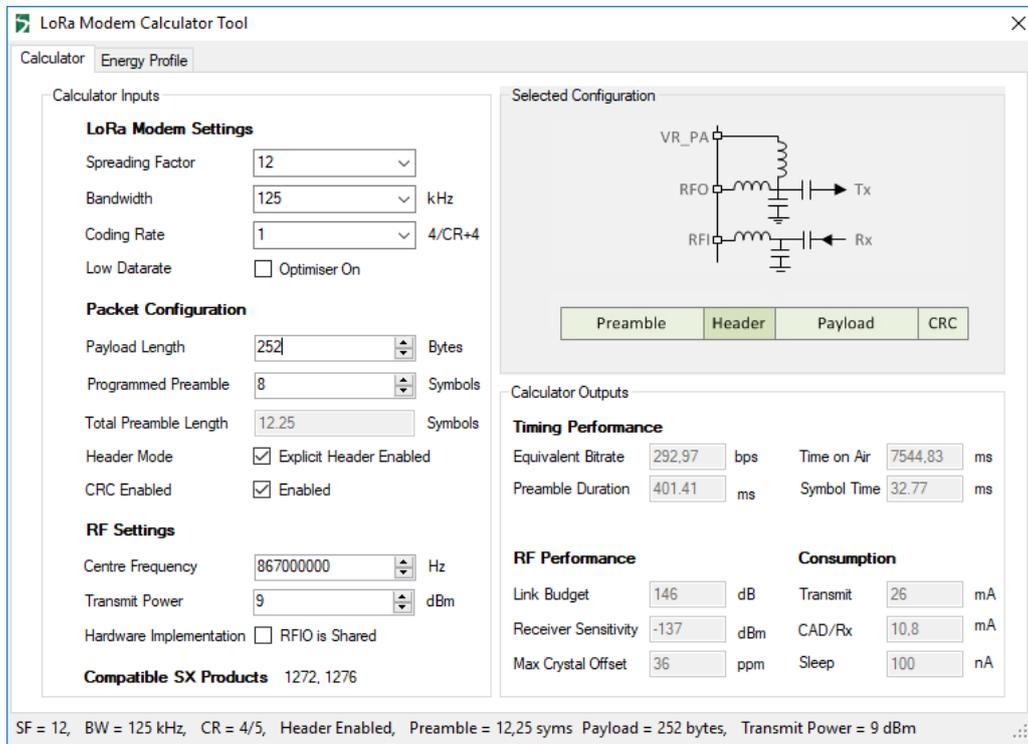


Figura C.1: Interfaz LoRa Calculator

En esa imagen se puede comprobar que en la parte inferior izquierda están los datos de consumo, que varían solamente según la potencia de transmisión que se haya indicado previamente.

## C.2. Modos de transmisión

Los modos de transmisión, son un parámetro fundamental principalmente por dos motivos:

- El modo de transmisión es el que limita la sensibilidad del sistema en recepción.
- El modo de transmisión es el que limita la tasa/capacidad en el envío.

Para diseñar los modos óptimos, tras un estudio previo, se seleccionó un valor para cada parámetro de los que aparecen en el cuadro C.2 y posteriormente se programaron en los registros de la librería RH\_RF95.

<b>Ancho de banda (KHz)</b>	7,8 - 10,4 - 15,6 - 20,8 - 31,25 - 41,7 - 62,5 - 125 - 250 - 500
<b>Coding rate</b>	4/5 - 2/3 - 4/7 - 1/2
<b>Spreading Factor</b>	6 - 7 - 8 - 9 - 10 - 11 - 12

Cuadro C.2: Parámetros para configuración de los modos

Cada uno de los parámetros anteriores representa una característica para la transmisión de datos.

- El ancho de banda es la longitud, medida en Hz, de la extensión de frecuencias en la que se concentra la mayor potencia de la señal. Es necesario añadir que, óptimamente el ancho de banda debe ser cuanto más pequeño posible, para evitar interferir con los demás sistemas de la red. Este parámetro, juega un papel muy importante tanto para la capacidad como para la sensibilidad del modo.
- El coding rate de un código de *Forward Error Correction*, se trata de la proporción de la secuencia de datos que es útil (por tanto no redundante). Es decir, si la tasa de codificación es  $k / n$ , para cada  $k$  bits de información útil, el codificador genera un total de  $n$  bits de datos, de los cuales  $n-k$  son redundantes. Por ello, cuanto más alto sea  $n$  frente a  $k$ , mayor será la robustez de la información enviada. Esto afecta directamente a la capacidad del modo.
- El Spreading factor se conoce como la relación que hay entre la velocidad de chip de un código (siendo este la cantidad de impulsos por segundo a los que se transmite o se recibe el código) y el *symbol rate*. La velocidad del chip es mayor que la velocidad del símbolo, lo que significa que un símbolo está representado por múltiples chips. Cabe añadir que este factor afecta directamente tanto en la capacidad como en la sensibilidad que soporta el modo.

Tras haber analizado todas las combinaciones existentes y haciendo uso de la herramienta “LoRa Modem Calculator Tool” se escogieron finalmente 12 modos, y se incluyeron en los registros correspondientes de la librería RH\_RF95. Para programarlos, era necesario cubrir 3 campos en los registros de memoria:

- El registro de configuración 0x1D.

- El registro de configuración 0x1E.
- El registro de configuración 0x26.

Para realizar dicha tarea se consultaron las especificaciones del fabricante “Semtech” y se generaron los códigos hexadecimales, que se debían escribir en los registros anteriores. Esto se muestra en las tablas: C.3, C.5 y C.7; junto con cuatro ejemplos de entre los doce modos programados en el sistema (tablas: C.4, C.6 y C.8). Los cuatro modos que se muestran para ilustrar lo mencionado son los siguientes:

- Modo 2 - Bw62\_5Cr46Sf4096
- Modo 3 - Bw125Cr45Sf4096
- Modo 5 - Bw250Cr45Sf2048
- Modo 11 - Bw500Cr45Sf128

Cada registro ocupa 8 bits que deben ser cubiertos con 1s o 0s según las especificaciones y posteriormente se debe convertir a hexadecimal dicho número binario.

<b>RegModemConfig1_0x1D</b>
-----------------------------

Pos [7:4] en el registro 0x1D			Pos [3:1] en el registro 0x1D		Pos [0] en el registro 0x1D
BW (kHz)	Value	Value (Hex)	Coding Rate	Value	ImplicidHaderModeOn
7,8	0000	0	4/5	001	0 ->Modo header explícito
10,4	0001	1	2/3	010	1->Modo header implícito
15,6	0010	2	4/7	011	Recomendado explícito
20,8	0011	3	1/2	100	
31,25	0100	4			
41,7	0101	5			
62,5	0110	6			
125	0111	7			
250	1000	8			
500	1001	9			

Cuadro C.3: Especificaciones para el registro de configuración 0x1D

0x1D										
Modo	Pos 7	Pos 6	Pos 5	Pos 4	Pos 3	Pos 2	Pos 1	Pos 0	Valor BIN	Valor HEX
2	0	1	1	0	0	1	0	0	01100100	64
3	0	1	1	1	0	0	1	0	01110010	72
5	1	0	0	0	0	0	1	0	10000010	82
11	1	0	0	1	0	0	1	0	10010010	92

Cuadro C.4: Ejemplos programados para el registro 0x1D

**RegModemConfig2\_0x1E**

Pos [7:4]			Pos [3]	Pos [2]	Pos [1:0]
SF	Value	Value (Hex)	TxContinuousMode	RxPayloadCRcON	SymbTimeout
6	0110	6	0 <- Modo normal de tx	0 <- CRC no habilitado	Timeout para la espera de datos, se recomienda 00
7	0111	7	1 <- Modo continuo de tx	1 <- CRC habilitado	
8	1000	8	Se recomienda modo normal de tx	Se recomienda habilitar	
9	1001	9			
10	1010	A			
11	1011	B			
12	1100	C			

Cuadro C.5: Especificaciones para el registro de configuración 0x1E

0x1E										
Modo	Pos 7	Pos 6	Pos 5	Pos 4	Pos 3	Pos 2	Pos 1	Pos 0	Valor BIN	Valor HEX
2	1	1	0	0	0	1	0	0	11000100	C4
3	1	1	0	0	0	1	0	0	11000100	C4
5	1	0	1	1	0	1	0	0	10110100	B4
11	0	1	1	1	0	1	0	0	01110100	74

Cuadro C.6: Ejemplos programados para el registro 0x1D

**RegModemConfig3\_0x26**

Pos [7:4]	Pos [3]	Pos [2]	Pos [1:0]
Espacio de memoria reservado, se establece a 0000	LowDataRateOptimize	AgcAutoON	Espacio de memoria reservado, se establece a 00
	0 <- no habilitado	0 <- Ajuste de ganancia en el LNA según el registro LNA LnaGain	
	1 <- habilitado. Solo debe ser habilitado cuando exceda los 16 ms el tiempo de símbolo (*)	1 <- Ajuste de ganancia en el LNA por loop interno en el Auto Gain Control (Recomendado)	
	(*) $T_s = (2\hat{S}F)/(BW)$		

Cuadro C.7: Especificaciones para el registro de configuración 0x26

0x26										
Modo	Pos 7	Pos 6	Pos 5	Pos 4	Pos 3	Pos 2	Pos 1	Pos 0	Valor BIN	Valor HEX
2	0	0	0	0	1	1	0	0	00001100	0C
3	0	0	0	0	1	1	0	0	00001100	0C
5	0	0	0	0	0	1	0	0	00000100	04
11	0	0	0	0	0	1	0	0	00000100	04

Cuadro C.8: Ejemplos programados para el registro 0x1D

Por tanto, después de las tablas de especificaciones y de ejemplificación de las especificaciones, se pueden obtener los resultados finales en la tabla C.9, los cuales se deben programar en el registro de la librería correspondiente.

Modos	0x1D	0x1E	0x26	Bit Rate (bps)	Sensibilidad (dBm)
<b>Bw62_5Cr46Sf4096</b>	64	C4	0C	122,1	-140
<b>Bw125Cr45Sf4096</b>	72	C4	0C	293	-137
<b>Bw250Cr45Sf2048</b>	82	B4	04	1074,2	-129
<b>Bw500Cr45Sf128</b>	92	74	04	21875	-117

Cuadro C.9: Registros de los modos de transmisión

### C.3. Tablas de encaminamiento e ID de los paquetes

En el conjunto del sistema se deben enumerar los nodos para un correcto direccionamiento de los paquetes. En este caso al haber tan solo dos nodos (el central y el del nodo cliente), esto no es un gran

problema, pero de cara a una ampliación del sistema este aspecto es muy importante para que no haya datos cruzados. El tamaño del campo donde se escribe la dirección, dentro de la cabecera dispone de 8 bits, por tanto, se podrán direccionar 256 nodos. Actualmente el nodo central tiene la dirección 0x1 y el nodo del cliente 0x2.

Para saber el contenido de cada trama y por tanto para un correcto tratamiento de los datos o una correcta estimación de canal, se establece en la cabecera de cada trama un ID. Éste debe ser conocido tanto en transmisión como en recepción para que no existan confusiones. Actualmente los ID empleados son los que se muestran en la figura C.10.

ID	Tipo de paquete
0x1	Estimación de canal
0x2	Dato de temperatura
0x3	Dato de humedad

Cuadro C.10: ID de los paquetes recibidos

## Apéndice D

# Estimación del canal radio

Con la finalidad tanto de optimizar los recursos a la hora de transmitir, como de introducir las mínimas interferencias en la red, se diseñó para este sistema una función de estimación de canal. Esta función se encarga de regular la potencia de transmisión del cliente, en función de la potencia que el nodo central recibe.

Concretamente, la función de estimación realiza los siguientes pasos:

1. Envía un paquete a máxima potencia, con un ID que indica que se trata de un paquete de estimación.
2. El cliente se pone en modo escucha para poder captar la respuesta del servidor.
3. El servidor al recibir un paquete que según su ID, se trata de uno de estimación, responde con el RSSI medido en la recepción de dicho paquete.
4. Si al cliente no le llega la respuesta después de 10 segundos de la primera transmisión se repite el envío.
5. Si el cliente recibe la respuesta del servidor, almacena el RSSI recibido, para su posterior análisis.
6. Se analiza el RSSI con el que el nodo central ha recibido el paquete de estimación y según sea su valor se establece la potencia en un intervalo u otro para posteriores transmisiones de datos.

Por tanto, cada vez que se llame a la función de estimar canal se realizarán los pasos anteriores, los cuales se muestran en forma de diagrama de flujo en la figura D.1.

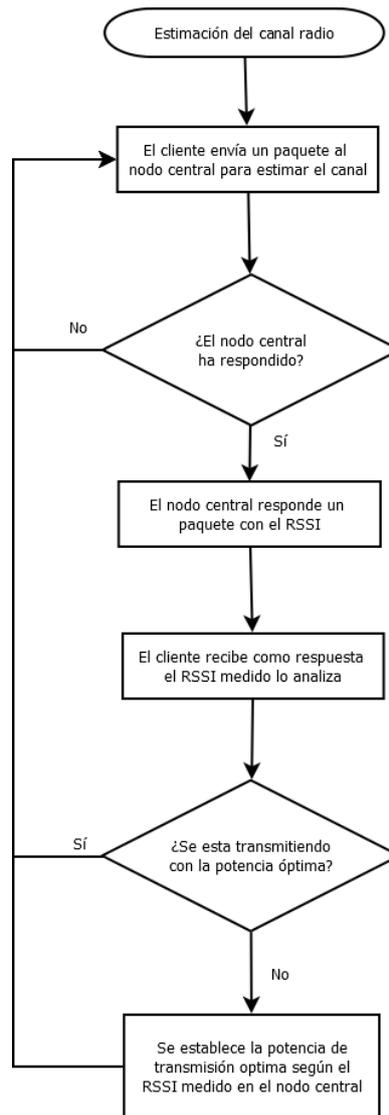


Figura D.1: Función estimación del canal