



Universidad de
Oviedo



ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN.

**MÁSTER UNIVERSITARIO EN INGENIERÍA DE
TELECOMUNICACIÓN**

ÁREA DE TEORÍA DE LA SEÑAL Y COMUNICACIONES

TRABAJO FIN DE MÁSTER

**ESTUDIO Y COMPARATIVA DE REDES IoT DESPLEGADAS
SOBRE LoRA Y 5G: ESTUDIO DE SISTEMAS**

**D. ÁLVAREZ LOZANO, PEDRO
TUTOR: D. RAFAEL GONZÁLEZ AYESTARÁN**

FECHA: JULIO DE 2022



Índice de contenidos

1.- Introducción	6
1.1.- Objetivos y motivación	7
1.2.- Estructura de la memoria	7
2.- Características de LoRa y 5G	9
2.1.- LoRa	9
2.1.1.- ¿Qué es LoRa?	9
2.1.2.- Especificación LoRaWAN	10
2.1.3.- Tipos de nodos finales	11
2.1.4.- Arquitectura de una red LoRa	13
2.1.5.- Formato de la capa física	13
2.1.6.- Formato de la capa MAC	14
2.1.7.- Seguridad en LoRa	15
2.2.- Conexión a redes LoRa	17
2.3.- 5G	20
2.3.1.- ¿Qué es 5G?	20
2.3.2.- Funcionamiento de la tecnología 5G	20
2.3.3.- Especificaciones de la tecnología 5G	21
2.3.4.- Arquitectura de la red 5G	22
2.3.4.1.- Arquitectura LTE/EPC	23
2.3.4.2.- Arquitectura de destino con NR y 5GC independiente	23
2.3.4.3.- Conectividad dual	24
2.3.5.- Seguridad en redes 5G	25
2.4.- Conexión a redes 5G	26
2.5.- Comparativa entre las redes LoRa y 5G	27
3.- Sistema de conexión de red 5G	29
3.1.- Descripción detallada del proyecto	29
3.2.- Componentes del sistema	32
3.3.- Manual de funcionamiento	35
3.3.1.- Configuración del módem 5G	35



3.3.1.1.- Modo QMI (interfaz wwan0).....	37
3.3.1.2.- Modo ECM (interfaz usb0).....	42
3.3.2.- Configuración de la RPi para compartir la conexión a la red 5G mediante WiFi	47
3.3.3.- Configuración de la RPi para compartir la conexión a la red 5G mediante Ethernet.....	54
4.- Pruebas de conexión	59
4.1.- Conectividad de la red WiFi	59
4.2.- Conectividad de la red Ethernet.....	63
5.- Conclusiones.....	66
Referencias.....	68
Anexo: Presupuesto.....	71



Índice de figuras

Figura 2. 1.- Arquitectura del protocolo LoRaWAN.....	11
Figura 2. 2.- Transmisión de un nodo de clase A.	12
Figura 2. 3.- Transmisión de un nodo de clase B.....	12
Figura 2. 4.- Transmisión de un nodo de clase C.....	12
Figura 2. 5.- Arquitectura de LoRa.	13
Figura 2. 6.- Formato de los mensajes de la capa física.....	14
Figura 2. 7.- Mensaje LoRa completo.....	15
Figura 2. 8.- Claves de sesión.	16
Figura 2. 9.- Mapa de Gateways disponibles en The Things Network.	17
Figura 2. 10.- Visualización del Gateway en TTN.	18
Figura 2. 11.- Sensor LHT65 funcionando en TTN.....	19
Figura 2. 12.- Especificaciones de la tecnología 5G.....	22
Figura 2. 13.- Esquema de arquitectura LTE/EPC.....	23
Figura 2. 14.- Esquema de arquitectura NR y 5GC.	24
Figura 2. 15.- Esquema de arquitectura dual.	25
Figura 2. 16.- Pantalla inicial de ThingsBoard.	26
Figura 3. 1.- Esquema del proyecto.	30
Figura 3. 2.- Raspberry Pi Hotspot.	31
Figura 3. 3.- Kit de LabSist de Raspberry Pi 4.	32
Figura 3. 4.- Kit de Queltec para el montaje del adaptador del módulo 5G a USB	
3.0.	33
<i>Figura 3. 5.- Tarjeta SIM de datos.</i>	34
Figura 3. 6.- Módulo 5G montado y con la SIM insertada..	34
<i>Figura 3. 7.- Ejecución del comando \$ifconfig en la terminal.....</i>	36
Figura 3. 8.- Fallo en la conexión a internet con el módulo 5G.....	36
Figura 3. 9.- Ejecución del comando \$lsusb en la terminal.	37
<i>Figura 3. 10.- Modo operativo del módem (QMI).....</i>	38
<i>Figura 3. 11.- Modo operativo de funcionamiento raw_ip (QMI).</i>	39



<i>Figura 3. 12.- Configuración de la APN de la SIM (QMI).</i>	40
<i>Figura 3. 13.- Configuración de udhcpc (QMI).</i>	40
<i>Figura 3. 14.- Comprobación de conexión del módulo 5G (QMI).</i>	41
<i>Figura 3. 15.- Ventana de ayuda de minicom (ECM).</i>	44
<i>Figura 3. 16.- Salir de minicom (ECM).</i>	44
<i>Figura 3. 17.- ifconfig de la interfaz usb0 (ECM).</i>	45
<i>Figura 3. 18.- Uso del driver cdc_ether (ECM).</i>	45
<i>Figura 3. 19.- Comprobación de conexión del módulo 5G (ECM).</i>	45
<i>Figura 3. 20.- Configuración de minicom (ECM).</i>	47
<i>Figura 3. 21.- Configuración del fichero dhcpcd.conf (WiFi).</i>	49
<i>Figura 3. 22.- Configuración del fichero dnsmasq.conf (WiFi).</i>	50
<i>Figura 3. 23.- Configuración del fichero hostapd.conf (WiFi).</i>	51
<i>Figura 3. 24.- Configuración del fichero hostapd (WiFi).</i>	52
<i>Figura 3. 25.- Configuración del fichero interfaces (WiFi).</i>	53
<i>Figura 3. 26.- Configuración del fichero dhcpcd.conf (Ethernet).</i>	55
<i>Figura 3. 27.- Configuración del fichero dnsmasq.conf (Ethernet).</i>	56
<i>Figura 3. 28.- Configuración del fichero sysctl.conf (Ethernet).</i>	57
<i>Figura 3. 29.- Configuración del fichero rc.local (Ethernet).</i>	58
<i>Figura 4. 1.- Conexión a red WiFi.</i>	59
<i>Figura 4. 2.- Test de velocidad de Internet de la red WiFi desde un dispositivo móvil.</i>	60
<i>Figura 4. 3.- Test de velocidad de Internet de la red WiFi desde un ordenador portátil.</i>	61
<i>Figura 4. 4.- Fallo en la conectividad a la red WiFi desde un dispositivo móvil.</i>	62
<i>Figura 4. 5.- Características de la red WiFi desde un dispositivo móvil.</i>	63
<i>Figura 4. 6.- Reconocimiento de la red Ethernet desde un ordenador portátil.</i>	64
<i>Figura 4. 7.- Test de velocidad de Internet de la red Ethernet desde un ordenador portátil.</i>	65



Índice de tablas

Tabla 2. 1.- Comparativa entre 5G y LoRa.	28
<i>Tabla A. 1.-</i> Coste de los recursos humanos.	71
<i>Tabla A. 2.-</i> Coste del hardware y los materiales.	72
<i>Tabla A. 3.-</i> Coste del software y los consumibles.	72
<i>Tabla A. 4.-</i> Coste total del proyecto sin IVA.	73
<i>Tabla A. 5.-</i> Coste final del proyecto.	73



1.- INTRODUCCIÓN

No hay duda de que las tecnologías de comunicaciones se encuentran en proceso de evolución continuo, por lo que a la hora de abordar distintos proyectos pueden surgir dudas a la hora de seleccionar la opción más adecuada de entre todas las posibles. Hay que tener en cuenta que no todas se adaptan a las necesidades de cualquier proyecto pero tampoco todos los dispositivos encajan en todas las redes o escenarios. Por este motivo es muy importante llevar a cabo un estudio previo y valorar lo positivo y lo negativo de las opciones existentes para obtener el mejor resultado posible.

En este documento se va a llevar a cabo un enfoque hacia dos tecnologías de comunicación, las cuales son 5G y LoRa en un despliegue de una red de *Internet of Things* (IoT) mediante diversos sensores y distintos métodos de operación para ver cuál es la mejor opción. Hay que tener en cuenta que el 5G sigue siendo relativamente nuevo y está en proceso de evolución, ya que en un futuro próximo puede ser una tecnología increíblemente poderosa cuando se trata de proyectos de IoT altamente distribuidos en cuanto a rendimiento, capacidad y segmentación virtual, lo cual era difícilmente obtenible con 4G o arquitecturas celulares previas. Por este motivo, para los proyectos de IoT que requieren el transporte inmediato de grandes cantidades de datos, 5G será el rey.

Si embargo, si no se requiere la transmisión de una gran cantidad de datos y sirve con el uso de una tecnología de banda estrecha, LoRa puede ser una opción muy adecuada, ya que, además de ser una tecnología de banda estrecha sin licencia, es muy eficiente en cuanto al consumo de energía, permitiendo transmisiones remotas de forma intermitente en horarios fijados, en lugar de realizar una transmisión continua. Gracias a esto la batería de los sensores LoRa puede durar varios años.

Por este motivo, dependiendo de las necesidades de la implementación que se pretende llevar a cabo, una opción puede ser muy superior a la otra o viceversa.

Una cosa que se debe aclarar de primeras, es que este trabajo no es un proyecto aislado; si no que es parte de un proyecto conjunto que surge como continuación del que ha realizado otra compañera, quien se ha enfocado más en el estudio de las redes y los métodos de conexión a ellas. El título de este otro proyecto es: “Estudio y comparativa de redes IoT desplegadas sobre LoRA y 5G: estudio de redes”.



1.1.- Objetivos y motivación

Uno de los principales objetivos de este proyecto y por el que se ha llevado a cabo ha sido la oportunidad de poder profundizar los conocimientos sobre la nueva generación de tecnología móvil, la cual es 5G; ya que actualmente se está comenzando a explotar y emplear en diversas aplicaciones y proyectos debido a la mejora que supone respecto a las tecnologías previas. Además, con el aumento del número de dispositivos inteligentes en la vida de las personas se buscó un enfoque hacia uno de los sectores de la industria con mayor proyección, el cual es el *Internet of Things*, por eso se ha seleccionado LoRa como tecnología para poder comparar sus prestaciones con 5G, ya que su uso se encuentra muy expandido.

Para llevar a cabo esto, se va a comenzar por un estudio de ambas tecnologías para poder conocerlas en profundidad, teniendo en cuenta todos los conceptos importantes, funcionalidades y métodos de operación; además de un estudio para determinar cómo es la conexión de dispositivos a una de estas redes.

Una vez que se tengan claros todos los conceptos teóricos se va a proceder con la elaboración de sistema de conexión de red 5G en el que se pretende diseñar un dispositivo que actúe como punto de acceso de conexión y permitirá a los dispositivos que se conecten a él poder acceder a la red 5G, ya sea de forma inalámbrica o mediante conexión por cable. Este es el principal objetivo y motivación del proyecto, ya que este tipo de dispositivos en el mercado actual son muy escasos y es muy complicado el poder acceder a ellos, además de muy costoso.

1.2.- Estructura de la memoria

La memoria, como se puede observar en el índice, se ha organizado en torno a 6 capítulos, tomando como primero esta breve introducción para conocer cuál es la finalidad del proyecto y poder describirlo brevemente.

En el segundo capítulo se explicarán con detalle ambas tecnologías (5G y LoRa) para comprenderlas con claridad y detallar sus modos de funcionamiento u operación. También se incluirá el proceso que se debe realizar para conectarse a una de estas redes



para ver la complejidad que esto pueda suponer. Una vez hecho esto, se expondrán los problemas y se buscarán las soluciones necesarias.

Tras tener todos los conceptos comentados previamente claros, en el tercer capítulo se va a detallar la elaboración de la solución propuesta a los problemas encontrados en el capítulo anterior. Además, se va a elaborar un manual para poder poner este sistema a funcionar sin ninguna complicación.

Tras haber desarrollado el sistema de conexión, en el cuarto capítulo se van a realizar varias pruebas de conexión paso a paso y de manera muy detallada para observar los resultados obtenidos.

En el quinto capítulo se va a incluir como anexo los presupuestos sobre todos los materiales, software y horas de trabajo que se han empleado en el proyecto.

Con el objetivo de cerrar el proyecto adecuadamente, en el sexto capítulo se realizarán las conclusiones sobre todo el proyecto, que van a funcionar a modo de recapitulación de las cosas más importantes y a tener en cuenta, además de los usos futuros o posibles expansiones del proyecto.

Por último, se muestran las referencias y bibliografía utilizadas para obtener más información sobre los conceptos teóricos necesarios para el desarrollo del proyecto.



2.- CARACTERÍSTICAS DE LoRa Y 5G

Para comprender claramente el desarrollo del proyecto, además de la información que se va a presentar en él, en este capítulo se va a empezar explicando las principales características y conceptos relacionados con el funcionamiento y la conexión a redes de ambas tecnologías; comenzando por LoRa y terminando por 5G. Esta sección servirá para sentar las bases del proyecto y determinar las necesidades del sistema de conexión de red 5G que se presentará en el siguiente capítulo.

2.1.- LoRa

Tal y como se acaba de decir, se va a comenzar con una descripción de la tecnología LoRa para dar todos los datos necesarios y poder determinar correctamente su funcionamiento.

2.1.1.- ¿Qué es LoRa?

Long Range (LoRa) es una tecnología de comunicación inalámbrica que emplea una modulación de espectro expandido usando la técnica *Chirp Spread Spectrum* (CSS), la cual consiste en modular el mensaje por medio de una señal que varía continuamente en frecuencia. Fue patentada por la empresa Semtech con el objetivo de tener un consumo de energía muy bajo, además de un largo alcance efectivo. [1] [2]

Estas características hacen que LoRa sea una tecnología ideal en redes IoT que requieran sensores alimentados mediante baterías y que pretendan tener un amplio ciclo de vida pudiendo llegar hasta los 10 años. Está categorizada dentro del tipo de redes *Low-Power Wide Area Network* (LPWAN) o redes de bajo consumo y área extensa, cuyas comunicaciones se encuentran por debajo del orden de los GHz y presentan una tasa de datos desde 0.3 kbps a 50 kbps dependiendo del rango y la duración del mensaje. Las distancias de transmisión pueden alcanzar hasta los 20 km.



2.1.2.- Especificación LoRaWAN

Es importante saber que LoRa es únicamente la tecnología de modulación que se emplea en un protocolo de red llamado *Low power Wide Area Network* (LPWAN) o como también es comúnmente conocido: LoRaWAN. Estas redes son las que permiten que los dispositivos IOT conectados intercambien pequeñas cantidades de datos a baja velocidad con un largo alcance y bajo consumo de energía. [3][4]

Este protocolo presenta una topología en estrella con los gateways enlazando los nodos finales con un servidor. Esta conexión entre gateway y servidor se realiza mediante IP gracias a tecnologías como Ethernet, WiFi o 3G principalmente; en cambio, la conexión entre los gateways y los nodos finales se lleva a cabo mediante LoRa. La tecnología puede ser single hop, cuando los nodos no se comunican entre ellos, o FSK (modulación por desplazamiento de frecuencia).

La comunicación es bidireccional en todo momento, a pesar de que lo habitual suele ser que los nodos finales manden mensajes al servidor, también puede resultar interesante para alguna aplicación en particular dar órdenes a los nodos para realizar una función determinada. Los mensajes que van del servidor a los nodos se les conoce como *uplink*.

Las frecuencias que se emplean en la comunicación entre los nodos y gateways pueden ser bastante variadas y con diferentes tasas de datos, aunque lo habitual en Europa es emplear la banda de frecuencias ISM; es decir, entre 863 y 870 MHz. Una cosa importante teniendo en cuenta la tasa de datos, sabiendo que el rango de variación es el que se mencionó en el apartado anterior, es que cada nodo se puede ajustar de manera independiente mediante una tasa de datos adaptativa (ADR).

Los nodos pueden transmitir datos a cualquier frecuencia disponible la tasa que se requiera en esa aplicación, teniendo en cuenta que se han de respetar las reglas de máximo ciclo de trabajo, máxima duración de transmisión y los cambios pseudoaleatorios del canal que se van realizando de manera automática para mejorar la robustez frente a interferencias.

A continuación, se muestra una imagen sobre la arquitectura del protocolo LoRaWAN para poder comprender de una mejor manera las partes que la componen:

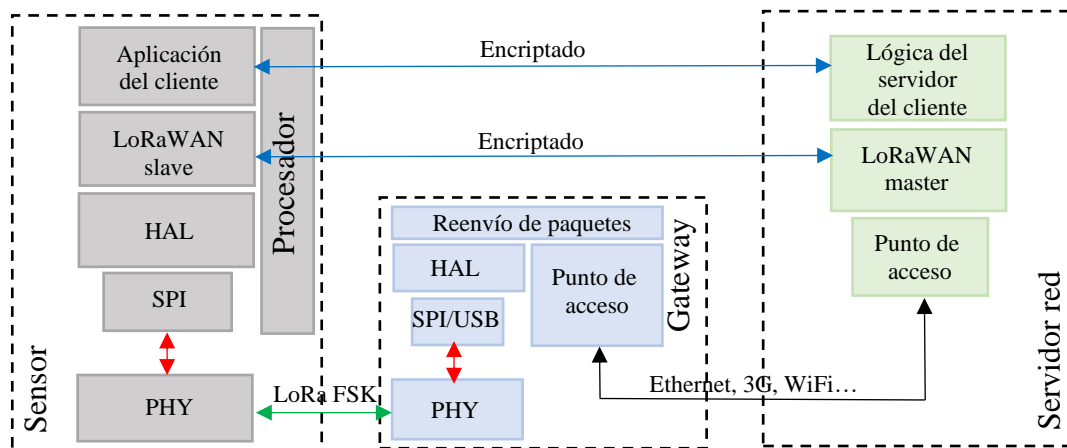


Figura 2. 1.- Arquitectura del protocolo LoRaWAN.

En esta figura se observan los elementos y conexiones de la arquitectura que se han comentado previamente: sensor, gateway y servidor. Además, en este caso en concreto se está empleando un método de funcionamiento *master/slave* (maestro/esclavo) en el que el sensor únicamente recibe las órdenes del servidor y le envía los datos que este solicita. También se puede observar que toda la información va encriptada, esto se explicará más adelante cuando se haga referencia a la seguridad del protocolo.

2.1.3.- Tipos de nodos finales

Los nodos finales de esta tecnología se pueden clasificar en tres clases, las cuales van a satisfacer las diferentes necesidades que pueda tener el proyecto que se esté llevando a cabo. Cada clase va a depender de una relación directa entre la latencia de comunicación del enlace descendente y la duración de la batería. [3][5]

- Clase A (*All*): esta clase, además de bidireccional, es completamente asíncrona, en ella el enlace ascendente (*uplink*) tiene un único slot para mientras que el enlace descendente (*downlink*) tiene dos. El *uplink* está programado por los dispositivos finales en función de sus necesidades. Tiene una alta probabilidad de colisiones si un nodo está transmitiendo y otro se inicia y decide transmitir en el mismo canal de frecuencia y con una configuración similar.

El *downlink* únicamente tiene lugar en los dos slots de recepción en un tiempo específico y justo después de que se envíe un paquete desde el nodo final, este se pone en modo escucha.

Esta es la clase que genera un mayor ahorro energético y es la más empleada cuando se necesitan comunicaciones esporádicas.

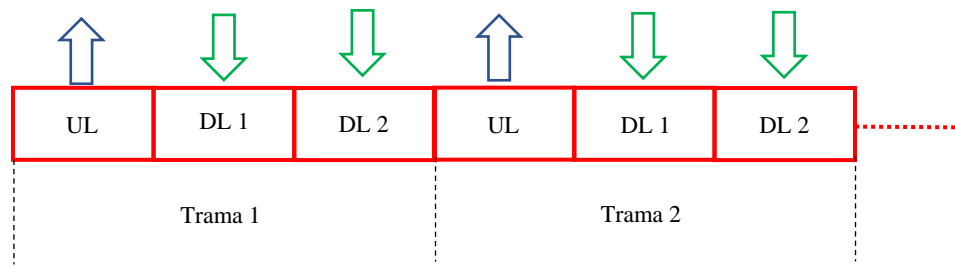


Figura 2. 2.- Transmisión de un nodo de clase A.

- Clase B (*Beacon*): también conocida como bidireccional programada, se caracteriza porque tiene slots de recepción aleatorios, ya que los de tipo downlink se planifican justo después de un uplink, dándoles un número determinado de ventanas de recepción y la duración de la escucha. Para empezar el downlink en un tiempo específico, el dispositivo recibe una señal de baliza sincronizada desde el gateway.

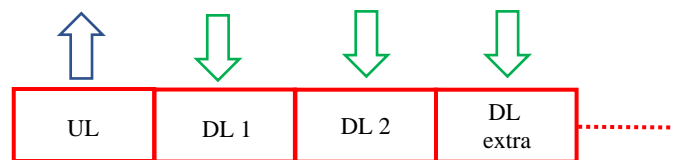


Figura 2. 3.- Transmisión de un nodo de clase B.

- Clase C (*Continuous*): bidireccional con un máximo de ventanas de recepción. Esta clase de dispositivos siempre están listos para recibir una transmisión, ya que siempre están escuchando a menos que se quiera transmitir. Debido a esto, esta es la clase que más energía consume de las tres, generando un alto rendimiento sin ahorro de batería. Sin embargo, estos son muy empleados en aplicaciones que demanden manejo del tráfico de datos en tiempo real.

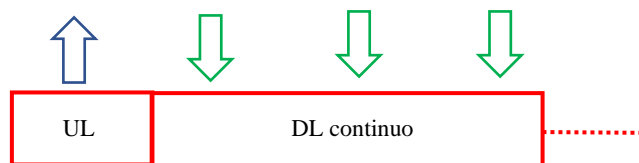


Figura 2. 4.- Transmisión de un nodo de clase C.

2.1.4.- Arquitectura de una red LoRa

La arquitectura de las redes LoRa se puede dividir generalmente en dos capas distintas entre sí, las cuales son: [3][4]

1. Una capa física empleando la técnica de modulación CSS.
2. Una capa basada en el protocolo MAC, a pesar de que en las comunicaciones LoRa el sistema implica una arquitectura de red de acceso específica.

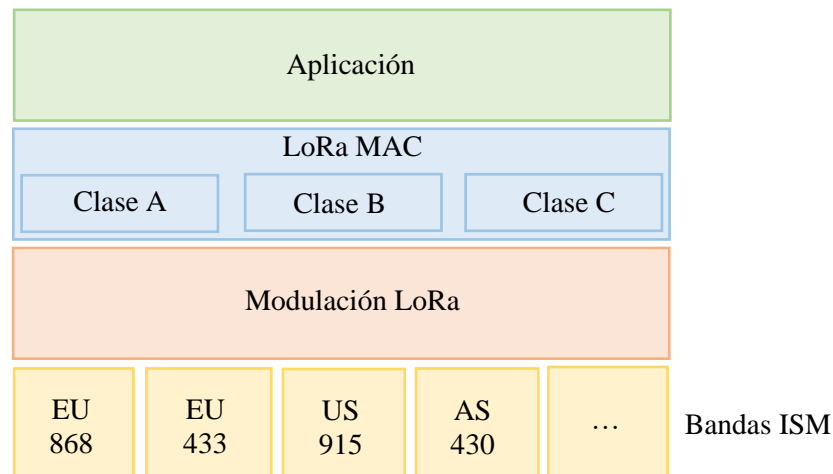


Figura 2. 5.- Arquitectura de LoRa.

Muchas de las redes existentes en la actualidad emplean una arquitectura de tipo *mesh*, lo que quiere decir que los nodos finales de manera individual reenvían la información de otros nodos para aumentar el rango de comunicación y, a la vez, el tamaño de celda. El problema es que esto supone también un aumento de la complejidad, reduce la capacidad de la red y disminuye la duración de la batería a medida que los nodos reciben y reenvían información de otros nodos que probablemente sea irrelevante para ellos. La mejor opción se basa en emplear una arquitectura en estrella si se busca preservar la batería cuando se quiere emplear en aplicaciones de largo alcance.

2.1.5.- Formato de la capa física

En la capa física de una red WAN LoRa, se puede encontrar un formato de trama para encapsular la carga de datos y añadir toda la información de contexto de la capa física. Este formato normalmente se compone de un preámbulo y los datos, aunque puede variar dependiendo de dos campos de trama adicionales que son opcionales; los cuales son: cabecera y CRC. [6]



- **Preámbulo:** El preámbulo comienza con una secuencia de pulsos constantes que cubren toda la banda de frecuencias de un determinado ancho de banda. Los dos últimos pulsos de este preámbulo codifican la palabra de sincronización, la cual es 1 byte (8 bits) que se utiliza para emparejar las redes LoRa que emplean la misma banda de frecuencia. Para ello, los dispositivos configurados con una determinada palabra de sincronización dejarán de escuchar una transmisión si la palabra de sincronización decodificada no coincide con la que se había configurado (si no tienen la misma palabra de sincronización). Esta palabra de sincronización va seguida de dos pulsos junto con un cuarto adicional, con una duración de 2,25 símbolos. Es importante comentar que la duración total de este preámbulo puede configurarse en un rango de entre 10,25 y 65.539,25 símbolos.
- **Cabecera opcional:** Después del preámbulo tenemos la posibilidad de añadir una cabecera opcional que se transmite con una tasa de codificación de 4/8 (si la tasa de codificación es k/n , por cada k bits de información útil, el codificador genera un total de n bits de datos, de los cuales $n-k$ son redundantes). Eso dice el tamaño de la carga útil (en bytes), la tasa de codificación que se va a emplear para el final de la transmisión y si hay o no un CRC de 16 bits para la carga útil al final de la trama. Además, la se incluye un CRC para poder descartar paquetes con cabeceras no válidas al receptor. Para almacenar el tamaño de la carga útil se utiliza 1 byte, por lo que el límite de la carga útil es de 255 bytes. La cabecera es opcional para poder desactivarla en situaciones en las que no es necesaria.

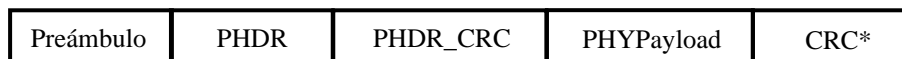


Figura 2. 6.- Formato de los mensajes de la capa física.

Es importante comentar que esta figura hace referencia a los mensajes de tipo *uplink*, ya que en los de tipo *downlink* no es necesario el CRC final.

2.1.6.- Formato de la capa MAC

En LoRaWAN, la capa de control de acceso al medio (*Medium Access Control* o MAC) sirve para gestionar todas las peticiones entrantes de transmisión de los dispositivos finales de nuestra red, permitiéndoles comunicarse a través de la modulación de LoRa. Esta capa utiliza las bandas ISM y gracias a esto se pueden crear redes IoT públicas o privadas de forma accesible, en cualquier lugar y siempre cuando se emplee el hardware y el software necesario para ello. [7]



El protocolo y la arquitectura de red de LoRaWAN determinan de forma precisa la duración de la batería de un nodo, la capacidad de la red, la calidad del servicio, la seguridad y la variedad de aplicaciones de red. A diferencia de LoRaWAN, LoRa no tiene reglas ni normas más allá de las limitaciones de la capa física ofrecida por los dispositivos o las impuestas por la librería utilizada.

Se diseñó para redes de sensores en las que todos ellos intercambiarían paquetes con una baja tasa de datos en largos intervalos de tiempo, como transmitir la temperatura de un acuario una vez al día.

Por tanto, resulta a su vez importante observar cómo es el formato de los mensajes en esta capa, tal y como se ha hecho en el apartado anterior para el caso de la capa física. Para ello se muestra a continuación una imagen en la que se puede ver cómo se descomponen en capas estos mensajes:

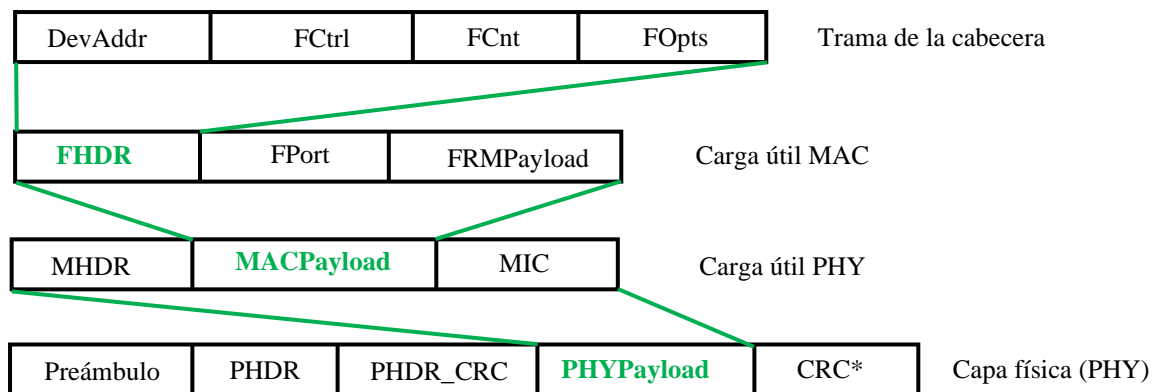


Figura 2. 7.- Mensaje LoRa completo.

La PHYPayload, también mostrada en el apartado anterior, se descompone en una nueva estructura, la cual se compone de encabezamiento (MHDR), MACPayload y código de integridad del mensaje (MIC).

La MACPayload a su vez se compone de encabezamiento (FHDR), puerto (FPort) y payload (FRMPayload). Esta contiene los elementos de la parte superior de la imagen; los cuales son: dirección del dispositivo (DevAddr), bits de control (FCtrl y FCnt) y opciones (FOpts).

2.1.7.- Seguridad en LoRa

En algunos escenarios, la seguridad es indispensable y necesaria teniendo en cuenta que se pueden tratar datos de toda una ciudad por ejemplo. Por ello, LoRa e

indistintamente LoRaWAN incorporan varias capas de cifrado, con sus correspondientes claves, que hacen uso del algoritmo de cifrado AES128 para proteger las comunicaciones. Estas capas de cifrado son: [3][4]

- Network Session Key: clave de 128 bits que garantiza la seguridad a nivel de red. Se utiliza para la interacción entre el nodo y el servidor de red. Esta clave sirve para validar la integridad de cada mensaje mediante su código de integridad del mensaje (MIC). Este MIC es similar a una suma de comprobación, salvo que impide la manipulación intencionada del mensaje. Para ello, LoRaWAN utiliza AES-CMAC.
- Application Session Key: clave de 128 bits que garantiza la seguridad a nivel de aplicación extremo a extremo (*end-to-end*). Se utiliza para el cifrado y descifrado de los datos. Estos datos están totalmente encriptados entre el nodo y el servidor. Esto significa que nadie más puede leer el contenido de los mensajes que se envíen o reciban.
- Application Key: clave de 128 bits que tiene una elevada seguridad a nivel de aplicación, se usa sólo en OTAA (*Over-The-Air-Activation*). Esta clave sólo es conocida por el dispositivo y la propia aplicación.

A continuación, se muestra una imagen a cerca de todas estas claves de sesión empleadas en LoRa:

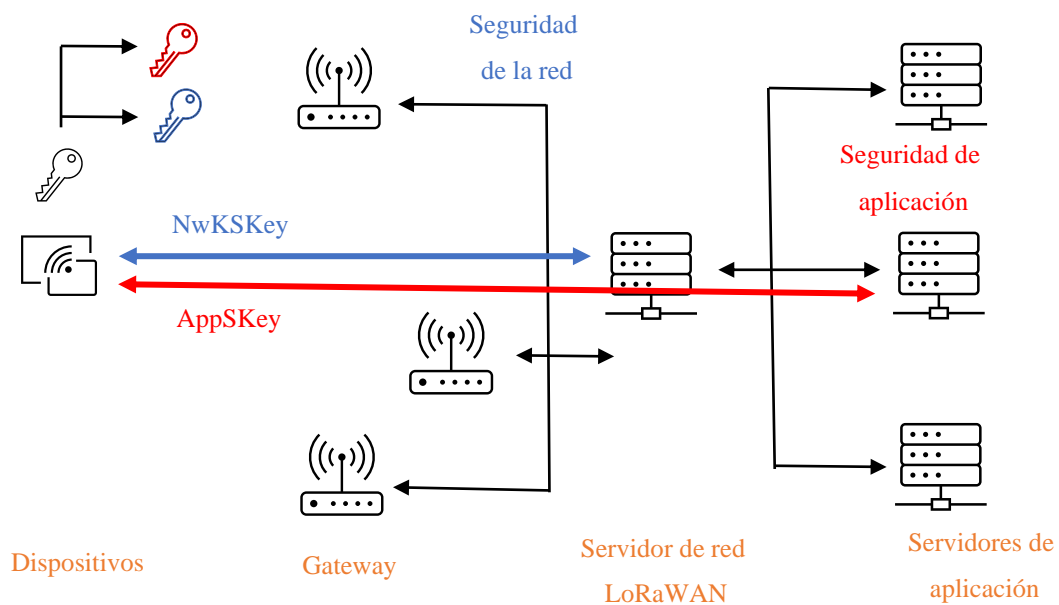


Figura 2. 8.- Claves de sesión.

2.2.- Conexión a redes LoRa

Tras haber comentado los principios básicos de esta tecnología, se va a proseguir con la explicación para poder realizar conexiones a una red LoRa mediante un despliegue IoT. Para ello, lo primero es comentar cómo se realiza el despliegue IoT y luego, cuál es el método seguido para conectarse. No se va a entrar mucho en detalle en temas de configuración, ya que eso se ha realizado en el otro proyecto, el cual se ha mencionado previamente. [8]

En primer lugar, el método seguido para conectarse a la red LoRa es gracias a la plataforma “The Things Network” (TTN), ya que esta proporciona el servidor LoRa de código abierto más utilizado. Esta web permite utilizar uno de los Gateways disponibles a lo largo del mundo, buscando uno que se encuentre en una ubicación cercana a la zona donde se esté realizando el proyecto, aunque también existe la opción de implementar un Gateway propio.

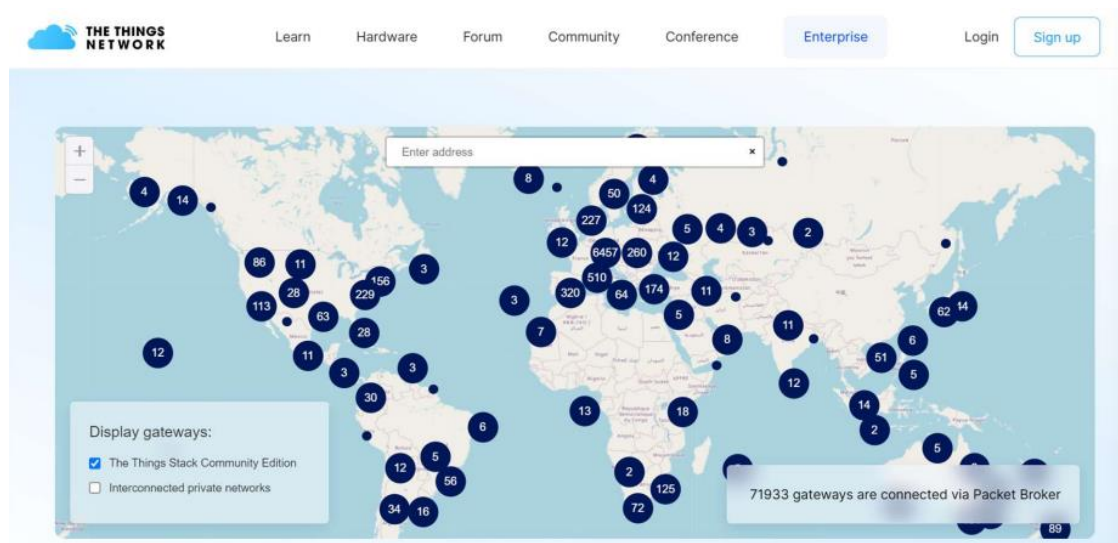


Figura 2. 9.- Mapa de Gateways disponibles en The Things Network. [8]

Para registrar el Gateway, tras haberse registrado en la web de TTN, hay que acceder a la consola para registrar los elementos que van a conformar la red, ya sea tanto el propio Gateway, como los dispositivos finales o nodos que se van a emplear. Este proceso es bastante sencillo y está bien guiado por la web.

El Gateway empleado es el “Mikrotik Routerboard LoRa8 kit”, al que se ha conectado una antena compatible mediante Ethernet; a su vez, el Gateway se conecta mediante

Ethernet a la red del laboratorio para transmitir los datos. Nada más que la conexión es correcta, el dispositivo crea una red inalámbrica a la que se puede acceder desde cualquier navegador mediante: <http://192.168.88.1>. Desde esta url se va a poder realizar toda la configuración del Gateway, asignándole su dirección IP, puerta de enlace y DNS, de LoRa y de los elementos que se quieran añadir a la red. Hay que tener en cuenta que si se modifica la IP, para acceder a él posteriormente se va a tener que poner en la url previa esta nueva IP en vez de la anterior. También este punto es importante anotar el identificador del Gateway, ya que va a ser necesario para registrarlo en TTN cuando se acabe la configuración.

En la siguiente imagen se puede mostrar cómo se verá por pantalla la información recibida por pantalla tras haber realizado una buena configuración del dispositivo:

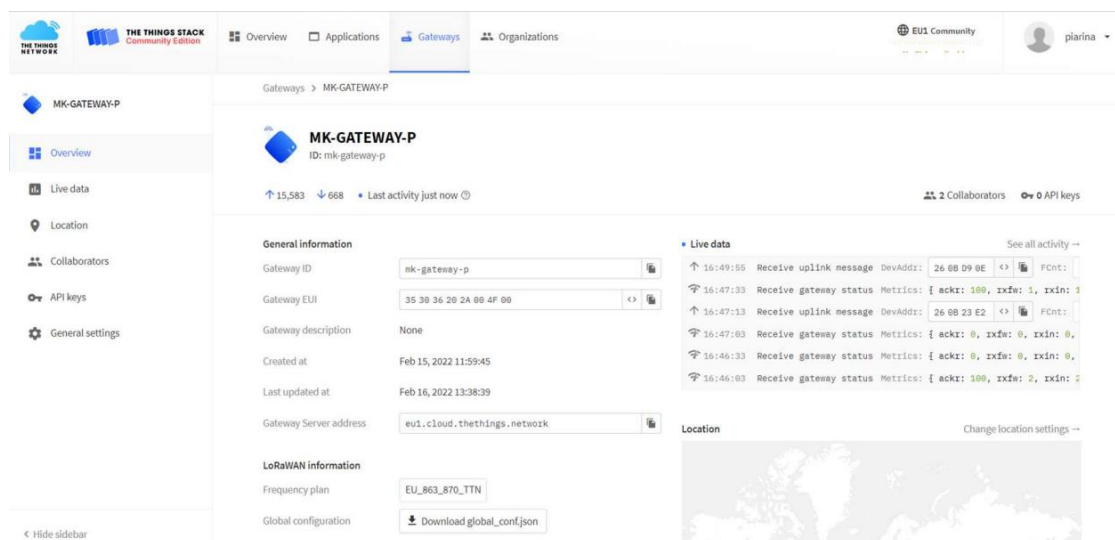


Figura 2. 10.- Visualización del Gateway en TTN. [8]

Una vez se ha hecho esto, ya se pueden enviar datos a la red de LoRa mediante la conexión de diferentes dispositivos, aunque aquí se va a emplear un sensor LHT65 compatible con LoRa, aunque podría haber infinidad de opciones que en cuanto a la conectividad, el proceso resultaría similar.

Este sensor se emplea para recoger datos de temperatura (tanto del entorno como la interna propia), humedad y tensión de la batería. Se caracteriza porque tiene un largo alcance y es bastante inmune frente a interferencias, a la vez que consume muy poco y su batería aguanta mucho tiempo, unos 10 años.

Al ser compatible con LoRa, lo primero que hay que hacer, antes de encenderlo, es crear una aplicación desde TTN para asignarle a ella y así, hacer que forme parte del Gateway.

Una vez dentro de la aplicación creada se debe registrar el dispositivo; esto se puede hacer de forma manual, o si es un dispositivo que se encuentra registrado dentro del repositorio de LoRaWAN. Al ser compatible, está registrado, por lo que el proceso es prácticamente automático.

Tras esto, simplemente es necesario encender el sensor y comprobar su conexión desde TTN, ya que en el dispositivo creado se van a empezar a recibir los mensajes, además de su estado.

A continuación, se muestra una imagen del dispositivo funcionando correctamente:

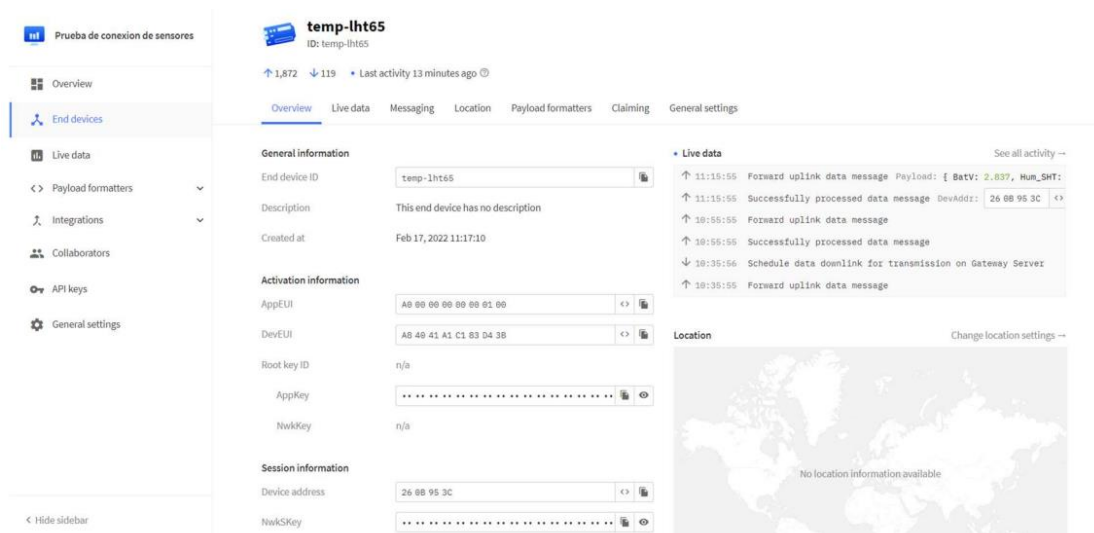


Figura 2. 11.- Sensor LHT65 funcionando en TTN. [8]

Si se quisiese conectar otro sensor que no fuese directamente compatible con LoRa, únicamente habría que tener en cuenta los valores necesarios para el registro manual del dispositivo; los cuales son: DevEui, AppEui y AppKey.

Con esto ya estaría explicado todo el proceso de conexión a la red LoRa, como se puede observar, es bastante sencillo y prácticamente automático una vez que se conoce el entorno y cómo trabajar con él.



2.3.- 5G

A continuación, se va a comenzar con una descripción de la quinta generación de telefonía móvil o 5G, para dar todos los datos necesarios y poder determinar correctamente su operabilidad.

2.3.1.- ¿Qué es 5G?

El término 5G hace referencia a la quinta generación de redes móviles; la cual proporciona mejoras en el ancho de banda y latencia, permitiendo una prestación de servicios mucho mayor que con las redes predecesoras. Su diseño busca ampliar las redes de telefonía 4G LTE existentes o reemplazarlas por completo en algunos casos determinados. [9]

Las redes 5G prometen velocidades de transmisión de datos de hasta 10 Gbps, latencia muy reducida y mejor cobertura en zonas remotas; aunque en la actualidad se limita a ciertas áreas y todavía no presenta una disponibilidad cercana a 4G LTE, ni mucho menos.

2.3.2.- Funcionamiento de la tecnología 5G

Las tecnologías que respaldan las redes 5G son bastante complejas. Además, se basa en enviar señales de muy baja frecuencia (menos de 6GHz) a largas distancias, las cuales dependen de torres celulares de alta potencia, el problema es que las altas velocidades teóricas, que se supone que debe alcanzar 5G, son muy difíciles de obtener teniendo en cuenta los requisitos de distancia y velocidad. [10]

También hay que tener en cuenta que cada vez hay más dispositivos móviles con conectividad que reducirán aún más las velocidades y degradarán las prestaciones cuanto mayor sea el número de ellos. Por este motivo, es necesario emplear nuevas tecnologías y métodos:

- Ondas milimétricas (mmWaves): son ondas a frecuencias muy altas (de los 20 a los 100 GHz) que pueden transmitir señales a velocidades increíbles. Desgraciadamente, no pueden recorrer largas distancias, hacer giros ni atravesar paredes. Para evitar esos problemas, se utilizan las frecuencias de banda media y baja. Sin embargo, si se instalan nodos de mmWave dentro de la línea de visión, las ondas de mayor frecuencia pueden saltar de un punto a otro y ofrecer el máximo de cobertura inalámbrica 5G con una latencia más baja.



- Conformado de haces: las torres celulares al emitir sus señales en todas las direcciones pueden generar mucha interferencia. Para evitar esto, la conformación de haces modera las señales y se enfoca en un solo flujo de datos para un usuario específico en un momento determinado. Una vez que se transfieren los datos, la señal se mueve hacia otro lugar, para poder cumplir con lo que solicitó otro usuario. Esta señal personalizada puede reducir considerablemente las interferencias entre las celdas celulares y lograr que la transmisión de datos sea más rápida y eficiente.
- Segmentación de la red: esta es una de las características más distintivas de la tecnología 5G, ya que permite que los proveedores dediquen segmentos virtuales de sus redes a usos específicos. Por ejemplo, los datos que se utilizan para el entretenimiento, la comunicación y el Internet harán uso de un segmento de la red, mientras que la transmisión de datos de máquina a máquina (una parte fundamental del Internet de las cosas o IoT) tendrá su propio segmento específico. Los datos esenciales, como los que se necesitan para los vehículos sin conductor, los servicios de emergencia y otras infraestructuras clave, tendrán un acceso exclusivo a la red 5G que no podrá ser utilizado por el resto de los servicios.
- Software-Defined Networking (SDN): utilizando redes definidas por software es posible configurar dinámicamente la red empleando software en lugar de hardware. Esto proporciona mejoras significativas en términos de flexibilidad y eficiencia.
- Network Function Virtualization (NFV): Reemplaza las funciones de red en dispositivos específicos, tales como encaminadores, equilibradores de carga y cortafuegos, con instancias virtualizadas que pueden ejecutarse en soportes físicos disponibles en el mercado, reduciendo así el coste de las modificaciones y actualizaciones de la red. Es un enfoque de red en evolución que permite la sustitución de dispositivos de hardware dedicados y costosos tales como routers, firewalls y equilibradores de carga con dispositivos de red basados en software que se ejecutan como máquinas virtuales en servidores estándares de la industria.

2.3.3.- Especificaciones de la tecnología 5G

Con la llegada de esta tecnología aumentaron y mejoraron mucho las características de las redes de telefonía móvil. Por este motivo se va a hacer un breve inciso en 8 de las principales especificaciones que ofrece: [11]

- Mejora de la tasa de datos, ya que el número de bits por unidad de tiempo que se transmiten va a ser de entre 10 y 100 veces mayor que en 4G.
- La amplitud de su disponibilidad de hasta casi el 100% y una cobertura del 100%.

- La reducción del tiempo de latencia traída por el avance que supone la tecnología *Edge Computing*.
- La mejora de la velocidad en redes con gran capacidad para transportar información; la cual es conocida como velocidad de banda ancha.
- Mayor duración de la batería en dispositivos que dependen de una conexión a Internet.
- Menor consumo energético y mayor capacidad de dispositivos conectados por área.

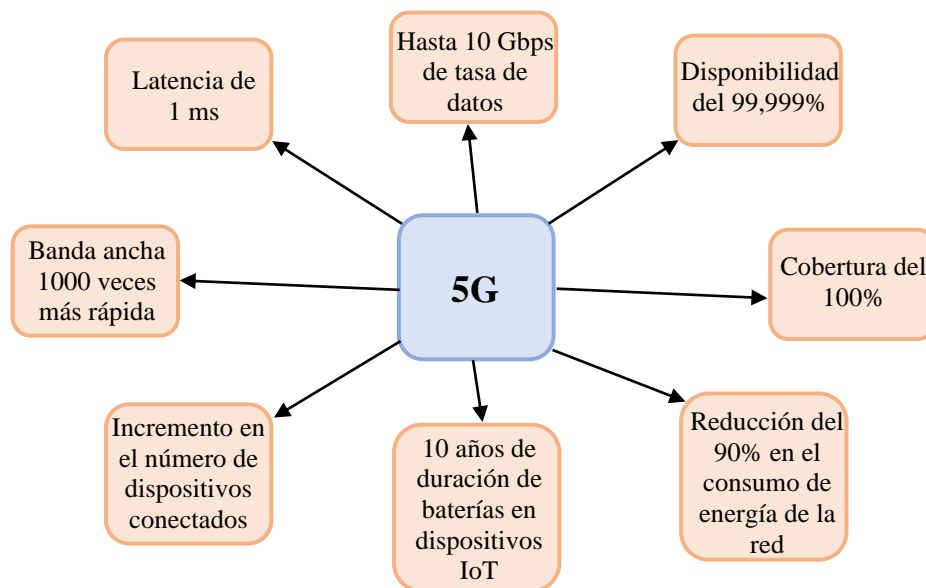


Figura 2. 12.- Especificaciones de la tecnología 5G.

El 5G no es una simple tecnología, sino que es el conjunto de muchos avances en sistemas de comunicación de radio y tecnologías con el fin de alcanzar una serie de estándares propuestos por la asociación 3GPP. Estos estándares se han ido lanzando con el paso del tiempo, de forma progresiva, consiguiendo evolucionar lo ya creado para el 4G y adaptarlo a las nuevas necesidades de la quinta generación.

2.3.4.- Arquitectura de la red 5G

La existencia de múltiples opciones de conectividad conlleva el riesgo de que distintos operadores implementen diferentes opciones, en un orden diferente y así avanzar de manera asíncrona, no consiguiendo un óptimo despliegue para el avance en la consecución de los estándares. Para reducir esta complejidad y proporcionar pasos que lleven hacia una arquitectura con desarrollo a largo plazo existen 3 opciones de implementación. [11]

2.3.4.1.- Arquitectura LTE/EPC

La arquitectura basada en LTE / EPC (Evolved Packet Core) representa el núcleo evolucionado de una red LTE. Según la asociación 3GPP, esta opción está incluida en las opciones de implementación inicial del 5G, sin embargo, la industria tecnológica no la cuenta para basar su despliegue inicial debido a su poca evolución respecto al 4G.

Esta arquitectura, mediante sus nodos hace que la red central LTE funcione, autentique las suscriptoras, determine el acceso a los suscriptores a la red y ayude fundamentalmente a la gestión de movilidad. En la siguiente figura se puede observar cómo el nodo de radio 5G se apoya en el nodo 4G para señalización, que está controlado por un núcleo 4G EPC.

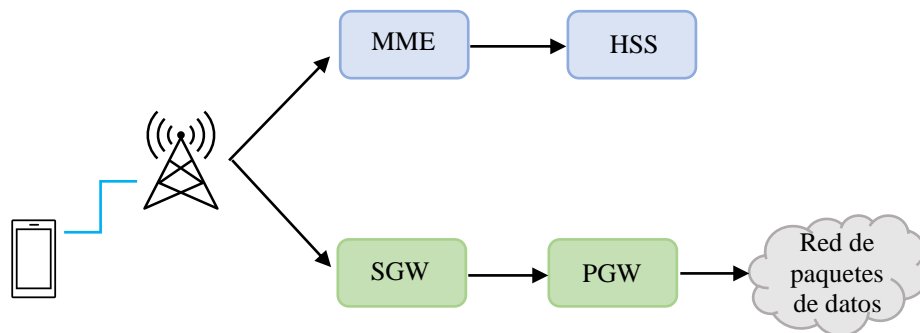


Figura 2. 13.- Esquema de arquitectura LTE/EPC.

El *Mobility Management Entity* (MME) autentica los usuarios comunicándose con el HSS, y la función de movilidad permite que el equipo de usuario acceda a la red y realice un seguimiento de su ubicación y estado.

El *Home Subscriber Server* (HSS) es la base de datos central que contiene todos los detalles relevantes sobre la información del suscriptor y la autenticación del usuario.

El *Service Gateway* (SGW) transporta datos IP desde el equipo usuario a la red principal LTE.

Por último, el *Packet Data Network Gateway* (PGW) enruta paquetes hacia y desde redes IP externas.

2.3.4.2.- Arquitectura de destino con NR y 5GC independiente

Esta arquitectura se puede implementar para casos de uso específicos en áreas locales donde los dispositivos se mantienen dentro de una buena cobertura de *New Radio* (NR).

El núcleo del 5G Core es el corazón de la red móvil, donde la conectividad se traduce en diferenciación de servicios y flexibilidad empresarial.

Establece conectividad confiable y segura a la red y acceso a sus servicios, determina la calidad del servicio y la aplica a través de políticas que permiten la diferenciación de servicios y maneja la movilidad de área amplia en toda la red.

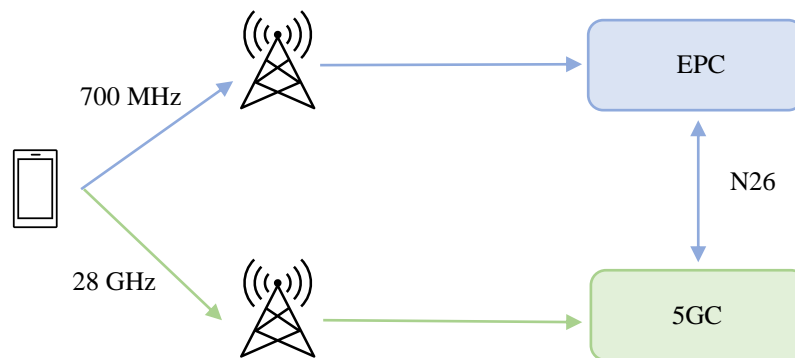


Figura 2. 14.- Esquema de arquitectura NR y 5GC.

2.3.4.3.- Conectividad dual

Por último, la conectividad dual se basa básicamente en permitir que un dispositivo se conecte simultáneamente a una red 5G y LTE transmitiendo y recibiendo datos por ambas interfaces aéreas logrando así un mayor rendimiento que conectándose sólo a una de ellas y unas velocidades de banda ancha mucho mayor.

Las dos bandas en las que se emiten simultáneamente señales son 2.4GHz y 5Ghz, esta última está menos congestionada lo que hace que se genere menos ruido e interferencias y además permite una mayor velocidad mediante el uso simultaneo de varios canales. Lo malo de la banda de 5GHz es el alcance, no consiguiendo tanto como la de 2.4GHz.

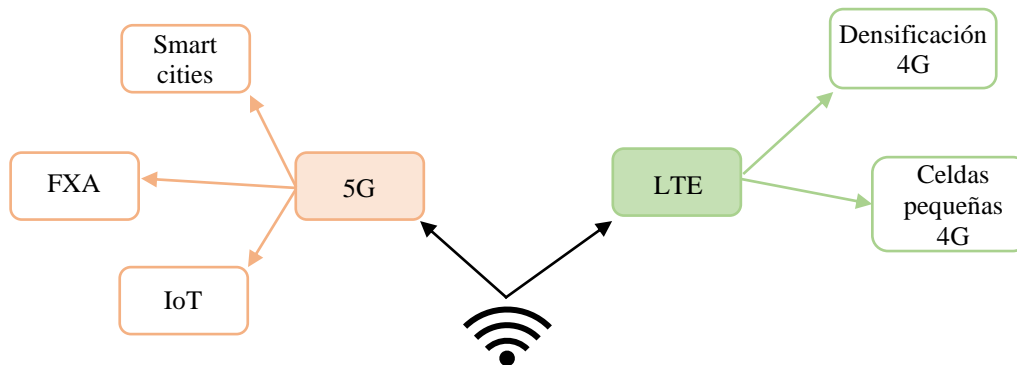


Figura 2. 15.- Esquema de arquitectura dual.

2.3.5.- Seguridad en redes 5G

La tecnología 5G se basa en las prácticas de seguridad de generaciones pasadas de tecnologías móviles, a pesar de que el modelo de confianza se ha hecho mucho más amplio, con más protagonistas implicados en el proceso de prestación de servicios. El IoT y la propagación de usuarios suponen un número exponencialmente mayor de puntos finales, donde muchos de estos datos de tráfico ya no están supervisados por personas. [10]

Entre las características de seguridad mejoradas de la tecnología 5G que se detallan en los estándares del 3GPP, se encuentran la autenticación unificada para separar la autenticación de los puntos de acceso, los protocolos de autenticación extensible para permitir transacciones seguras, las políticas de seguridad flexible para abordar más casos prácticos y los identificadores permanentes de suscriptores (SUPI) para garantizar la privacidad en la red.

A medida que la implementación de la tecnología 5G continúe y los nodos de rendimiento críticos se virtualicen cada vez más, los operadores tendrán que monitorizar y evaluar constantemente el rendimiento de la seguridad. Cumplir las prácticas recomendadas pasa por la monitorización de la seguridad de las redes de extremo a extremo a lo largo de las aplicaciones, los dispositivos y la arquitectura del sistema.

2.4.- Conexión a redes 5G

Al igual que se ha hecho para el caso de LoRa, y tras conocer un poco qué es la tecnología 5G, se va a proceder a explicar la conectividad en este tipo de redes realizando un despliegue de tipo IoT. Se ha de añadir otra vez que esto se va a comentar sin entrar mucho en profundidad porque ya se exploró más en detalle en el proyecto conjunto. [8]

Para llevar a cabo esto se emplearon dos sensores: un sensor de temperatura y humedad (DHT11) y un sensor de intensidad lumínica (LDR). Además de esto va a ser necesaria una Raspberry Pi para enviar los datos de los sensores a la plataforma IoT seleccionada, que en este caso es “Thingsboard IoT Gateway”.

Esta plataforma IoT también es de código abierto, y el primer paso a realizar es ponerla en marcha y configurarla lanzando el servidor Thingsboard para poder emplear una demo en la nube. Al tener una RPi, todo este proceso se ha hecho desde ella.

Tras tener el servidor de Thingsboard corriendo, tras haberlo configurado con el tutorial dado por la propia plataforma, se puede acceder a la plataforma utilizando un navegador con la siguiente url: <http://156.35.117.17:8080/>, siendo 156.35.117.17 la IP de acceso a internet de la RPi. Tras loguearse, se puede observar la siguiente pantalla:

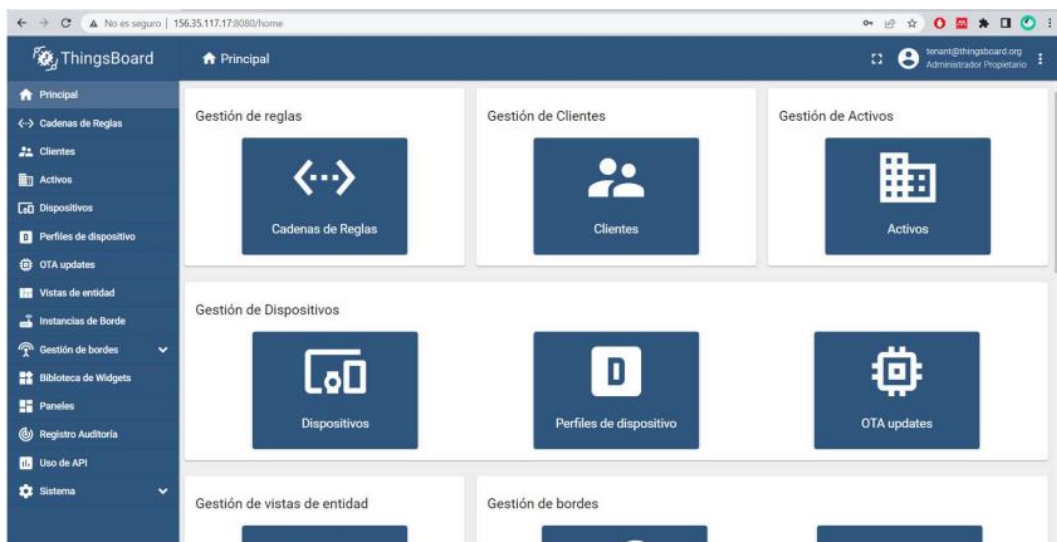


Figura 2. 16.- Pantalla inicial de ThingsBoard. [8]

Ahora, al igual que en el caso de LoRa, es necesario crear un nuevo dispositivo que se va a corresponder con la RPi, ya que los valores de los sensores se van a almacenar bajo el nombre de este.

Tras haber configurado esto, el siguiente paso es poner en funcionamiento los sensores mediante su correcto conexionado en los pines de la RPi y la ejecución de un



código Python para que estos funcionen correctamente y proporcionen una buena lectura de los datos.

El problema de esta configuración surge al intentar enviar los datos mediante una red 5G, ya que el proceso se vuelve mucho más complejo, ya que hay que investigar si los dispositivos comerciales pueden proveer de red 5G al despliegue IoT a implementar, así como investigar el funcionamiento y puesta en marcha del dispositivo.

Para todo esto, también es necesario tener disponible una tarjeta SIM de datos que permita conectarse a la red 5G; y configurarla correctamente para que funcione cómo se desea. En el caso de este proyecto, para que funcione dentro de la red del laboratorio.

Se han tenido en cuenta varios métodos para dotar de conectividad 5G al sistema, como por ejemplo: un router móvil, un teléfono móvil que permite compartir la red 5G a la RPi utilizando un cable USB y un módulo 5G con su correspondiente adaptador 5G a USB 3.0 junto con las antenas necesarias para el correcto funcionamiento del módulo.

Tras dotar de conectividad 5G a la RPi mediante uno de estos sistemas, se deberían recibir los datos de los sensores en la aplicación de ThingsBoard.

La cuestión es que configurar estos sistemas para compartir la red 5G es bastante tedioso y lleva bastante tiempo, es por ello que este método requiere mucho más trabajo y esfuerzo que LoRa, en comparación.

Más adelante en este proyecto se va a mostrar el proceso de configuración de la tercera opción mencionada; es decir, el módulo 5G con el adaptador. Y gracias a este problema surge la idea de este proyecto, ya que resultaría muy sencillo tener un sistema al que conectarse ya sea por WiFi o Ethernet, y te permita tener acceso directo a la red 5G sin complicaciones ni configuraciones adicionales.

2.5.- Comparativa entre las redes LoRa y 5G

A la vista de toda la información que se ha ido explicando y de los casos prácticos que han sucedido, puede resultar interesante comparar estos dos protocolos para poder determinar cuál es más beneficioso a la hora del despliegue de una red en función de las necesidades de cada proyecto.

En primer lugar, y atendiendo a la tasa de datos, LoRa no tiene nada que hacer con 50 Kbps de pico frente a los 20 Gbps de 5G; por este motivo si el proyecto requiere un empleo masivo de datos 5G es la mejor opción. Por el contrario si no se requiere trabajar con muchos datos, ni de manera continua, LoRa permite trabajar con consumos muy bajos



haciendo que las baterías de los dispositivos que vayan a formar parte de la red duren mucho tiempo, del orden de varios años.

En cuanto a términos de cobertura LoRa abarca mucho más terreno que 5G, teniendo un rango efectivo de entre 10 y 40 Km, mientras que 5G podría dar cobertura a 1 Km de distancia como mucho.

Si se quiere realizar un despliegue de LoRa, se puede llevar a cabo de manera individual siempre y cuando se disponga del hardware necesario, es decir, el Gateway, la antena y un router con conexión Ethernet para poder conectarlo a una red, ya sea alámbrica o inalámbrica. En cambio, si se quiere hacer un despliegue 5G, va a ser necesario contratar la red a una empresa de telefonía móvil ajena.

Otra cosa muy a tener en cuenta es la dificultad, ya que como se ha ido comentando, realizar un red LoRa y conectar dispositivos es muy sencillo, casi automático en muchos casos. En cambio, 5G da más problemas y hay que realizar muchas más configuraciones, además de tener muchas más cosas en cuenta como la SIM, compatibilidades, etc.

Atendiendo a la capacidad de la red, ambas redes están pensadas para dar servicio a muchos dispositivos a la vez, teniendo 5G como capacidad teórica 1.000.000 de dispositivos por Km². LoRa a pesar de ser buena en esto también, nunca a va a acercarse a esas cifras, ya que a pesar de que no se encuentre información a cerca de un valor determinado, 5G es la mejor opción existente.

Por último, teniendo en cuenta el aspecto financiero, 5G al ser una tecnología más novedosa cuenta que un menor número de dispositivos comerciales y esto conlleva a precios mucho más elevados. En cambio LoRa al ser una tecnología mas asentada en el sector, los precios hoy en día son bastante reducidos en comparación, además de existir multitud de dispositivos compatibles.

En la siguiente tabla se pueden observar estas características mejor:

Tipo de red	Banda de frecuencia	Rango de cobertura	Tasa de datos	Capacidad	Complejidad	Disponibilidad de dispositivos
5G	ISM (868 MHz en Europa)	10-40 Km (zona rural) y 1-5 Km (zona urbana)	< 50 Kbps	Elevada	Reducida	Muy elevada
LoRa	700 MHz y 3,5 GHz	1 Km	< 20 Gbps	Muy elevada	Muy elevada	Limitada

Tabla 2. 1.- Comparativa entre 5G y LoRa.



3.- SISTEMA DE CONEXIÓN DE RED 5G

Una vez que se ha comentado todo el proceso para las conexiones a las redes empleando ambas tecnologías, se puede observar que en el caso de 5G la dificultad es muy superior, ya que para LoRa, prácticamente es directo. Por este motivo, en este capítulo se va a realizar un sistema para la conexión de dispositivos a una red 5G ya sea de forma inalámbrica, a través de una red WiFi, o mediante cable gracias a una conexión de tipo Ethernet.

3.1.- Descripción detallada del proyecto

Las redes 5G siguen siendo muy novedosas en la actualidad, es por ello que tanto la información como los proyectos que se llevan a cabo siguen siendo pocos, ya que se encuentra en un proceso de continua evolución. Además, todos los dispositivos o componentes compatibles con esta tecnología, ya sea cómo para realizar un estudio o uso particular, tienen un acceso muy limitado y precios excesivamente altos. Es por ello que en este proyecto se pretende buscar una solución e implementar un sistema de conexión a redes 5G cuyo precio sea relativamente contenido y accesible.

Una vez que está clara la finalidad y tras pensar en diferentes opciones, se terminó decidiendo que la mejor era el empleo de una Raspberry Pi 4 como dispositivo que va a realizar la función de *router*, haciendo lo que se conoce como un *hotspot*. Esto es debido a cuenta con varias interfaces (Wifi, Ethernet y USB), tiene la potencia necesaria para que se puedan conectar varios dispositivos a la vez sin degradar mucho las condiciones de la red, su precio no es excesivamente elevado y además, la obtención de este tipo de dispositivos es sencilla tanto para uso doméstico como profesional.

Por ello, se pretende elaborar un sistema que facilite el acceso a la red 5G a otros dispositivos que se quieran conectar en cualquier momento de una manera rápida y sencilla. A continuación, se muestra una primera imagen para tener una noción más visual de todo lo que se va explicando: [12]

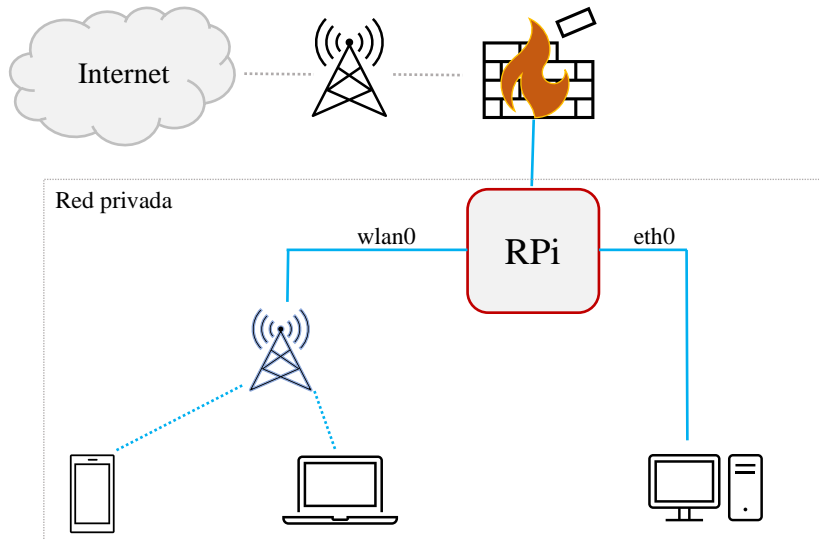


Figura 3. 1.- Esquema del proyecto.

Haciendo referencia a la imagen, como elemento que va a proporcionar la conectividad a internet mediante la red 5G se va a emplear un módem usb, el cual va a tener una tarjeta SIM de datos que permita acceder a la red 5G. También es necesario incorporarle unas antenas compatibles con la banda de frecuencias apropiada, ya que si no, las prestaciones serían mucho peores.

Este módem se puede conectar a la Raspberry Pi 4 mediante dos modos de funcionamiento diferentes: QMI y EMC. Esto se explicará más adelante durante el manual de funcionamiento, ya que difieren un poco a la hora del posterior funcionamiento, ya que el modo QMI funciona a través de la interfaz wwan0 y, en cambio, el modo EMC emplea la interfaz usb0, por lo que se debe seleccionar uno de los dos atendiendo a la opción que pueda ser mejor para el proyecto.

Una vez que se haya configurado el módem y se haya comprobado su correcto funcionamiento dentro de la red 5G (se puede verificar realizando un ping hacia un dominio público de internet como por ejemplo www.google.com), se procedió a realizar la red. Para ello se emplearon dos raspberrys en lugar de una, duplicando así el proyecto en dos configuraciones:

- La primera Raspberry se va a utilizar para el desarrollo de la red 5G de conexión inalámbrica mediante WiFi; además, el módem se ha configurado en el modo de operación EMC.

- La segunda Raspberry se va a utilizar para el desarrollo de la red 5G de conexión por cable mediante Ethernet; además, el módem se ha configurado en el modo de operación EMC.

La cuestión de realizar el proyecto de manera separada, en vez de integrarlo todo sobre la misma raspberry fue por problemas con el software, ya que la configuración de una conectividad afectaba a la otra y daba problemas a la hora de conectar dispositivos a la red. Además, el rendimiento era bastante más bajo, por lo que disponiendo de suficientes raspberrys se optó por esta solución, debido a que de esta manera la realización de las pruebas de conexión fueron mejores.

Una vez que se ha aclarado esto, se va a mostrar en la siguiente imagen el uno de los sistemas en funcionamiento:



Figura 3. 2.- Raspberry Pi Hotspot.

Esta figura da una primera idea de cómo es el sistema en funcionamiento, aunque en el siguiente apartado se va a entrar más en detalle con todos los componentes de manera separada.

Para la elaboración de las redes se tuvieron en cuenta varias opciones, destacando un software llamado OpenWrt, el cual es un sistema operativo basado en Linux que permite desarrollar múltiples aplicaciones en Raspberry, aunque generaba un problema de compatibilidad con el modem USB que se empleó, ya que no lo reconocía correctamente y perdía la conexión cada poco. Por este motivo, se prefirió no modificar el sistema

operativo propio de Raspberry (Raspberry Pi OS) y así, poder disponer de todas las funcionalidades del equipo además de la finalidad que se les da con el proyecto.

Por último, tras comprobar el funcionamiento de ambas configuraciones (WiFi y Ethernet), se realizaron una serie de pruebas de conexión con distintos equipos, tal y cómo se va a mostrar más adelante dentro de este capítulo.

3.2.- Componentes del sistema

Tal y cómo se ha comentado, en este apartado se van a detallar todos los componentes de hardware empleados para la realización del proyecto.

En primer lugar, el primer elemento necesario para el sistema propuesto, como ya se ha ido comentando a lo largo de los apartados anteriores, es una Raspberry Pi 4. En particular se ha empleado un kit de LabSist que ya viene con todos los elementos necesarios para el correcto desarrollo de un proyecto en este entorno; los cuales son: tarjeta micro SD de memoria con el sistema operativo Raspberry Pi OS, una placa de refrigeración compuesta de dos ventiladores y dos radiadores, un adaptador de micro SD a USB tipo C y A, un cable de carga y dos cables de microHDMI a HDMI.

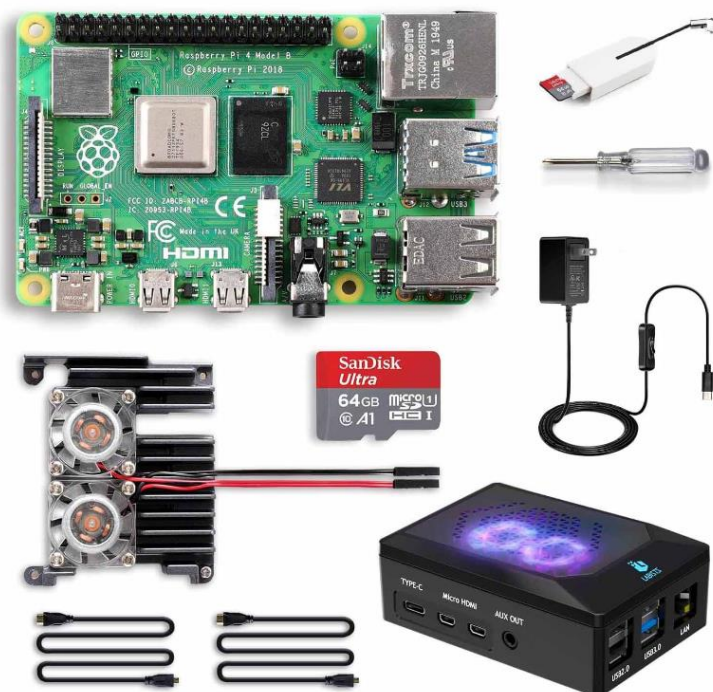


Figura 3. 3.- Kit de LabSist de Raspberry Pi 4. [13]

A continuación, se va a hablar de los elementos que componen el módulo de red 5G, para ello, además del propio módem 5G, también es necesario un adaptador que permita conectarse por USB 3.0 a la Raspberry, además, tal y como se ha dicho previamente, también se le soldaron unas antenas al módem que permitan el correcto funcionamiento. Estos elementos se adquirieron juntos en un kit del fabricante Queltec.

Por último, el último elemento que lo compone es una tarjeta SIM de datos que permita la conexión 5G. Es importante comentar que esta tarjeta SIM va a estar asociada a la red 5G del laboratorio donde se realizó el proyecto, es decir, fuera de ese laboratorio no permite el uso de datos.

A continuación, se muestran unas imágenes de los componentes para que se puedan observar con un mayor detalle:



Figura 3. 4.- Kit de Queltec para el montaje del adaptador del módulo 5G a USB 3.0.



Figura 3. 5.- Tarjeta SIM de datos.



Figura 3. 6.- Módulo 5G montado y con la SIM insertada..

Estos son los componentes básicos necesarios para elaborar el proyecto en su totalidad; únicamente falta por añadir que es necesario tanto un cable Ethernet para poder realizar y testear la comunicación por cable, además del equipamiento necesario para trabajar con la raspberry (monitor, teclado...) y los equipos que generan la red 5G dentro del laboratorio donde se ha llevado a cabo el proyecto.

Más adelante, en los anexos de esta memoria se van a encontrar los presupuestos para tener una idea de lo que costaría realizar un proyecto similar a este.



3.3.- Manual de funcionamiento

A continuación, en esta parte de la memoria se va a desarrollar el proceso de configuración de los equipos para poder obtener el propósito del trabajo. Para ello se va a elaborar un manual sencillo, comentando todos los pasos que son necesarios, junto con otras opciones que pueden resultar interesantes.

Para ello, este apartado se va a dividir en dos subapartados, diferenciando, en primer lugar, la configuración del módem USB para dar señal a la raspberry; y, en segundo lugar, se van a explicar las configuraciones para poder conectarse a la red de forma cableada e inalámbrica.

3.3.1.- Configuración del módem 5G

En primer lugar, tal y cómo se ha dicho previamente, se va a proceder a explicar las dos posibles configuraciones del módulo 5G para que se pueda utilizar la más conveniente. [14]

No está claro que una configuración sea mejor que la otra, ni mucho menos, pero en función de las pruebas que se han realizado, tanto la latencia como las velocidades de carga son mejores con el controlador del modo EMC (cdc_ether); aunque tampoco hay tanta diferencia. Por ese motivo, además de una mayor facilidad es el modo que se recomienda ya que tiene una mejor compatibilidad con Linux.

Tras haber comentado esto y antes de configurar uno de los dos modos, es necesario hacer unas comprobaciones iniciales en la raspberry, teniendo en cuenta que se va a emplear con el sistema operativo recién iniciado.

El módulo 5G va a ser reconocido por la RPi nada más que se conecte, sin la necesidad de la instalación de ningún driver adicional. Esto se puede observar ejecutando el comando:

```
$ ifconfig
```

Con este, se puede apreciar que se le asigna automáticamente una dirección IP del tipo wwan0, pero a pesar de esto, no es posible conectarse a ninguna red, ya que si se intenta realizar un ping a cualquier dominio público, se observa que no se llega al destino, por lo que es necesaria la configuración del dispositivo aunque aparentemente no lo fuese. Este proceso se puede apreciar en las siguientes imágenes:



```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~ $ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether e4:5f:01:31:29:86 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 35 bytes 3035 (2.9 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 35 bytes 3035 (2.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether e4:5f:01:31:29:87 txqueuelen 1000 (Ethernet)
    RX packets 957 bytes 1064642 (1.0 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 485 bytes 56849 (55.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wwan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 169.254.205.197 netmask 255.255.0.0 broadcast 169.254.255.255
    inet6 fe80::7e2d:6bab:3bb6:1896 prefixlen 64 scopeid 0x20<link>
    ether 76:08:34:6d:05:24 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 60 dropped 0 overruns 0 carrier 0 collisions 0

pi@raspberrypi:~ $
```

Figura 3. 7.- Ejecución del comando \$ifconfig en la terminal.

```
pi@raspberrypi:~ $ ping -I wwan0 www.google.com -c 5
ping: www.google.com: Fallo temporal en la resolución del nombre
pi@raspberrypi:~ $
```

Figura 3. 8.- Fallo en la conexión a internet con el módulo 5G.

Una vez se ha comprobado esto se va a proseguir con las comprobaciones iniciales de la RPi; comenzando por observar si se detecta el módulo Quectel correctamente mediante los siguientes comandos:

\$ lsusb

\$ lsusb -t

Ambos comandos sacan la información que se está buscando, lo único, que el segundo la presenta de una forma más resumida; aunque para lo que se necesita ahora mismo con el primero es suficiente.



```
pi@raspberrypi:~$ ping -I wwan0 www.google.com -c 5
ping: www.google.com: Fallo temporal en la resolución del nombre
pi@raspberrypi:~$ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 004: ID c0f4:01e0
Bus 001 Device 006: ID 2c7c:0800 Quectel Wireless Solutions Co., Ltd.
Bus 001 Device 003: ID 0000:3825
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
pi@raspberrypi:~$
```

Figura 3. 9.- Ejecución del comando \$lsusb en la terminal.

Con este comando ya se puede observar que se reconoce perfectamente el dispositivo, si por lo que fuese el dispositivo no aparece al ejecutar este comando, puede ser porque el módulo se encuentre mal conectado o no funcione correctamente.

Se pueden realizar más comprobaciones como por ejemplo los logs de kernel para observar que hay mensajes desde qmi_wan o también se puede mirar si esta listado cómo cdc-wdm0; para ello se pueden ejecutar los siguientes comandos teniendo en cuenta el orden que se acaba de mencionar:

```
$ dmesg | grep qmi
```

```
$ ls /dev/cdc*
```

Una vez que se han hecho todas estas comprobaciones iniciales, el dispositivo está listo para empezar con su configuración.

3.3.1.1.- Modo QMI (interfaz wwan0)

Esta interfaz es una de las más utilizadas para interactuar con diferentes dispositivos y en el caso de los módem, suele ser la que viene configurada por defecto tal y cómo se ha podido apreciar anteriormente. Por ello, en este apartado se va a enseñar el cómo configurarla correctamente mediante comandos en la terminal de Linux.

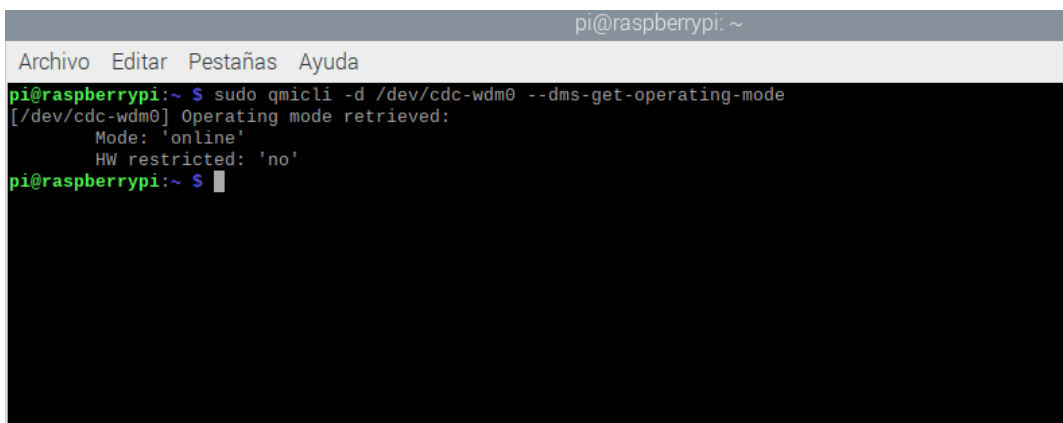
Primero de todo, es necesario instalar una serie de librerías para interactuar con el módem a través de QMI y configurarlo, para ello se ejecuta el siguiente comando:

```
$ sudo apt update && sudo apt install libqmi-utils udhpcp
```

Una vez que ha concluido el proceso de descarga se debe chequear el modo en el que está operando el módulo 5G; esto se realiza de la siguiente manera:

```
$ sudo qmicli -d /dev/cdc-wdm0 --dms-get-operating-mode
```

Una vez hecho esto, se debe obtener la siguiente salida por pantalla:



```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~ $ sudo qmicli -d /dev/cdc-wdm0 --dms-get-operating-mode
[/dev/cdc-wdm0] Operating mode retrieved:
  Mode: 'online'
  HW restricted: 'no'
pi@raspberrypi:~ $
```

Figura 3. 10.- Modo operativo del módem (QMI).

Si la respuesta a este comando fuese: “Resource temporarily unavailable” tras unos minutos, va a ser necesario parar el ModemManager y volver a probar el comando anterior. Para parar el ModemManager se debe poner el siguiente comando:

```
$ sudo systemctl stop ModemManager
```

También puede suceder que el modo operativo no retorne online, en ese caso se debe forzar a que empiece a funcionar de esta manera.

```
$ sudo qmicli -d /dev/cdc-wdm0 --dms-set-operating-mode='online'
```

Cuando ya se tenga bien configurado, tal y como se ha mostrado en la imagen anterior, se puede pasar al siguiente paso, el cual es poner la interfaz en modo `raw_ip` mediante los siguientes comandos:

```
$ sudo ip link set wwan0 down
```

```
$ echo 'Y' | sudo tee /sys/class/net/wwan0/qmi/raw_ip
```

```
$ sudo ip link set wwan0 up
```

El segundo comando debería devolver por Y pantalla al ejecutarse, y una vez que se ha hecho, para confirmar que esté funcionando en este modo se puede hacer mediante el siguiente comando:

```
$ sudo qmicli -d /dev/cdc-wdm0 --wda-get-data-format
```

En la siguiente imagen se puede observar cómo es este proceso y cómo aparece el modo `raw_ip` en el campo `Link Layer protocol`.

```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~ $ sudo ip link set wwan0 down
pi@raspberrypi:~ $ echo 'Y' | sudo tee /sys/class/net/wwan0/qmi/raw_ip
Y
pi@raspberrypi:~ $ sudo ip link set wwan0 up
pi@raspberrypi:~ $ sudo qmicli -d /dev/cdc-wdm0 --wda-get-data-format
[/dev/cdc-wdm0] Successfully got data format
      QoS flow header: no
      Link layer protocol: 'raw-ip'
      Uplink data aggregation protocol: 'disabled'
      Downlink data aggregation protocol: 'disabled'
      NDP signature: '0'
      Uplink data aggregation max size: '0'
      Downlink data aggregation max size: '0'
pi@raspberrypi:~ $ █
```

Figura 3. 11.- Modo operativo de funcionamiento raw_ip (QMI).

Ahora es el momento de conectar la tarjeta SIM de datos a la red, para ello, únicamente es necesario conocer el *Access Point Name* (APN) que tenga asignada; en el caso de la que se utilizó en este proyecto es: “oai.ipv4”. Este proceso se muestra a continuación:

```
$ sudo qmicli -p -d /dev/cdc-wdm0 --device-open-net='net-raw-ip|net-no-qos-
header'--wds-start-network='apn='YOUR_APN',ip-type=4'--client-no-release-cid
```

Es en este comando donde hay que sustituir el valor de la APN de la SIM que se vaya a emplear por YOUR_APN.

Otra cosa interesante es que si se tiene un usuario y contraseña, estos se pueden añadir detrás de la APN en el comando de la siguiente manera: `apn='YOUR_APN',username='YOUR_USERNAME',password='YOUR_PASSWORD',ip-type=4`.

Este proceso se muestra a continuación en la siguiente imagen, aunque en este proyecto no sé tuvieron en cuenta ni el usuario ni la contraseña.


```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~$ sudo ip link set wwan0 down
pi@raspberrypi:~$ echo 'Y' | sudo tee /sys/class/net/wwan0/qmi/raw_ip
Y
pi@raspberrypi:~$ sudo ip link set wwan0 up
pi@raspberrypi:~$ sudo qmicli -d /dev/cdc-wdm0 --wda-get-data-format
[/dev/cdc-wdm0] Successfully got data format
QoS flow header: no
Link layer protocol: 'raw-ip'
Uplink data aggregation protocol: 'disabled'
Downlink data aggregation protocol: 'disabled'
NDP signature: '0'
Uplink data aggregation max size: '0'
Downlink data aggregation max size: '0'
pi@raspberrypi:~$ sudo qmicli -p -d /dev/cdc-wdm0 --device-open-net='net-raw-ip|net-no-qos-header' --wds-start-network
= "apn='oai.ipv4',ip-type=4" --client-no-release-cid
[/dev/cdc-wdm0] Network started
Packet data handle: '1704603856'
[/dev/cdc-wdm0] Client ID not released:
Service: 'wds'
CID: '15'
pi@raspberrypi:~$
```

Figura 3. 12.- Configuración de la APN de la SIM (QMI).

El último paso que queda para poder utilizar el módem es configurar `udhcpc` para asignarle una IP por defecto, además de una ruta. Esto se hace de la siguiente manera:

```
$ sudo udhcpc -q -f -i wwan0
```

```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~$ sudo ip link set wwan0 down
pi@raspberrypi:~$ echo 'Y' | sudo tee /sys/class/net/wwan0/qmi/raw_ip
Y
pi@raspberrypi:~$ sudo ip link set wwan0 up
pi@raspberrypi:~$ sudo qmicli -d /dev/cdc-wdm0 --wda-get-data-format
[/dev/cdc-wdm0] Successfully got data format
QoS flow header: no
Link layer protocol: 'raw-ip'
Uplink data aggregation protocol: 'disabled'
Downlink data aggregation protocol: 'disabled'
NDP signature: '0'
Uplink data aggregation max size: '0'
Downlink data aggregation max size: '0'
pi@raspberrypi:~$ sudo qmicli -p -d /dev/cdc-wdm0 --device-open-net='net-raw-ip|net-no-qos-header' --wds-start-network
= "apn='oai.ipv4',ip-type=4" --client-no-release-cid
[/dev/cdc-wdm0] Network started
Packet data handle: '1704603856'
[/dev/cdc-wdm0] Client ID not released:
Service: 'wds'
CID: '15'
pi@raspberrypi:~$ sudo udhcpc -q -f -i wwan0
udhcpc: started, v1.30.1
No resolv.conf for interface wwan0.udhcpc
udhcpc: sending discover
udhcpc: sending select for 12.1.1.5
udhcpc: lease of 12.1.1.5 obtained, lease time 7200
Too few arguments.
Too few arguments.
pi@raspberrypi:~$
```

Figura 3. 13.- Configuración de `udhcpc` (QMI).



Con esta configuración ya sería posible conectarse a internet a través del módem. Esto se puede comprobar con un simple ping y viendo que ya hay recepción de datos, tal y cómo se muestra:

```
$ ping -I wwan0 www.google.com -c 5
```

```
pi@raspberrypi:~$ ping -I wwan0 www.google.com -c 5
PING www.google.com (142.250.200.100) from 12.1.1.5 wwan0: 56(84) bytes of data.
64 bytes from mad41s13-in-f4.1e100.net (142.250.200.100): icmp_seq=1 ttl=114 time=39.9 ms
64 bytes from mad41s13-in-f4.1e100.net (142.250.200.100): icmp_seq=2 ttl=114 time=91.5 ms
64 bytes from mad41s13-in-f4.1e100.net (142.250.200.100): icmp_seq=3 ttl=114 time=39.9 ms
64 bytes from mad41s13-in-f4.1e100.net (142.250.200.100): icmp_seq=4 ttl=114 time=40.5 ms
64 bytes from mad41s13-in-f4.1e100.net (142.250.200.100): icmp_seq=5 ttl=114 time=29.8 ms

--- www.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4ms
rtt min/avg/max/mdev = 29.842/48.321/91.473/21.940 ms
pi@raspberrypi:~$
```

Figura 3. 14.- Comprobación de conexión del módulo 5G (QMI).

También se puede lanzar el comando “\$ ip a” para observar las configuraciones del módem, aunque con el ping ya se haya verificado que funciona perfectamente.

Aunque pueda parecer que con esto ya está todo hecho, surge un problema, y es que al apagar o reiniciar la RPi se pierde la configuración. Por este motivo habría que volver a lanzar una serie de comandos para que vuelva a funcionar correctamente.

```
$ sudo ip link set wwan0 down
$ echo 'Y' | sudo tee /sys/class/net/wwan0/qmi/raw_ip
$ sudo ip link set wwan0 up
$ sudo qmicli -p -d /dev/cdc-wdm0 --device-open-net='net-raw-ip|net-no-qos-
header'—wds-start-network='apn='YOUR_APN',ip-type=4'—client-no-release-cid
$ sudo udhcpc -q -f -i wwan0
```

Esto resulta poco práctico, por lo que a continuación se propone una solución para que no sea necesario tener que lanzar ningún comando y que nada más se inicie la RPi y se le conecte el módem, ya empiece a funcionar correctamente. Para ello se ha creado un archivo para la interfaz wwan0 en la ruta `/etc/network/interfaces.d/wwan0` con el comando:

```
$ sudo nano /etc/network/interfaces.d/wwan0
```

Este fichero tiene que contener la siguiente información:

```
auto wwan0
iface wwan0 inet manual
```



```
pre-up ifconfig wwan0 down
pre-up echo Y > /sys/class/net/wwan0/qmi/raw_ip
pre-up for _ in $(seq 1 10); do /usr/bin/test -c /dev/cdc-wdm0 && break; /bin/sleep 1;
done
pre-up for _ in $(seq 1 10); do /usr/bin/qmicli -d /dev/cdc-wdm0 --nas-get-signal-
strength && break; /bin/sleep 1; done
pre-up sudo qmicli -p -d /dev/cdc-wdm0 --device-open-net='net-raw-ip|net-no-qos-
header'--wds-start-network='apn='oai.ipv4',ip-type=4'--client-no-release-cid
pre-up udhcpc -i wwan0
post-down /usr/bin/qmi-network /dev/cdc-wdm0 stop
```

Esta configuración va a hacer que siempre que se inicie la RPi con el módulo conectado se va a configurar automáticamente y se va a tener conexión perfectamente. Si por lo que sea sólo se quiere quitar el módulo 5G y volver a conectarlo, se podría o reiniciar la RPi, o ejecutar los comandos de configuración que se comentaron antes.

Con esto ya quedaría explicada cómo realizar la configuración mediante el método QMI, empleando la interfaz wwan0.

3.3.1.2.- Modo ECM (interfaz usb0)

A continuación se va a explicar el método empleado para configurar el módulo 5G mediante la interfaz usb0. Se ha de comentar que ha sido el método empleado para el desarrollo del proyecto, ya que presenta una mayor compatibilidad y, además, es mucho más sencillo de implementar. También resulta interesante que después de un reinicio de la RPi o al desconectar el módulo y volver a conectarlo, este sigue funcionando perfectamente.

Antes de nada, si se ha realizado la configuración QMI previamente, va a ser necesario borrar o comentar todas las líneas de código que había en `/etc/network/interfaces.d/wwan0`, y también, por si acaso, asegurar que la interfaz wwan0 está apagada mediante el comando:

```
$ sudo ifdown wwan0
```

Una vez se ha comprobado esto, se puede comenzar con la configuración del modo ECM; para ello va a ser necesario instalar “minicom” para comunicarse con el módem mediante el puerto serie.



```
$ sudo apt install minicom -y
```

Una vez que el proceso de instalación haya concluido, se puede iniciar minicom y empezar una conexión serie. Para ello se lanza el siguiente comando:

```
$ minicom -D /dev/ttyUSB2 -b 115200
```

Al abrirse esta consola de comandos, lo primero que hay que hacer es escribir AT y presionar 'Enter', de este modo se debería recibir un OK, y así se verifica que la comunicación esté activa con el módem.

A continuación, se debe chequear el modo de funcionamiento del módem mediante el comando:

```
AT+QCFG="usbnet"
```

Al presionar 'Enter' se debería recibir +QCFG: "usbnet", 0 por pantalla, aunque lo que se pretende es que en lugar de un '0' sea un '1', ya que es lo que hace que el módem esté en modo ECM. Generalmente se va a recibir un '0', pero si por lo que sea, se obtiene un '1', el siguiente comando no va a ser necesario, aunque si se hace tampoco afecta a nada.

```
AT+QCFG="usbnet",1
```

Al introducir este comando el módem se debería reiniciar automáticamente, pero si no fuese así, es necesario forzar el reinicio de la siguiente manera:

```
AT+CFUN=1,1
```

Al lanzar el comando hay que esperar unos segundos hasta que aparezca una ventana en la que ponga Cannot open /dev/ttyUSB2!. Esta se irá y empezará a aparecer en la consola información hasta que el módem se reinicie por completo.

En el momento que pase esto, se va a volver a lanzar un AT para ver si se obtiene un OK y así poder proceder con la configuración de la APN de la tarjeta SIM mediante el siguiente comando:

```
AT+CGDCONT=1,"IP","YOUR_APN"
```

Aquí, exactamente igual que para QMI, el campo YOUR_APN se va a tener que sustituir por la APN correspondiente, que en el caso de este proyecto va a ser oaï . ipv4.

Una vez que se haya hecho esto hay que reiniciar otra vez el módem de la misma manera que se hizo antes; es decir:

```
AT+CFUN=1,1
```

Tras esperar a que se reinicie, ya se va a poder salir de minicom presionando ‘Ctrl+A’ y luego ‘Z’, si se ha hecho bien, aparecerá la ventana de ayuda de minicom. Una vez aquí, hay que pulsar ‘X’ y confirmar que quieres salir presionando ‘Enter’.

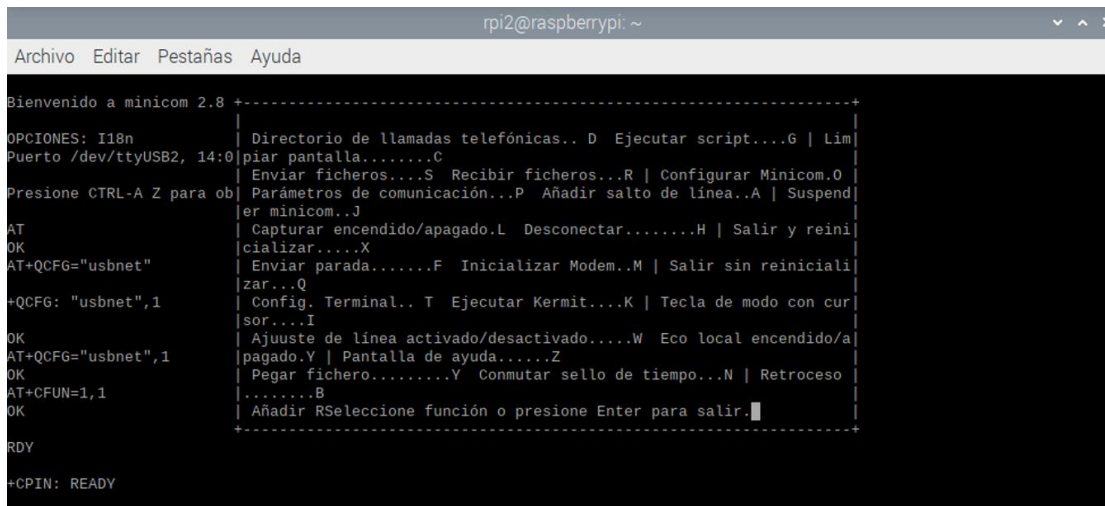


Figura 3. 15.- Ventana de ayuda de minicom (ECM).



Figura 3. 16.- Salir de minicom (ECM).

Ahora sólo queda reiniciar la RPi con `$ sudo reboot` y comprobar que se tiene una dirección IP asignada en la interfaz `usb0`, mediante el comando:

```
$ ifconfig usb0
```



```
rpiwifi@raspberrypi:~ $ ifconfig usb0
usb0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.225.35 netmask 255.255.255.0 broadcast 192.168.225.255
    inet6 fe80::f83e:7c02:2c5c:81fe prefixlen 64 scopeid 0x20<link>
    ether 3e:15:d8:4e:a3:67 txqueuelen 1000 (Ethernet)
    RX packets 230 bytes 83551 (81.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 298 bytes 47257 (46.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figura 3. 17.- ifconfig de la interfaz usb0 (ECM).

También puede resultar interesante confirmar que el módem esté usando el driver `cdc_ether` en lugar del `qmi_wwan`:

```
$ lsusb
$ lsusb -t
```

```
rpiwifi@raspberrypi:~ $ lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 005: ID 0000:3825 USB OPTICAL MOUSE
Bus 001 Device 004: ID 2c7c:0800 Quectel Wireless Solutions Co., Ltd. RM502Q-AE
Bus 001 Device 003: ID c0f4:01e0 USB usb keyboard
Bus 001 Device 002: ID 2109:3431 VIA Labs, Inc. Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
rpiwifi@raspberrypi:~ $ lsusb -t
/: Bus 02.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/4p, 5000M
/: Bus 01.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/1p, 480M
   |__ Port 1: Dev 2, If 0, Class=Hub, Driver=hub/4p, 480M
      |__ Port 2: Dev 3, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M
      |__ Port 2: Dev 3, If 1, Class=Human Interface Device, Driver=usbhid, 1.5M
      |__ Port 3: Dev 4, If 0, Class=Vendor Specific Class, Driver=option, 480M
      |__ Port 3: Dev 4, If 1, Class=Vendor Specific Class, Driver=option, 480M
      |__ Port 3: Dev 4, If 2, Class=Vendor Specific Class, Driver=option, 480M
      |__ Port 3: Dev 4, If 3, Class=Vendor Specific Class, Driver=option, 480M
      |__ Port 3: Dev 4, If 4, Class=Communications, Driver=cdc_ether, 480M
      |__ Port 3: Dev 4, If 5, Class=CDC Data, Driver=cdc_ether, 480M
      |__ Port 4: Dev 5, If 0, Class=Human Interface Device, Driver=usbhid, 1.5M
```

Figura 3. 18.- Uso del driver `cdc_ether` (ECM).

Ahora queda comprobar que se puede llegar a internet empleando el módulo 5G. Para ello tal y como se ha hecho en el apartado anterior se va a hacer un ping www.google.com:

```
$ ping -I usb0 www.google.com -c 5
```

```
rpiwifi@raspberrypi:~ $ ping -I usb0 www.google.com -c 5
PING www.google.com (216.58.215.164) from 192.168.225.35 usb0: 56(84) bytes of data.
64 bytes from mad41s07-in-f4.1e100.net (216.58.215.164): icmp_seq=1 ttl=113 time=39.9 ms
64 bytes from mad41s07-in-f4.1e100.net (216.58.215.164): icmp_seq=2 ttl=113 time=37.4 ms
64 bytes from mad41s07-in-f4.1e100.net (216.58.215.164): icmp_seq=3 ttl=113 time=47.1 ms
64 bytes from mad41s07-in-f4.1e100.net (216.58.215.164): icmp_seq=4 ttl=113 time=34.3 ms
64 bytes from mad41s07-in-f4.1e100.net (216.58.215.164): icmp_seq=5 ttl=113 time=36.3 ms

--- www.google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4006ms
rtt min/avg/max/mdev = 34.332/38.992/47.088/4.430 ms
```

Figura 3. 19.- Comprobación de conexión del módulo 5G (ECM).



Por último, queda comentar que tanto para deshabilitar como para habilitar la interfaz usb0 se pueden emplear los siguientes comandos respectivamente:

```
$ sudo ip link set usb0 down
```

```
$ sudo ip link set usb0 up
```

Con esto el módem ya estaría completamente configurado y listo para funcionar. Hay que tener en cuenta que para conectarse mediante él se debe quitar la red WiFi, si la hubiese, ya que las RPi por defecto emplean la mejor conectividad de entre todas las existentes.

En la siguiente imagen se va a mostrar todo el proceso de configuración de minicom junto; el cual va a servir como apoyo por si algo no hubiese quedado claro en la explicación previa.



```
Archivo Editar Pestañas Ayuda
rpi2@raspberrypi: ~
Bienvenido a minicom 2.8
OPCIONES: I18n
Puerto /dev/ttyUSB2, 14:08:26
Presione CTRL-A Z para obtener ayuda sobre teclas especiales
AT
OK
AT+QCFG="usbnet"
+QCFG: "usbnet",1
OK
AT+QCFG="usbnet",1
OK
AT+CFUN=1,1
OK
RDY
+CPIN: READY
+QUSIM: 1
AT
OK
+CFUN: 1
+QIND: SMS DONE
+QIND: PB DONE
AT+CGDCONT=1,"IP","oai.ipv4"
OK
AT+CFUN=1,1
OK
RDY
+CPIN: READY
+QUSIM: 1
+CFUN: 1
+QIND: SMS DONE
AT
OK
+QIND: PB DONE
AT
OK
[
```

Figura 3. 20.- Configuración de minicom (ECM).

3.3.2.- Configuración de la RPi para compartir la conexión a la red 5G mediante WiFi

Tras haber realizado una de las dos configuraciones anteriores y haber comprobado el correcto funcionamiento del módulo 5G, se va a proceder con la configuración de la RPi para que desarrolle la función de un router, compartiendo la señal de la red 5G proporcionada por la SIM; esta configuración también puede ser conocida como 5G *Hotspot* si se pretende buscar por Internet. [15] [16]



Para comenzar, se va a volver a abrir una terminal en la RPi y se va a comenzar por lanzar un comando para actualizar el sistema operativo, por si acaso no estuviese en la última versión:

```
$ sudo apt update && sudo apt upgrade -y
```

Tras esperar a que se complete el proceso de actualización es recomendable hacer un reinicio del sistema:

```
$ sudo reboot
```

Esto se hace por si acaso luego surgieran algunas incompatibilidades a la hora de lanzar otros comandos, aunque no sea completamente necesario.

Una vez se ha comentado esto y para empezar con la configuración, se va a instalar una nueva librería necesaria para llevar a cabo esta función:

```
$ sudo apt install hostapd dnsmasq -y
```

Cuando acabe el proceso de instalación, va a ser necesario detener los servicios de la librería que se acaba de instalar, ya que por defecto se inician automáticamente.

```
$ sudo systemctl stop hostapd dnsmasq
```

Una vez se ha hecho esto, se va a comenzar con la configuración de la red WiFi, para ello se debe abrir el fichero `dhcpcd.conf`.

```
$ sudo nano /etc/dhcpcd.conf
```

Dentro de este fichero y al final del todo, se van a tener que añadir las siguientes líneas para darle una IP estática a la interfaz y denegarlas para que con cada conexión se les asignen otras direcciones IP distintas.

```
interface wlan0
static ip_address=192.168.10.10/24
denyinterfaces wlan0 usb0
```

En la siguiente imagen se puede observar con más detalle este proceso para evitar que queden dudas:

```
rpiwifi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
GNU nano 5.4 /etc/dhcpd.conf
option interface_mtu
# Most distributions have NTP support.
#option ntp_servers
# A ServerID is required by RFC2131.
require dhcp_server_identifier
# Generate SLAAC address using the Hardware Address of the interface
#slaac hwaddr
# OR generate Stable Private IPv6 Addresses based from the DUID
slaac private
# Example static IP configuration:
#interface eth0
#static ip_address=192.168.0.10/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
#static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1
# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0
#static ip_address=192.168.1.23/24
#static routers=192.168.1.1
#static domain_name_servers=192.168.1.1
# fallback to static profile on eth0
#interface eth0
#fallback static_eth0
interface wlan0
static ip_address=192.168.10.10/24
denyinterfaces wlan0 usb0
^G Ayuda ^O Guardar ^W Buscar ^K Cortar ^T Ejecutar ^C Ubicación M-U Deshacer
^X Salir ^R Leer fich. ^\ Reemplazar ^U Pegar ^J Justificar ^_ Ir a línea M-E Rehacer
```

Figura 3. 21.- Configuración del fichero dhcpd.conf (WiFi).

La IP que se le asigne no debería generar ningún problema, pero por si acaso, se recomienda emplear alguna que no se utilice en las redes con las que se está desarrollando el proceso, ya que luego pueden surgir errores.

Hay que tener en cuenta que la interfaz empleada por el módulo 5G es la usb0, si por lo que sea se quisiese realizar este proyecto con otra interfaz, únicamente habría que cambiar todos los usb0 que vayan apareciendo en el proceso por la interfaz deseada, ya sea eth0, wwan0...

Para proseguir, se debe configurar el servidor DHCP (dnsmasq), guardando la configuración vieja primero, y luego modificando el fichero de configuración que esté vacío. Esto se hace empleando los siguientes comandos:

```
$ sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.orig
$ sudo nano /etc/dnsmasq.conf
```

A este fichero se le van a añadir dos líneas de comandos para configurar el rango de IPs dadas mediante el protocolo DHCP. Aquí se deben emplear IPs que se encuentren dentro de la misma subred que la que se había configurado como estática previamente, sin incluirla a ella misma, tal y como se va a hacer a continuación:



```
interface=wlan0  
dhcp-range=192.168.10.11,192.168.10.30,255.255.255.0,24h
```

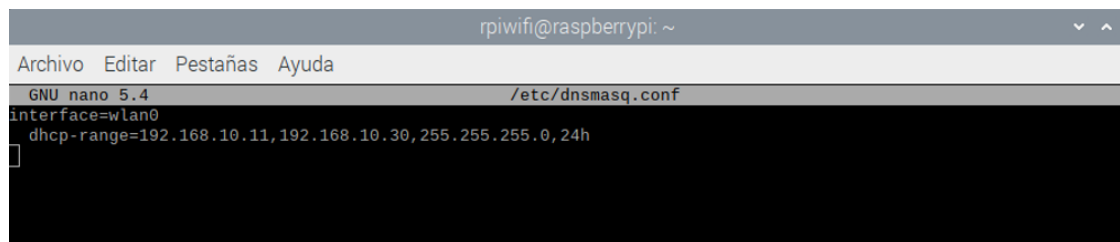


Figura 3. 22.- Configuración del fichero dnsmasq.conf (WiFi).

Tras haber hecho esto, se va a proceder con la configuración del *Access Point* o punto de acceso para poder encontrar la red WiFi que va a generar la RPi, y así conectarse a ella. En este caso se va a hacer una configuración con protección por contraseña, al igual que un router convencional, para que únicamente quien sepa la contraseña pueda usar los datos de la red 5G que se va a generar.

Para llevar a cabo esto, es necesario crear un nuevo fichero llamado `hostapd.conf`:

```
$ sudo nano /etc/hostapd/hostapd.conf
```

Dentro de este fichero se van a añadir todos los comandos de configuración que se muestran a continuación, únicamente habría que modificar el nombre de la red (`ssid`) y la contraseña que se le quiera poner (`wpa_passphrase`).

```
interface=wlan0  
bridge=br0  
hw_mode=g  
channel=7  
wmm_enabled=0  
macaddr_acl=0  
auth_algs=1  
ignore_broadcast_ssid=0  
wpa=2  
wpa_key_mgmt=WPA-PSK  
wpa_pairwise=TKIP  
rsn_pairwise=CCMP
```

ssid=NETWORK

wpa_passphrase=PASSWORD

En la siguiente imagen se puede la configuración específica de este proyecto:

```
rpiwifi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
GNU nano 5.4 /etc/hostapd/hostapd.conf
interface=wlan0
bridge=br0
hw_mode=g
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
ssid=RPiThin5G
wpa_passphrase=catedrathin5g

```

Figura 3. 23.- Configuración del fichero *hostapd.conf* (WiFi).

Como se puede observar, el nombre de la red en este caso particular es “RPiThin5G” y la contraseña que se le ha puesto es “catedrathin5g”.

Ahora hay que decirle a la RPi dónde puede encontrar este fichero de configuración, para ello, dentro el fichero “hostapd” va a haber que sustituir una línea tal y cómo se muestra a continuación:

```
$ sudo nano /etc/default/hostapd
```

Hay que buscar la parte del código que pone `#DAEMON_CONF=""` y cambiarlo por:
`DAEMON_CONF="/etc/hostapd/hostapd.conf"`

```
rpiwifi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
GNU nano 5.4 /etc/default/hostapd
# Defaults for hostapd initscript
#
# WARNING: The DAEMON_CONF setting has been deprecated and will be removed
# in future package releases.
#
# See /usr/share/doc/hostapd/README.Debian for information about alternative
# methods of managing hostapd.
#
# Uncomment and set DAEMON_CONF to the absolute path of a hostapd configuration
# file and hostapd will be started during system boot. An example configuration
# file can be found at /usr/share/doc/hostapd/examples/hostapd.conf.gz
DAEMON_CONF="/etc/hostapd/hostapd.conf"
#
# Additional daemon options to be appended to hostapd command:-
#   -d show more debug messages (-dd for even more)
#   -K include key data in debug messages
#   -t include timestamps in some debug messages
#
# Note that -B (daemon mode) and -P (pidfile) options are automatically
# configured by the init.d script and must not be added to DAEMON_OPTS.
#
#DAEMON_OPTS=""
[ 23 líneas leídas ]
AG Ayuda      AO Guardar   AW Buscar   AK Cortar   AT Ejecutar   AC Ubicación M-U Deshacer
AX Salir     AR Leer fich. AU Reemplazar AU Pegar    AJ Justificar A Ir a línea M-E Rehacer
```

Figura 3. 24.- Configuración del fichero hostapd (WiFi).

El siguiente paso a realizar es reenviar la conexión a internet a través de un *bridge*, para ello es necesario instalar otra librería:

```
$ sudo apt-get install bridge-utils -y
```

Cuando el proceso de instalación haya concluido se puede ya crear el *bridge* que se va a emplear y decirle cual es la interfaz que va a enviar los datos y, por consiguiente, introducirlos en el bridge. En el caso de este proyecto es *usb0*.

```
$ sudo brctl addbr br0
```

```
$ sudo brctl addif br0 usb0
```

El nombre del bridge en este caso es *br0*, aunque se lo podría asignar otro nombre sin problemas, o incluso, crear varios bridges, dependiendo de las necesidades del proyecto, aunque aquí no se haya llevado a cabo.

Tras haberlo creado, toca configurarlo modificando el fichero “*interfaces*”:

```
$ sudo nano /etc/network/interfaces
```

Para ello, hay que añadir al final del mismo una serie de comandos, para conseguir que después de un reinicio siga realizando la función de compartir WiFi la RPi y no se desconfigure.

```
auto br0
iface br0 inet manual
bridge_ports usb0 wlan0
```

Este proceso se observa más detalladamente en la siguiente imagen:

```

rpiwifi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
GNU nano 5.4 /etc/network/interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source /etc/network/interfaces.d/*

auto br0
iface br0 inet manual
bridge_ports usb0 wlan0

```

Figura 3. 25.- Configuración del fichero interfaces (WiFi).

Únicamente queda por comprobar si el servicio del hostapd está enmascarado, y si lo está habría que desenmascararlo y habilitarlo de la siguiente manera:

```
$ sudo systemctl unmask hostapd
$ sudo systemctl enable hostapd
```

Por último, se reinicia el sistema y ya debería estar funcionando perfectamente sin interrupción.

```
$ sudo reboot
```



3.3.3.- Configuración de la RPi para compartir la conexión a la red 5G mediante Ethernet

En este apartado se va a explicar el proceso que hay que realizar para compartir la conectividad 5G que proporciona el módulo que se está empleando a través de un cable Ethernet. Este método es bastante más sencillo que el anterior, y al ser por cable proporciona una mejor señal. [17] [18] [19]

En primer lugar, tal y cómo se hizo para el caso de la configuración anterior, va a ser necesario hacer una actualización del sistema por si acaso.

```
$ sudo apt-get update
```

Y tras realizarla se deben instalar dos librerías: una que ya se empleó anteriormente y resultará conocida ya que es dnsmasq, y una segunda que va a servir como firewall de la red; la cual es iptables.

```
$ sudo apt-get install dnsmasq
```

```
$ sudo apt-get install iptables
```

Cuando termine el proceso de instalación, se van a empezar a configurar algunos archivos, comenzando por dhcpd.conf:

```
$ sudo nano /etc/dhcpd.conf
```

Dentro de este y al final del fichero se tienen que añadir dos comandos para asignarle una dirección IP estática a la interfaz eth0; la cual es la interfaz del puerto Ethernet en la RPi 4.

```
interface eth0
```

```
static ip_address=192.168.4.1/24
```

Esto se puede observar con mayor detalle en la imagen que se muestra a continuación, aunque el proceso va a ser similar al realizado en el apartado anterior.



```
rpieth@raspberrypi: ~
GNU nano 5.4 /etc/dhcpd.conf
# A list of options to request from the DHCP server.
option domain_name_servers, domain_name, domain_search, host_name
option classless_static_routes
# Respect the network MTU. This is applied to DHCP routes.
option interface_mtu

# Most distributions have NTP support.
option ntp_servers

# A ServerID is required by RFC2131.
require dhcp_server_identifier

# Generate SLAAC address using the Hardware Address of the interface
slaac hwaddr
# OR generate Stable Private IPv6 Addresses based from the DUID
slaac private

# Example static IP configuration:
#interface eth0
#static ip_address=192.168.0.10/24
#static ip_address=f651:42f8:caae:d92e::ff/64
#static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1

# It is possible to fall back to a static IP if DHCP fails:
# define static profile
#profile static_eth0
#static ip_address=192.168.1.23/24
#static routers=192.168.1.1
#static domain_name_servers=192.168.1.1

# fallback to static profile on eth0
#interface eth0
#fallback static_eth0

interface eth0
static ip_address=192.168.4.1/24
]
```

Figura 3. 26.- Configuración del fichero dhcpd.conf (Ethernet).

Tras haber guardado el fichero, el siguiente paso es realizar una copia de seguridad de los datos existentes en el fichero dnsmasq.conf a el fichero dnsmasq.conf.bak, por si acaso surgiese algún inconveniente. Para ello se emplea el siguiente comando:

```
$ sudo mv /etc/dnsmasq.conf /etc/dnsmasq.conf.bak
```

Y tras haber hecho esto, ya se puede configurar el fichero dnsmasq.conf sin miedo a estropear la configuración de la RPi.

```
$ sudo nano /etc/dnsmasq.conf
```

Al abrirse el fichero se puede observar que está vacío, es por ello que en él hay que añadir las siguientes líneas:

```
interface=eth0
```

```
dhcp-range=192.168.4.8,192.168.4.250,255.255.255.0,24h
```




```
rpieth@raspberrypi: ~
GNU nano 5.4 /etc/dnsmasq.conf
interface=eth0
dhcp-range=192.168.4.8,192.168.4.250,255.255.255.0,24h
```

Figura 3. 27.- Configuración del fichero dnsmasq.conf (Ethernet).

Tal y cómo se puede observar en esta imagen, se le está asignando un rango de direcciones IP, esto es por si se quisiesen conectar varios dispositivos mediante esta conexión Ethernet, aunque sería necesario emplear un hub o switch intermedio para aumentar el número de puertos Ethernet.

Ya solamente quedan dos pasos más por realizar y ya estaría la configuración; el primero de ellos consiste en modificar el archivo sysctl.conf, descomentando una línea que va a habilitar el reenvío de datos mediante IPv4.

```
$ sudo nano /etc/sysctl.conf
```

La línea que hay que descomentar es la que contiene el comando `net.ipv4.ip_forward=1`, aunque se puede ver mejor en la siguiente imagen:



```
GNU nano 5.4 /etc/sysctl.conf
# /etc/sysctl.conf - Configuration file for setting system variables
# See /etc/sysctl.d/ for additional system variables.
# See sysctl.conf (5) for information.
#
#kernel.domainname = example.com
# Uncomment the following to stop low-level messages on console
#kernel.printk = 3 4 1 3
#####
# Functions previously found in netbase
#
# Uncomment the next two lines to enable Spoof protection (reverse-path filter)
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
#net.ipv4.conf.default.rp_filter=1
#net.ipv4.conf.all.rp_filter=1
# Uncomment the next line to enable TCP/IP SYN cookies
# See http://lwn.net/Articles/277146/
# Note: This may impact IPv6 TCP sessions too
#net.ipv4.tcp_syncookies=1
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
# Uncomment the next line to enable packet forwarding for IPv6
# Enabling this option disables Stateless Address Autoconfiguration
# based on Router Advertisements for this host
#net.ipv6.conf.all.forwarding=1
#####
# Additional settings - these settings can improve the network
# security of the host and prevent against some network attacks
# including spoofing attacks and man in the middle attacks through
# redirection. Some network environments, however, require that these
```

Figura 3. 28.- Configuración del fichero sysctl.conf (Ethernet).

El segundo, y último paso a realizar, consiste en incluir información en el fichero rc.local para determinar el flujo de los datos de origen a destino. En el caso de este proyecto, desde usb0 a eth0, aunque si se quisiese compartir desde el WiFi al Ethernet, por ejemplo, bastaría con cambiar donde pone usb0 por wlan0.

```
$ sudo nano /etc/rc.local
```

Una vez dentro del fichero hay que añadir el siguiente comando justo encima de `exit 0`:

```
iptables -t nat -A POSTROUTING -o usb0 -j MASQUERADE
```

La siguiente imagen sirve para poder comprender esto de una mejor forma.



```
rchivo  Editor  Pestañas  Ayuda
rpieth@raspberrypi: ~
GNU nano 5.4 /etc/rc.local *
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
# By default this script does nothing.
#
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
  printf "My IP address is %s\n" "$_IP"
fi

iptables -t nat -A POSTROUTING -o usb0 -j MASQUERADE

exit 0
```

Figura 3. 29.- Configuración del fichero rc.local (Ethernet).

Con esto ya concluiría el proceso de configuración de esta red, e incluso apagando la RPi o reiniciándola, cuando se conecte el cable Ethernet a un dispositivo, este va a empezar a compartir los datos.

4.- PRUEBAS DE CONEXIÓN

En este capítulo se van a mostrar y comentar los resultados obtenidos de las pruebas de conexión que se han realizado con ambas configuraciones, tras haberlas configurado cómo se ha mostrado en el capítulo anterior.

Para realizar este proceso se han empleado principalmente dos dispositivos; los cuales son: un teléfono móvil y un ordenador portátil, ya que la para probar la conectividad a través de Ethernet sólo se tenía puerto en el ordenador.

4.1.- Conectividad de la red WiFi

En primer lugar se va a testear la conexión mediante la red WiFi, por seguir un poco el orden de desarrollo del proyecto. Para ello, teniendo la RPi funcionando en el modo WiFi, se va a ir a la parte de selección de redes WiFi y en esta debería aparecer la RPi con el nombre que se le puso en el apartado ssid durante la configuración del fichero hostapd.conf, que en el caso de este proyecto es “RPiThin5G”. Esto se puede observar en la siguiente imagen:

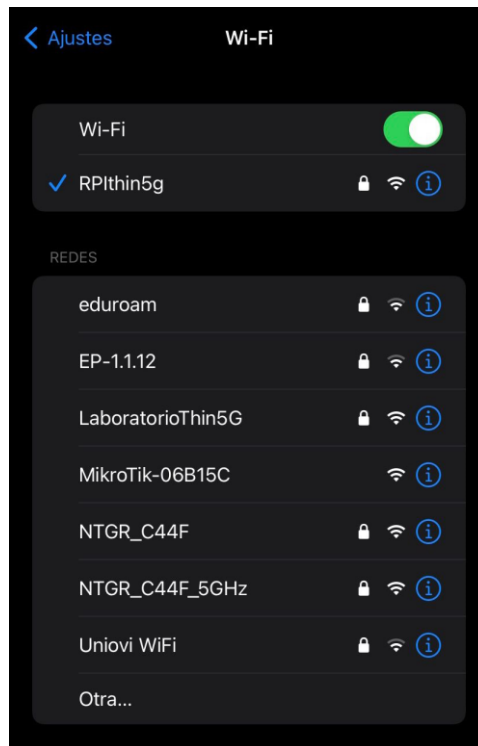


Figura 4. 1.- Conexión a red WiFi.



Hay que tener en cuenta que cuando se seleccione la red, esta va a pedir la contraseña que se le había puesto previamente, y si todo está bien y la red está activa se podrá conectar de manera exitosa como se acaba de mostrar.

Las tasas de datos no son extremadamente altas, pero funciona bastante bien e incluso conectando varios dispositivos a la vez, mantiene sus capacidades muy bien, sin mucha degradación.

Las pruebas de velocidad de la red se muestran en las siguientes imágenes, tanto para un dispositivo móvil, como desde el ordenador portátil:

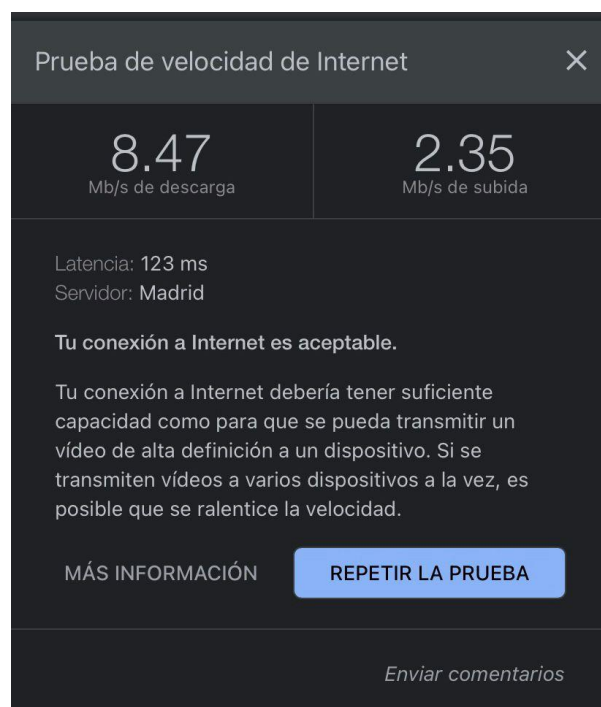


Figura 4. 2.- Test de velocidad de Internet de la red WiFi desde un dispositivo móvil.

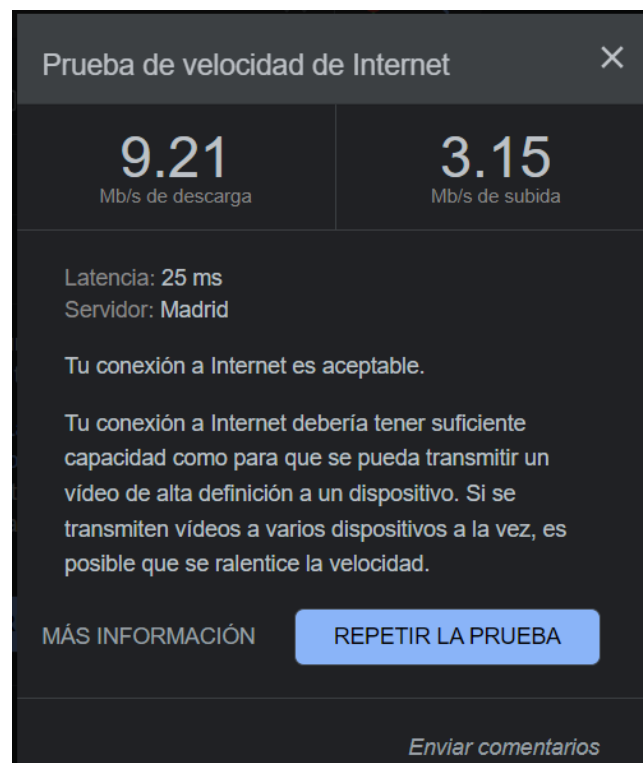


Figura 4. 3.- Test de velocidad de Internet de la red WiFi desde un ordenador portátil.

Tras haber realizado varias pruebas, y a la vista de las imágenes, se puede concluir que la velocidad de internet ronda los 9 Mbps de subida y los 3 Mbps de bajada. Es una conectividad aceptable, aunque tampoco increíblemente buena, ya que para muchas acciones se nota que va lenta la red.

También hay varios problemas con esta red que deben ser comentados, ya que la mayoría de las veces la red se cae y no permite que se conecten dispositivos a ella, aunque no es un problema de la configuración mostrada en el apartado anterior, ya que al probar con una tarjeta de datos personal, el sistema funcionaba correctamente y sin caídas al conectar varios dispositivos, por lo que es un problema interno de la red. Además, las RPis no funcionan realmente bien conectadas a una red 5G como la presente en el laboratorio donde se realizó el proyecto, y muchas veces surgen problemas de conexión o caídas de la red que reiniciando se suelen solucionar, pero al cabo de un rato acaban volviendo a suceder en muchas ocasiones.

Si por lo que sea se quisiese probar con otra tarjeta SIM de datos, únicamente habría que desconectar el módulo 5G, extraer la SIM que tiene, colocar la nueva y volver a conectarlo. Una vez hecho esto, sólo habría que reconfigurar el módem en el modo de

operación que se estaba empleando previamente (siempre recomendable la interfaz usb0), ya que habrá que repetir los comandos de configuración hasta llegar al que pide el APN del dispositivo, donde habría que poner el asociado a la nueva SIM.

Una cosa curiosa y a la vez interesante, es que no sólo el módem guarda la configuración, sino que emplea a la SIM como un dispositivo a parte y al meter los comandos, se configura de manera independiente, por lo que es importante configurarla con su APN u otras acciones que se hayan realizado sobre ella antes de extraerla y volver a conectarla a un terminal móvil, si e s el caso, ya que este no la va a reconocer.

Tras determinar que era un problema de red y no de la configuración del módem o la Raspberry, se van a mostrar a continuación imágenes de cómo, a pesar de estar conectado, surgían errores de acceso a internet, además de la configuración interna de esta WiFi que se muestran gracias a la interfaz proporcionada por el dispositivo móvil.

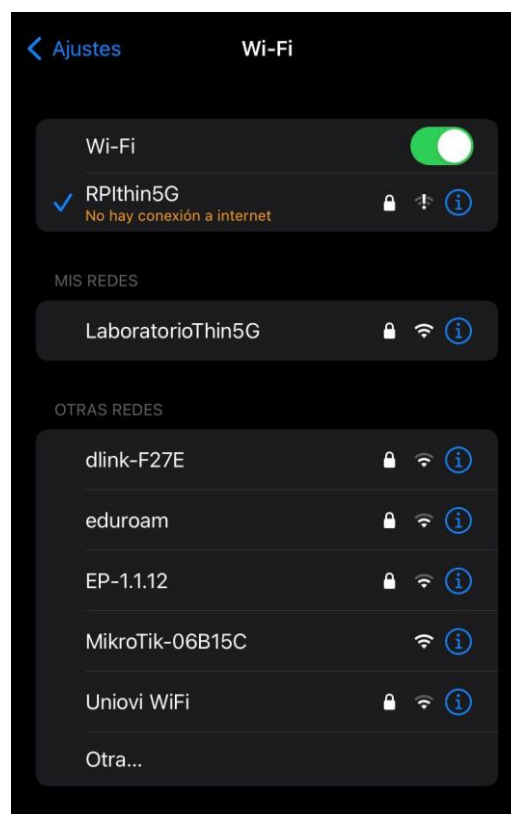


Figura 4. 4.- Fallo en la conectividad a la red WiFi desde un dispositivo móvil.

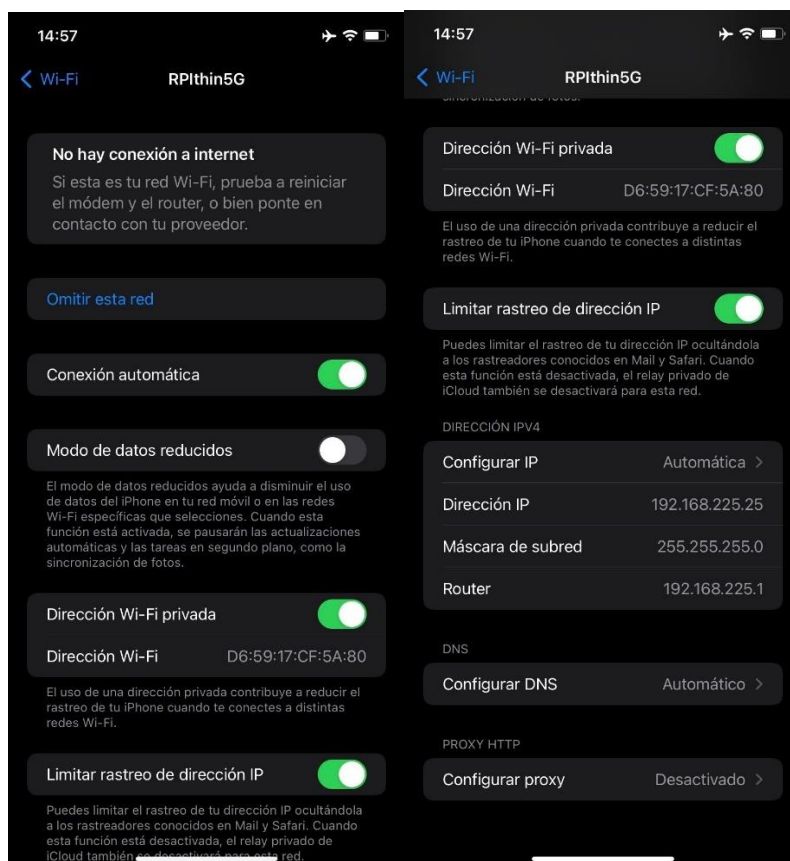


Figura 4. 5.- Características de la red WiFi desde un dispositivo móvil.

En esta segunda imagen se puede observar como tiene asignado una dirección IP dentro del rango de las que proporciona el módulo 5G (192.168.225.25) y también, cuál es la dirección IP de la puerta de enlace o la propia del módulo; en este caso es la 192.168.225.1.

4.2.- Conectividad de la red Ethernet

Una vez que se han comentado las pruebas realizadas sobre la red WiFi, con los inconvenientes que han ocurrido, es el turno de realizar las comprobaciones sobre la red que va a proporcionar conectividad mediante Ethernet.

En primer lugar, la conectividad es mucho más sencilla. Sólo es necesario conectar el cable Ethernet a la RPi y al dispositivo, en este caso un ordenador portátil, y ya se empezará a tener conectividad a Internet sin necesidad de claves u otra acción diferente. Da igual que se haga un reinicio en la RPi o en el ordenador, que este dispositivo va a ser robusto y va a volver a dar conectividad sin ningún problema.

Tal y cómo se ha comentado, se configuró un rango de IPs para conectar varios dispositivos, aunque no ha sido posible realizar estas comprobaciones en el proyecto. Por lo que únicamente se ha probado con un dispositivo, dando los resultados que se muestran a continuación:



Figura 4. 6.- Reconocimiento de la red Ethernet desde un ordenador portátil.

Esta imagen se ha realizado tras conectar el cable Ethernet al ordenador portátil. La conectividad suele ser casi instantánea, aunque en alguna ocasión, es necesario esperar algunos segundos hasta que se sincronice y reconozca el dispositivo.

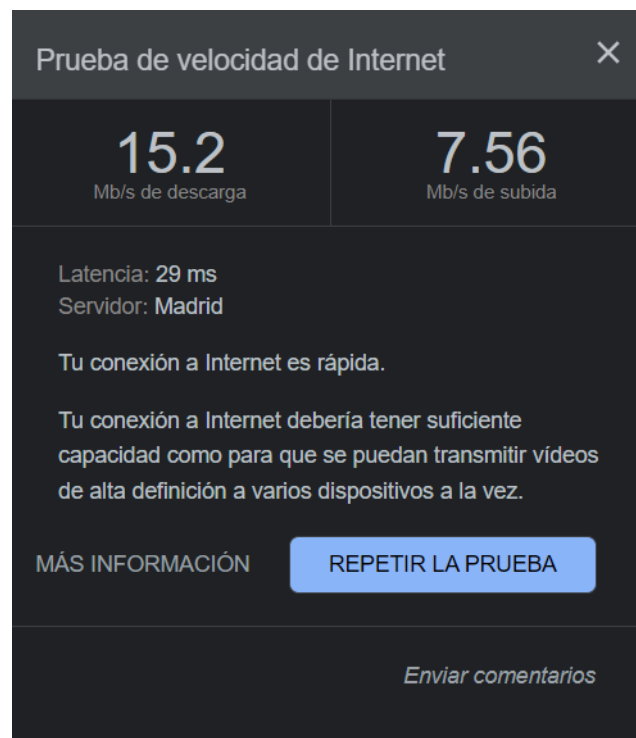


Figura 4. 7.- Test de velocidad de Internet de la red Ethernet desde un ordenador portátil.

Como se puede observar, los resultados de velocidad de la red cableada, tal y como era de esperar, son mucho mejores que para el caso de la red WiFi; permitiendo navegar por internet o realizar cualquier actividad sin ningún problema. Además, la latencia sigue siendo bastante buena y no se notan retardos en la señal.



5.- CONCLUSIONES

Durante el desarrollo de este proyecto se han estudiado dos de las tecnologías de transmisión de datos más importantes en la actualidad; las cuales son LoRa y 5G, ya que la oportunidad que ofrecía el laboratorio de 5G de poder realizar un proyecto con ambas y conocer su funcionamiento era muy interesante.

Tras explicar los conceptos básicos para poder comprenderlas, se han elaborado unas redes IoT para la monitorización de datos proporcionados por diferentes sensores, aunque el grueso de esta parte se haya desarrollado en el proyecto conjunto.

Mientras se iban desarrollando las redes se han visto las dificultades, beneficios y problemas surgidos; es por esto por lo que surge la idea de desarrollar un sistema que permita conectarse a la red 5G sin complicaciones, ya que es mucho más difícil y requiere tener muchas más cosas en cuenta que el caso de LoRa. Por este motivo y barajando las opciones disponibles, se llega a la conclusión de emplear una RPi como método de conexión a la red 5G, empleando un módulo 5G que permita la conectividad a través del puerto USB para así no utilizar el puerto Ethernet y poder emplearlo para ser un método de conectarse a la red 5G; además de por WiFi.

Para ello es necesario configurar la tarjeta SIM de datos y posteriormente los módulos, ya que el módem por defecto viene configurado en QMI pero no permite acceso a internet.

Estos módems tiene dos configuraciones posibles: QMI , tal y como se acaba de mencionar, y ECM. Entonces, estudiando ambos e informándose sobre la utilidad se ha decidido por el modo ECM e implementando la interfaz usb0; ya que además de sencillez y compatibilidad, permite que se mantenga la configuración después de un reinicio de la RPi o al desconectar y volver a conectar el módulo 5G.

Tras tener claro el modo en el que se va a utilizar el módem y configurarlos, el siguiente paso fue configurar las RPis para compartir la red 5G mediante WiFi y Ethernet. En un principio, se intentó hacer con una sólo RPi, pero algunas de las configuraciones necesarias para un método pisaban al otro y siempre había uno que no funcionaba bien o los en la mayoría de ocasiones. Por este motivo se tuvieron que emplear dos RPis diferentes.

En cuanto a al método para compartir la red 5G a través de WiFi, no ha dado resultados realmente buenos por motivos de la red, ya que al conectar un dispositivo se volvía inestable y se caía aunque lo reconociese y en el dispositivo pusiera que había conexión. Se ha verificado que la configuración es correcta empleando una SIM con conectividad 4G LTE de datos, ya que únicamente configurando la SIM en el módem, sin



tocar la configuración de modo empleado para compartir WiFi, se obtenía conexión de red y era posible navegar por internet sin ningún problema.

Por otro lado, el método de conexión a la red 5G mediante cable fue mucho más sencillo y dio resultados realmente buenos. Además, la velocidad de la conexión incluso después de un reinicio es muy buena.

Por último, y tras verificar las pruebas de conexión, se ha desarrollado un estudio financiero del proyecto; teniendo en cuenta lo que podría costar realizarlo a nivel profesional por un ingeniero junior.

En trabajos futuros queda pendiente el poder implementarlo todo sobre una misma RPi y que funcione siempre correctamente, ya que hay un software, el cual se ha mencionado previamente, que funciona realmente bien y es muy recomendable para realizar este tipo de proyectos con RPis. El problema es que al emplearlo pierdes la interfaz propia de Raspberry Pi OS y únicamente tienes acceso a una ventana de comandos, por lo que la RPi perdería cualquier otra función que se le quiera dar, ya que con las configuraciones realizadas en este proyecto, siguen completamente funcionales. Por otra parte, no había mucha información relacionada con este software y todas las pruebas que se realizaron salieron mal, e incluso había alguna incompatibilidad con el hardware empleado, aunque se encuentra en continua evolución y se esperan más actualizaciones para implementaciones 5G.

Por último, comentar que el sector IoT y en particular, las tecnologías de LoRa y 5G son el presente y el futuro en nuestra sociedad, pudiendo tener infinidad de aplicaciones en muchos proyectos. Por este motivo el proyecto ha resultado muy atractivo y llevadero, además de interesante, ya que se ha aprendido mucho con su realización.



REFERENCIAS

- [1] FERRER, Vicent. Vicentferrer.com [En línea]. «Qué es Lora y Lorawan». <[https://vicentferrer.com/lora-lorawan/#:~:text=LoRa%20significa%20%C2%ABlargo%20alcance%C2%BB%20\(%2C%20y%20bajo%20consumo%20de%20energ%C3%ADa](https://vicentferrer.com/lora-lorawan/#:~:text=LoRa%20significa%20%C2%ABlargo%20alcance%C2%BB%20(%2C%20y%20bajo%20consumo%20de%20energ%C3%ADa)> [Consulta: 22 de mayo de 2022].
- [2] Alai Secure [En línea]. «LORA: ¿Qué es y para qué sirve?». <<https://alaisecure.es/glosario/lora-que-es-y-para-que-sirve>> [Consulta: 25 de mayo de 2022].
- [3] GÓMEZ ORTIZ, G. J. (Julio de 2017) [En línea]. «Diseño y fabricación de capa de modulaciones basada en LoRa para plataforma modular de redes de sensores inalámbricas». Tutor: PORTILLA BERRUECO, J. Proyecto Final de Carrera. Madrid: Universidad Politécnica de Madrid. <<http://www.etsit.upm.es/escuela/unidades-administrativas-y-servicios/biblioteca/buscar/tesis-y-proyectos.htmlhttps://uvadoc.uva.es/handle/10324/30636>> [Consulta: 25 de mayo de 2022].
- [4] The Things Network [En línea]. «LoRaWAN». <<https://www.thethingsnetwork.org/docs/lorawan/>> [Consulta: 27 de mayo de 2022].
- [5] The Things Stack [En línea]. «Devices». <<https://www.thethingsindustries.com/docs/devices/>> [Consulta: 27 de mayo de 2022].
- [6] AlfaIoT [En línea]. «LORA». <<https://alfaiot.com/tecnologias-iot/lora/>> [Consulta: 25 de mayo de 2022].
- [7] RF Wireless World [En línea]. «LoRaWAN MAC layer message formats | LoRaWAN MAC commands». <<https://www.rfwireless-world.com/Tutorials/LoRaWAN-MAC-layer-inside.html>> [Consulta: 26 de mayo de 2022].
- [8] CAÑIZO OTERO, P. (Junio de 2022). «Estudio y comparativa de redes IoT desplegadas sobre LoRA y 5G: estudio de redes». Tutor: GONZÁLEZ AYESTARÁN, R. Proyecto de fin de Máster. Asturias: Universidad de Oviedo. [Consulta: 1 de julio de 2022].



[9] NILESH GANATRA, A. (Otoño de 2017) [En línea]. «Developments of 5G Technology». Proyecto Final de Carrera. Illinois: Governors State University. <<https://opus.govst.edu/cgi/viewcontent.cgi?article=1106&context=theses>>

[Consulta: 10 de julio de 2022].

[10] DIGI [En línea]. «¿Qué es la arquitectura de red 5G?». <<https://es.digi.com/blog/post/5g-network-architecture#:~:text=El%20n%C3%BAcleo%205G%20utiliza%20una,se%20muestra%20en%20el%20diagrama>> [Consulta: 10 de julio de 2022].

[11] I'MNOVATION [En línea]. «Qué es el 5G y cómo está hiperconectando». <https://www.imnovation-hub.com/es/sociedad/que-es-5g/?gclid=CjwKCAjw2rmWBhB4EiwAiJ0mtRrJUCZl6Op9IBV9PXgKAoyiITEIOT8dft6Nw7402ytd0068rZZpVhoCVXMQAvD_BwE&_adin=02021864894>

[Consulta: 8 de julio de 2022].

[12] Motorpasión [En línea]. «WiFi, un panel a lo Tesla o una consola retro por muy poco dinero: 17 proyectos para usar una Raspberry Pi en tu coche». <<https://www.motorpasion.com/tecnologia/wifi-un-panel-a-lo-tesla-o-una-consola-retro-por-muy-poco-dinero-17-proyectos-para-usar-una-raspberry-pi-en-tu-coche>>

[Consulta: 28 de mayo de 2022].

[13] Labsist [En línea]. «LABISTS Raspberry Pi 4 8GB RAM Starter Kit with 64GB Micro SD Card (Raspbian)». <<https://labists.com/collections/raspberry-pi-4-kit/products/raspberry-pi-4-8gb-ram-starter-kit-with-64gb>> [Consulta: 20 de junio de 2022].

[14] GEERLING, Jeff. jeffgeerling [En línea]. «Using 4G LTE wireless modems on a Raspberry Pi». <<https://www.jeffgeerling.com/blog/2022/using-4g-lte-wireless-modems-on-raspberry-pi>> [Consulta: 2 de junio de 2022].

[15] Sixfab [En línea]. «Setting up a Raspberry Pi as an LTE Hotspot». <<https://docs.sixfab.com/page/setting-up-a-raspberry-pi-as-an-lte-hotspot>>

[Consulta: 10 de junio de 2022].

[16] HERTZ, Daniel. Maker Pro [En línea]. «How to Create a 4G Hotspot With Raspberry Pi». <<https://maker.pro/raspberry-pi/tutorial/how-to-create-a-4g-hotspot-with-raspberry-pi>> [Consulta: 2 de junio de 2022].



[17] YouTube [En línea]. «Sharing the Raspberry Pi's WiFi over the Ethernet Port». <<https://www.youtube.com/watch?v=TtLNue7gzZA&list=LL&index=2>> [Consulta: 16 de junio de 2022].

[18] StackExchange [En línea]. «Share 4G USB Internet on Raspberry Pi 4 over WiFi and Ethernet». <<https://raspberrypi.stackexchange.com/questions/107372/share-4g-usb-internet-on-raspberry-pi-4-over-wifi-and-ethernet>> [Consulta: 11 de junio de 2022].

[19] NAWAWI, Umar. Github [En línea]. «Using a Raspberry Pi 3 as a Wifi access point and bridge». <<https://gist.github.com/umardx/23cc3ad93119151a16f486c7cf01096c>> [Consulta: 3 de junio de 2022].

[20] ComputerWeekly.es [En línea]. «LoRa vs. 5G: ¿Pueden coexistir para la conectividad de red IoT?». <<https://www.computerweekly.com/es/respuesta/LoRa-vs-5G-Pueden-coexistir-para-la-conectividad-de-red-IoT>> [Consulta: 23 de mayo de 2022].

[21] Izertis [En línea]. «Redes de comunicación IoT: de GPRS a 5G». <<https://www.izertis.com/es/-/blog/redes-iot-desde-gprs-sigfox-lora-hasta-5g>> [Consulta: 23 de mayo de 2022].

[22] Superuser [En línea]. «Setting up a wifi hotspot with brctl and hostapd». <<https://superuser.com/questions/709879/setting-up-a-wifi-hotspot-with-brctl-and-hostapd>> [Consulta: 2 de junio de 2022].

[23] OpenWrt [En línea]. «How to use LTE modem in QMI mode for WAN». <<https://openwrt.org/docs/guide-user/network/wan/wwan/lte dongle>> [Consulta: 8 de junio de 2022].

[24] RODRÍGUEZ LALLANA, M. (Junio de 2018) [En línea]. «Diseño de un sensor de temperatura IOT para la red LoRa». Tutora: HERRÁEZ SÁNCHEZ, M. Proyecto Final de Carrera. Valladolid: Universidad de Valladolid. <<https://uvadoc.uva.es/handle/10324/30636>> [Consulta: 29 de mayo de 2022].



ANEXO: PRESUPUESTO

En este anexo se pretende realizar una estimación del coste del proyecto si hubiera sido desarrollado profesionalmente. Para ello se toman en cuenta las siguientes consideraciones iniciales:

- El proyecto lo va a desarrollar una sola persona con un nivel profesional similar al de un ingeniero junior.
- El período de desarrollo del proyecto abarca unos 4 meses, en los que no se ha seguido una jornada de trabajo continua. Es decir, no se ha seguido por ejemplo una jornada de oficina de 8 horas al día, sino que se ha ido realizando según la disponibilidad de tiempo (incluyendo trabajo en sábados, domingos y festivos). Se realizará por tanto una estimación de las horas de trabajo totales.

El presupuesto se va a dividir en dos bloques principales: recursos humanos y recursos materiales. A continuación, se comienza a realizar su descripción.

Recursos humanos:

Se van a desarrollar en función de la estimación de horas empleadas para el desarrollo del proyecto. El resultado total se podría aproximar a unas 60 horas de trabajo totales desde sus inicios hasta el momento de la entrega del proyecto.

Para los costes de los recursos humanos se considera un coste por hora de 22€/h teniendo en cuenta la posición de ingeniero junior; en este precio se incluye el costo mensual de los honorarios del trabajador, los porcentajes correspondientes de pago de seguridad social, desempleo, formación profesional e IRPF. Incluso de forma anual está incluido el pago de pagas extras, vacaciones y las indemnizaciones correspondientes por despido.

Por lo que, en total, tomando las 60 horas estimadas, los gastos en recursos humanos serían (22€/h * 60h): 1320,00 €.

Esto se va a resumir adecuadamente en la siguiente tabla:

Elementos	Costes unitarios (€)	N.º de unidades	Costes totales (€)
Recursos humanos: Ingeniero Junior	1320,00	1	1320,00
Coste total de recursos humanos			1320,00

Tabla A. 1.- Coste de los recursos humanos.



Recursos materiales:

Los recursos materiales se van a desglosar en dos tablas; una que se refiere al hardware y materiales, y otra para el software y consumibles.

Con respecto al hardware y los materiales necesarios para la elaboración de este proyecto, se obtiene la siguiente tabla:

Elementos	Modelo	Costes unitarios (€)	Nº de unidades	Costes totales (€)
Módulos 5G Quectel	RM502Q-AE	462,22	1	462,22
Antenas 5G	-	12,69	2	25,38
Cable ethernet 1m	-	2,46	1	2,46
Cable SMA-MHF4	-	84,78	2	169,56
Adaptador M.2 5G a USB 3.0	-	20,89	1	20,89
Raspberry Pi	Model 4 8GB Kit con tarjeta 64GB	155,99	2	311,80
Coste total del hardware y los materiales				992,31

Tabla A. 2.- Coste del hardware y los materiales.

A continuación, se procede con la exposición de todo el software y los consumibles necesarios para la realización del trabajo mediante la siguiente tabla.

Elementos	Costes unitarios (€/mes)	Periodo de tiempo (mes)	Costes totales (€)
Electricidad consumida	8,00	4	32,00
Conexión a internet	30,00	4	120,00
Tarifa de datos móviles	20,00	4	80,00
Material de oficina: papel, tinta impresora...	7,50	4	30,00
Microsoft 365	-	-	129,00
Coste total del software y los consumibles			391,00

Tabla A. 3.- Coste del software y los consumibles.



Resumen del presupuesto:

Todos los costes totales de cada tipo de recurso se unificarán en la siguiente tabla para determinar el cote total del proyecto.

Recursos	Costes (€)
Recursos humanos	1320,00
Hardware y materiales	992,31
Software y consumibles	391,00
Coste	2703,31
Costes indirectos (5%)	135,17
Coste total	2838,48

Tabla A. 4.- Coste total del proyecto sin IVA.

Resumen final	Coste (€)
Coste total	2838,48
Porcentaje de gastos generales (13%)	369,01
Porcentaje de beneficio industrial (6%)	170,31
Coste total sin IVA	3377,80
Porcentaje de IVA (21%)	709,34
Coste final del proyecto (IVA incluido)	4087,14

Tabla A. 5.- Coste final del proyecto.

Por tanto, el presupuesto estimado para llevar a cabo el proyecto en su totalidad es de unos 4087,14 €.