Universidad del País Vasco
Euskal Herriko Unibertsitatea

Facultad de Informática

Grado de Ingeniería Informática

▪ Proyecto Fin de Grado ▪

Ingeniería de Computadores

Integración de dispositivos IoT en una red comunitaria

Autor

Alejandro Reyes Díez

Director

Julián Alberto Lafuente Rojo

Promotora

Maider Likona Santamarina

Septiembre 2017

informatika fakultatea
facultad de informática

Agradecimientos

Agradezco a mi tutor Alberto Lafuente por ofrecerme este proyecto y con ello brindarme la valiosa oportunidad de conocer a toda la gente que me ha acompañado en este trayecto.

Agradezco a los miembros de GISA por contar conmigo para su idea y por su respaldo incondicional a lo largo de este camino.

Agradezco a toda la gente que he conocido gracias a esta aventura, personas relacionadas con el mundo del IoT que transmiten una pasión contagiosa por el desarrollo de las tecnologías que lo conforman.

Agradezco a los compañeros que he conocido en el transcurso de estos años. Y especialmente se lo agradezco a mis amigos Edorta, Urtzi, Edgar, Asier, Gonzalo, Igor, Aitor, Beñat, Aritz y Eneko con los que he tenido la suerte de vivir esta etapa de mi vida.

Y, por último, se lo agradezco a mi familia por su apoyo incondicional, ya que sin ellos nada de esto hubiese sido posible.

Muchas gracias a todos.

Resumen

Conforme se va expandiendo el Internet de las Cosas (IoT) la demanda de dispositivos interconectados aumenta. Como respuesta, se han comenzado a desarrollar estándares que ofrecen alto rango de cobertura y bajo consumo energético, características que serán indispensables en un nuevo escenario en el que se prevé que multitud de dispositivos precisen de gran conectividad.

Uno de los estándares que más despunta es LPWAN, que optimiza el alcance, la vida de la batería y el coste de producción a cambio de una menor frecuencia de transmisión de datos, haciéndolo así un estándar idóneo para el IoT.

Las características de estas nuevas tecnologías ofrecen la posibilidad inédita de dar conectividad a sensores en lugares anteriormente inalcanzables. No obstante, esto no viene sin nuevos retos. Una de las tecnologías LPWAN más destacadas, LoRaWAN, es un protocolo no-IP, por lo tanto, requiere de un gateway para conectarse a una red IP como Internet.

Para alcanzar lugares remotos este gateway, en muchos casos, tiene que colocarse en zonas en las que conseguir acceso a Internet no es un problema trivial. En este punto, las redes comunitarias, como Guifi.net, pueden proveer una conectividad que las distribuidoras de Internet convencionales no son capaces de proporcionar.

En este estudio se van a analizar las posibilidades de conectividad que tienen las redes de sensores valiéndose de las redes comunitarias para poder conectarse a Internet, para ello se va a desarrollar una prueba de concepto (PoC) de una red LoRaWAN que será conectada a Internet mediante Guifi.net.

Palabras clave: IoT, LPWAN, LoRa, LoRaWAN, Red de sensores, Redes comunitarias, Guifi.net

Abstract

As the Internet of Things (IoT) expands, the demand for interconnected devices increases. In response, standards that offer a long range coverage and low energy consumption have been developed. This features will be essential in a new scenario in which it is expected that many devices will be required high connectivity.

One of the most prominent standards is LPWAN, which optimizes reach, battery life and manufacturing cost in exchange for a lower data rate, making it a very suitable standard for IoT.

The capabilities of these new technologies offer the unprecedented possibility of providing connectivity to sensors in previously unreachable places. However, this also raises new challenges. One of the most outstanding LPWAN technologies, LoRaWAN, is a non-IP based protocol, therefore, it requires a gateway to connect to an IP network such as the Internet.

In order to reach remote locations in many cases this gateway has to be positioned in areas where obtaining Internet access is not a trivial problem. It is at this point, when community networks, such as Guifi.net, can provide the connectivity that regular Internet providers are not able to supply.

This thesis will analyze the possibilities of connectivity for sensor networks using community networks to reach Internet connection, for this purpose a proof of concept (PoC) of a LoRaWAN network will be developed and connected to the Internet through Guifi.net.

Key words: IoT, LPWAN, LoRa, LoRaWAN, Sensor networks, Community networks, Guifi.net

Laburpena

Gauzen Interneta (IoT) hedatzen den heinean, gailu inter konektatuen eskaria handituz doa. Honi erantzun emateko, estaldura maila handia eta energia kontsumo baxua eskaintzen dituzten estandarrak garatzen hasi dira. Ezaugarri hauek, etorkizunean ezinbestekoak izango dira, gailu asko elkar-konektatuta egon beharko baitira.

Gehien gailentzen den estandarretako bat LPWAN da. Estandar honek irismena, bateriaren bizitza eta ekoizpen kostua optimizatzen ditu datu igorpen frekuentzia txikiago baten truke, modu honetan, estandar aproposa bihurtuz IoT-rako.

Teknologia berri hauen ezaugarriek, ordurarte ezinezkoa zen tokietan, sentsoreei konektibitatea eskaintzea ahalbidetzen du. Hala eta guztiz ere, honek erronka berriak dakartza. LPWAN teknologietako batek, LoRaWAN, ez du IP protokoloa erabiltzen, horregatik, gateway baten beharra du Internet bezalako IP sare batera konektatzeko.

Urrutiko leku hauetara heltzeko, askotan gateway hau Interneterako sarbidea lortzea erraza ez den lekuetan kokatu behar da. Horretarako, komunitate sareek, Guifi.net alegia, ohiko Internet zerbitzu hornitzaileek eskaini ezin duten konektibitatea eskain dezakete.

Lan honetan sentsore sareek dituzten konektibitate aukerak aztertuko dira, komunitate sareetaz baliatuz Internetera konektatzeko. Honetarako, kontzeptu froga (PoC) bat garatuko da, non LoRaWAN sare bat Guifi.net-en bitartez Internetera konektatuko den.

Hitz gakoak: IoT, LPWAN, LoRa, LoRaWAN, sentsore sareak, komunitate sareak, Guifi.net

Índice

Lista de Figuras

Lista de Tablas

1

1. Introducción

Este primer capítulo sirve para introducir al lector a la obra. En él se contextualiza el contenido del trabajo y se ofrece una visión holística del problema.

La importancia del acceso a las comunicaciones cada vez se ve más reconocido. Tanto es así que en el artículo 19 de la Declaración Universal de los Derechos Humanos de 1948 se declara el derecho a la comunicación (Organización de las Naciones Unidas, 1948). Normalmente, este derecho ha sido directamente asociado a los medios tradicionales de difusión de información, ya sean, televisión, prensa, radio…

Pero como muestra el estudio realizado en EEUU por el Pew Research Center (Pew Research Center, 2016), se puede observar como hay una tendencia clara a decantarse por Internet como medio para recibir la información. En este estudio, se puede advertir cómo esta tendencia se acrecienta en los jóvenes, y muestra cómo las personas entre 18-49 años de edad eligen Internet como su medio de divulgación preferido, por encima de opciones antiguamente mayoritarias como la televisión.

En este escenario, en 2016 el Consejo de Derechos Humanos de las Naciones Unidas publica una resolución no vinculante (Human Rights Council, 2016) que declara que el acceso a Internet es un derecho humano y lo hace apoyándose en que Internet tiene un gran potencial para acelerar el progreso humano.

Estos hechos denotan la magnitud que está alcanzando Internet, pero a pesar de que las leyes y resoluciones avanzan, Internet lo hace más rápido. En los últimos años, ha surgido un nuevo movimiento marcado por la predominancia de las conexiones entre dispositivos (M2M), conocido como *Internet of things* (IoT) que literalmente significa, "Internet de las cosas".

El IoT se ha transformado en uno de los términos más populares en la industria tecnológica, y tanto es así que se ha convertido en un elemento transversal de nuestra sociedad. En este nuevo paradigma se prevé que haya multitud de dispositivos interconectados y que sean capaces de compartir información a tiempo real, esto presenta opciones que anteriormente no eran posibles como la creación de ciudades inteligentes (*Smart Cities*), casas inteligentes o la industria 4.0 que consiste en valerse del IoT para optimizar procesos de producción.

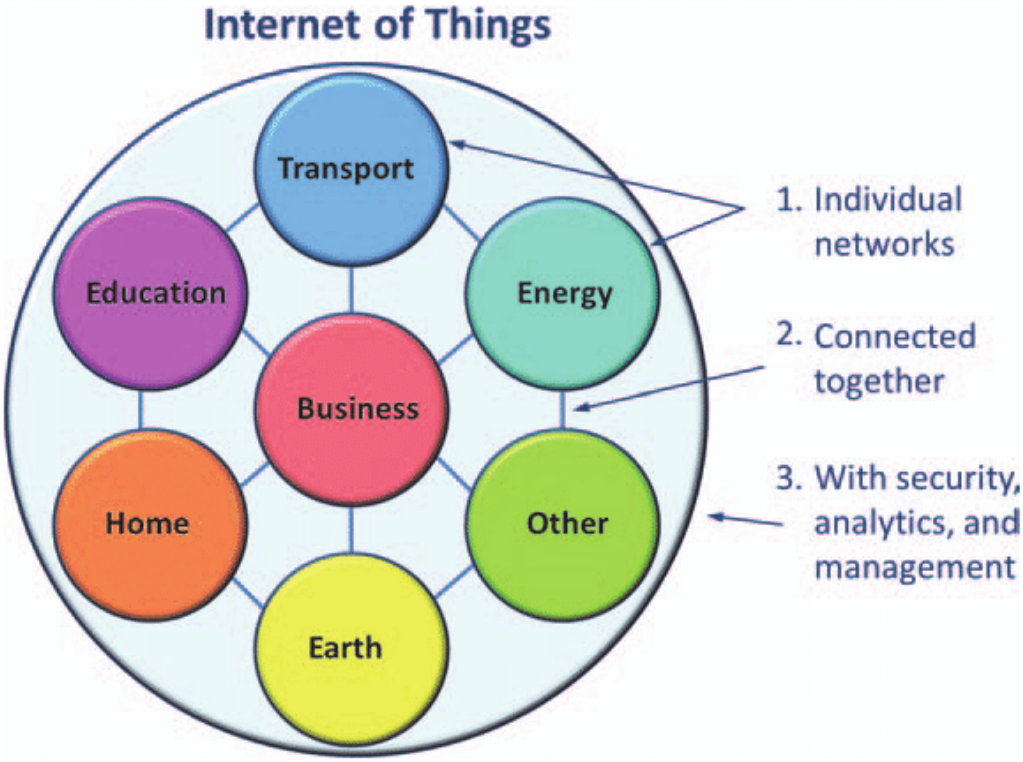


Figura: 1.1. Casos de uso IoT.

Fuente: (Cisco, 2011)

Gartner, en un estudio publicado en 2017 (Gartner, 2017), pronostica que habrá 8,4 billones (americanos) de dispositivos conectados a finales de 2017 y que para 2020 llegará a los 20.4 billones. A su vez, la inversión en estos dispositivos alcanzará la cifra de 2 trillones de dólares.

Además, existe una tendencia actual en el ámbito empresarial que consiste en externalizar los servicios. Según una encuesta difundida por Deloitte (Deloitte Consulting LLP, 2016), la externalización de servicios se va a acrecentar en el futuro, ya que, según sus encuestados el *outsourcing* es muy beneficioso a la hora de ahorrar costes (Un 59% lo eligieron como una de las razones principales) o concentrarse en las labores principales de la empresa (Un 57% lo eligieron como una de las razones principales).

Estos datos unidos crean un nicho de mercado poco explotado y de alta rentabilidad. Este sería crear una infraestructura de comunicaciones IoT para luego ofrecerla como servicio a los clientes (IaaS). Algunas empresas reconocidas se han apuntado ya a esta oportunidad (Intel, Cisco (Casey, 2015), Sigfox..), lo cual demuestra el potencial de este joven mercado.

De esta manera, se consiguen dos objetivos, por una parte, el beneficio económico que otorga rentabilidad al proyecto y, por otra parte, unido a lo comentado sobre el derecho a la comunicación se podrá crear una infraestructura que facilitará a pequeñas empresas y particulares acceder al mundo IoT.

La involucración de una empresa en un proyecto de estas características es muy loable, ya que, se alinea con el noveno objetivo del PNUD para el desarrollo sostenible:

"La inversión en infraestructura y la innovación son motores fundamentales del crecimiento y el desarrollo económico. […] Más de 4.000 millones de personas aún no tienen acceso a Internet […] Reducir esta brecha digital es crucial para garantizar el acceso igualitario a la información y el conocimiento, y promover la innovación y el emprendimiento." (PNUD, 2015)

2

2. Objetivos y Alcance

La función de este capítulo es analizar los motivos que han llevado a la realización de este TFG. Junto a esto, se definirán los objetivos concretos del proyecto y el alcance que los limita. Seguidamente, se mostrará el EDT con los paquetes de trabajo principales que se han seguido durante el transcurso del trabajo y, por último, se le hará al lector un recorrido rápido por lo que podrá encontrar en los sucesivos apartados.

2.1. Motivación

Este proyecto se desarrolla bajo la tutela de la empresa Gipuzkoako Software Askea Elkartea (GISA). Es una empresa que tiene su campo de acción en la zona de Guipúzcoa y tiene acceso a zonas óptimas para la colocación de gateways.

Esto se debe a tres características principales:

- **Visibilidad:** Los lugares en su mayoría están localizados en montañas o sitios altos con buena visibilidad de la zona.
- **Acceso a suministro eléctrico:** En estos lugares ya hay acceso al suministro eléctrico, lo cual evita la necesidad de llevar baterías portables u otro tipo de soluciones.
- **Conectividad a redes comunitarias:** Estos lugares tienen acceso a antenas de la red comunitaria Guifi.net[1] que abarca parte de la geografía guipuzcoana y desde la cual se puede acceder a Internet.

Se puede comprobar la utilidad de estas zonas observando otras iniciativas locales, como la descrita en el artículo "*Wireless Sensor Networks for Bird Traking*" (Burgos, Gamecho, Gardeazabal, Gómez-Calzado, & Lafuente, 2016), en la que se describe un ejemplo de aplicación de red de sensores para el seguimiento de pájaros.

En este artículo se plantea el siguiente problema:

"A pesar de que los datos pueden quedarse en la estación base […] conectar la estación base a Internet hace posible convertir el sistema de seguimiento en un servicio web. Aun así, […], específicamente en nuestra colonia los métodos convencionales de conexión a Internet no están disponibles. Esto se debe a que la colonia se encuentra lejos de cualquier área habitada y fuera del alcance de las redes móviles. […] Actualmente se está explorando la posibilidad de acceso a Guifi.net […]"[2] (Burgos, Gamecho, Gardeazabal, Gómez-Calzado, & Lafuente, 2016)

Lo que aquí se describe es un problema que es común en Guipúzcoa debido a su geografía característica, ya que su orografía accidentada hace que sea más difícil el acceso a algunos de los métodos convencionales de conexión a Internet.

La empresa GISA puede ofrecer solución a empresas o particulares en esta misma situación. Esto es gracias a las características de las zonas para colocación de gateways que se han descrito con anterioridad, es decir, buen lugar de posicionamiento para los gateways y conexión a Internet mediante la red comunitaria Guifi.net.

[1] www.guifi.net
[2] Texto traducido del inglés.

Asimismo, se debería aprovechar el hecho de que GISA es capaz de ofrecer conexión a Internet a las redes de sensores para a su vez desarrollar un servicio web que permita la visualización de datos en tiempo real.

Además, hay que tener en cuenta que la red de sensores para el seguimiento de pájaros es un ejemplo interesante, pero es solo un ejemplo de muchos, multitud de sectores han advertido las posibilidades que ofrecen las redes de sensores. De esta forma han nacido iniciativas como las *Smart cities* o la industria 4.0. Iniciativas que se valen de las redes de sensores, para ofrecer más comodidades a los ciudadanos o que sirven para aumentar la productividad de una empresa.

De estos datos se deduce que la creación de nuevas redes de sensores va a cobrar mucho interés en los próximos años. Y para sacar todo el partido posible a estas redes la integración de la información producida en Internet es un aspecto clave.

2.2. Objetivos

2.2.1. Objetivo general

Desarrollar y documentar una prueba de concepto de una red de sensores conectada a Internet mediante redes comunitarias. Además, investigar opciones para crear un servicio web desde el que les sea posible a los clientes acceder a sus datos.

Para este desarrollo, se realizará un análisis detallado de las tecnologías a utilizar, teniendo en cuenta parámetros como: la optimización de la conectividad, el alcance y la rentabilidad del proyecto.

2.2.2. Objetivos específicos

- Analizar y evaluar las diferentes tecnologías disponibles para comunicar los sensores.
- Explorar y evaluar las diferentes opciones para comunicar la red de sensores con Internet.
- Explorar y evaluar las diferentes opciones existentes para dejar la información persistente y accesible mediante un servicio web.
- Preparar el equipamiento o desarrollar herramientas para la realización de pruebas.
- Obtener medidas mediante experimentación con las tecnologías seleccionadas.
- Analizar posibilidades de implantación teniendo en cuenta la zona de implantación y las características de las tecnologías escogidas.
- Plantear usos posibles que se le podrían dar a la infraestructura de red.
- Desarrollar una valoración económica del proyecto y analizar las distintas formas de monetización del mismo.
- Explorar opciones básicas de seguridad de la red.

- Generar recomendaciones basadas en los conocimientos adquiridos para la fase de puesta en producción de la infraestructura de red.

2.3. Metodología

Como se ha explicado anteriormente este proyecto es una PoC o prueba de concepto, es decir, este trabajo forma parte de una metodología en la que se ha decidido hacer un prototipo para comprobar la viabilidad de una idea.

A la hora de desarrollar esta prueba de concepto también se ha seguido una metodología específica:

Paso 1: Definir el problema

Se plantea el problema al que se enfrenta este trabajo marcando unos objetivos claros y una planificación que defina los diferentes paquetes de trabajo y el tiempo que se les va a dedicar.

Paso 2: Estudio bibliográfico

Se realiza un estudio de las tecnologías involucradas en el proyecto. La finalidad de este estudio es tener una base suficiente para poder tomar decisiones en la fase de diseño.

Paso 3: Diseño de la solución

Se diseña una solución que cumpla con los objetivos principales. Para ello se analizan diferentes tecnologías y se observa de qué manera cumplen los objetivos.

Paso 4: Implementación de la solución

Se implementa y configura la solución propuesta.

Paso 5: Diseñar escenarios de prueba

Se diseñan escenarios de prueba que ayuden a comprobar que la solución es funcional y que cumple con los objetivos.

Paso 6: Analizar los resultados y elaborar conclusiones

Después de hacer las pruebas se hace un análisis de los resultados que han devuelto las pruebas y basándose en los objetivos que se habían planteado al principio se valora el éxito del prototipo.

En caso de ser exitoso se proponen trabajos futuros para convertir esa PoC en una implementación real y en caso de considerar que no es válido se concluye que es mejor no seguir esa vía para el desarrollo final y se analizan los errores cometidos y consejos para nuevos prototipos.

2.4. Alcance

2.4.1. Exclusiones

Se excluye de este proyecto todo lo que no ha sido mencionado en los objetivos, entre lo que destaca:

- La implementación de la red en un escenario de producción.
- Al no ser un prototipo público se excluye analizar los requerimientos para cumplimentar las leyes de protección de datos.
- Pruebas completas de esfuerzo para testear el backend.
- Crear un backend escalable y que ofrezca alta disponibilidad.

2.4.2. Esquema de Descomposición del Trabajo (EDT)

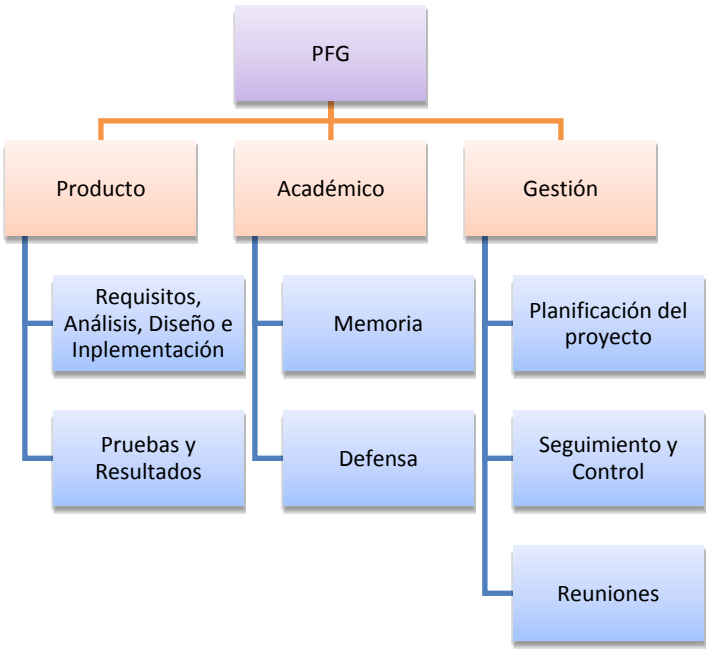


Figura: 2.1. EDT del proyecto.

Fuente: Diseñado por el autor.

2.5. Estructura de la memoria

El principio de cada uno de los capítulos contiene una descripción detallada de lo que se puede encontrar en ellos, pero, como idea general la estructura de la memoria es la siguiente:

- En el capítulo 3, Estado del arte, contiene un análisis de la historia de las tecnologías que están involucradas en esta obra.
- En el capítulo 4, Análisis del problema, se hace un análisis y comparación de las soluciones existentes y se selecciona la que más se ajusta a las necesidades del proyecto.
- En el capítulo 5, Diseño de la solución, se describe concretamente el software y el hardware elegido y se ofrece una explicación completa de la arquitectura.
- En el capítulo 6, Implantación y Pruebas, se hace un análisis del entorno y en base a este se describe la toma de decisiones para la implantación del proyecto en un escenario real. Además, se detallan los procesos para hacer las pruebas.
- En el capítulo 7, Valoración económica, se hace un análisis de costes y se propone un modelo de explotación de la solución.
- En el capítulo 8, Gestión, se hace un análisis de las áreas más relevantes de la gestión del proyecto.
- En el capítulo 9, Conclusiones y Trabajos futuros, junto con la revisión de las conclusiones extraídas por el autor durante el proyecto se plantean líneas que el autor considera interesantes para continuar.
- Por último, en los anexos se ofrecen las configuraciones tanto de gateways, sensores y backend, como de, la conexión a Guifi.net.

3

3. Estado del arte

En este capítulo se realiza un análisis del estado del arte del tema tratado en esta obra. Se comienza con una introducción al IoT y las diferentes formas que existen para conseguir que se conecte a la red, luego, se profundizará en las tecnologías centrales que trata este documento, LoRa y LoRaWAN. Finalmente, se comentará la importancia de las redes comunitarias en la conectividad global, en concreto el ejemplo más cercano y exitoso: guifi.net.

3.1. Conectividad IoT

Junto con el auge del IoT se empiezan a desarrollar tecnologías de comunicación preparadas para este nuevo escenario. Dentro de estas nuevas tecnologías existen tres grupos destacados:

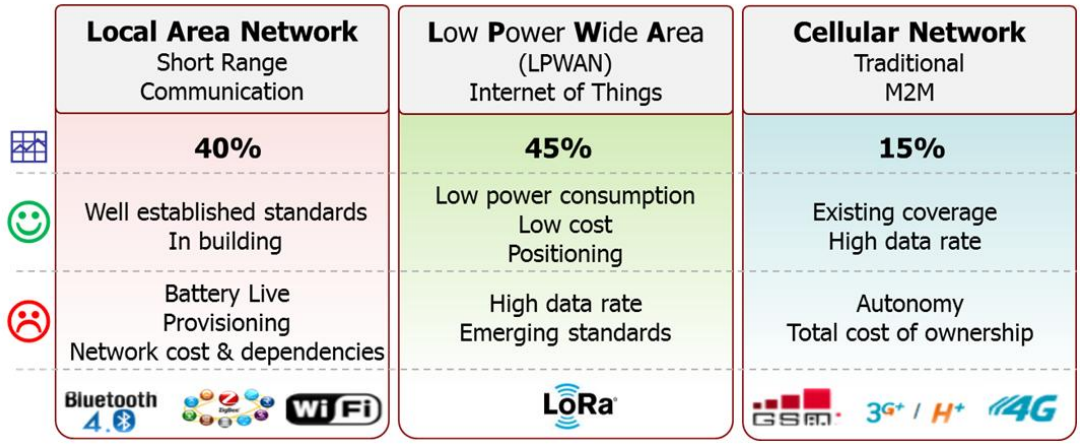


Figura: 3.1. Soluciones para conectividad IoT.

Fuente: (Lora Alliance, 2015).

En la **Figura: 3.1** la primera de las soluciones (Local Area Network), es demasiado genérica y abarca demasiadas tecnologías, algunas nada preparadas para el IoT, como por ejemplo el WiFi. Así que se hablará de una variante específica de las LAN, las Low-Rate WPAN.

3.1.1. Low-Rate Wireless Personal Area Networks

Las tecnologías LR-WPAN crean pequeñas redes, normalmente conectan dispositivos en un espacio pequeño como, por ejemplo, un domicilio.

El estándar IEEE 802.15.4 define el nivel físico y la capa MAC de estos protocolos. El énfasis de esta tecnología es ser útil para crear redes de tamaño reducido y de velocidad de envío de datos pequeña, pero consiguiendo disminuir en gran cantidad el consumo de los dispositivos.

A nivel de topología normalmente se configuran en redes malladas, aproximadamente el área de comunicación para cada uno de los dispositivos es de 10 metros y tienen una tasa de transferencia de 250 kbit/s dependiendo la banda que se use.

Estándares conocidos que se incluirían en esta categoría son Bluethoot Low Energy (BLE) o Z-Wave.

3.1.2. Cellular IoT

Otra de las soluciones que se proponen para la conectividad IoT son las redes móviles. Pero las redes móviles tradicionales como 4G o LTE tenían un consumo muy alto y no eran adecuadas para el nuevo escenario, en el que la transmisión es de muchos dispositivos, pero en cantidades pequeñas.

Pero las empresas que usaban estas tecnologías tenían un gran terreno recorrido. Tenían en su control la infraestructura para ofrecer acceso a Internet, por lo tanto, sólo necesitaban crear nuevos estándares personalizados para el IoT.

Los ejemplos más importantes son LTE-M, NB-IoT y EC-GSM.

- LTE-M: Una de las primeras soluciones que se propuso, la primera en completar el objetivo de reducir drásticamente el consumo y el coste. Esta solución es totalmente compatible con la red LTE.
- NB-IoT: Este estándar se diferencia del anterior porque usa una tecnología diferente que no le permite operar en la banda LTE, por lo tanto, el coste es algo mayor a la hora de desplegar infraestructura. A pesar de esto se postula como la solución más barata ya que a diferencia de LTE-M no tiene la necesidad de usar gateways, por lo que la información se transmite directamente hacia los servidores.
- EC-GSM: Esta solución es la mejora de la red GSM para adaptarla al IoT, se sirve de la red GSM e incluye mejoras a la cobertura y al rango de los dispositivos.

3.1.3. Low Power Wide Area Networks

Las tecnologías LPWAN nacen como alternativa a las anteriores, lo hacen ofreciendo un consumo muy reducido y un rango de alcance muy alto a cambio de una tasa de datos bastante menor (Figura: 3.2).

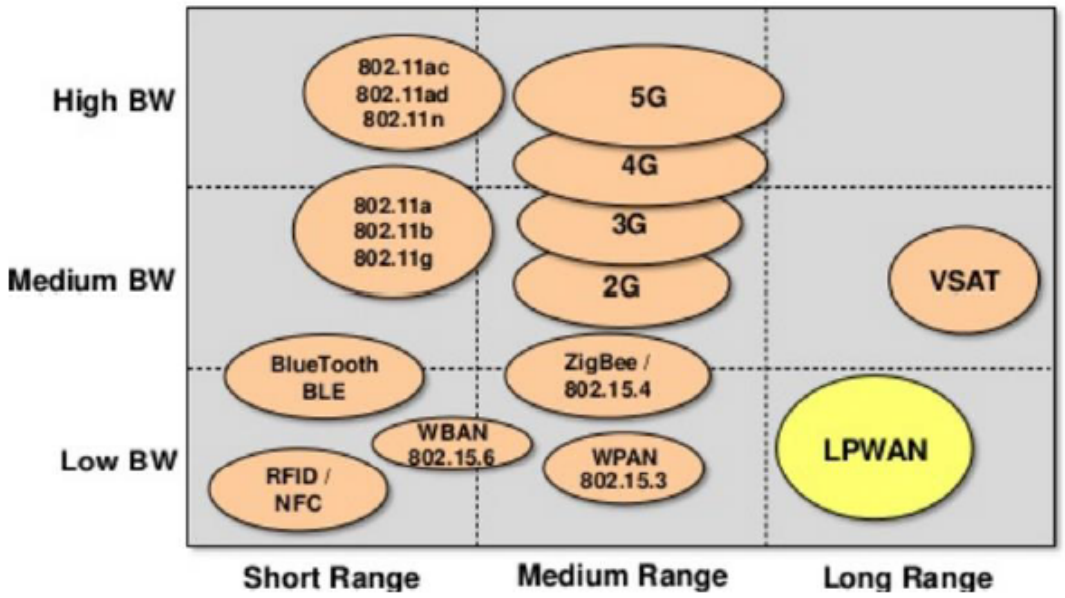


Figura: 3.2. Relación entre el ancho de banda y alcance en tecnologías inalámbricas.

Autor: (Egli, 2015).

Las tecnologías LPWAN son un grupo amplio, pero ofrecen un rango de alcance de 5 – 40km, una duración de batería de hasta 10 años y un coste por dispositivo muy reducido.

Dentro de estas tecnologías existen dos que destacan entre las demás, LoRaWAN y Sigfox. En el ámbito tecnológico se diferencian en la forma en la que hacen la modulación, mientras LoRaWAN usa LoRa (CSS), Sigfox utiliza una técnica llamada UNB (Banda Ultra Estrecha).

Pero la mayor diferencia está en el modelo de negocio, esto se entiende si se mira la historia de ambas compañías, mientras LoRa está producido por Semtech Corporation una empresa dedicada a crear hardware, Sigfox es una empresa de telecomunicaciones.

Por lo tanto, Semtech Corporation tiene como objetivo potenciar la venta de chips, esto lo consiguen creando un protocolo libre por encima de LoRa pero manteniendo LoRa como privado. En cambio, Sigfox tiene como objetivo convertirse en el operador del IoT, para conseguir esto permiten que se usen chips de una gama muy amplia de fabricantes, pero ellos se encargan de dar el servicio de red.

Además de estas dos tecnologías existen multitud más, quedan resumidos los atributos principales de algunas de ellas en la siguiente tabla:

	LoRaWAN	Sigfox	Weightless - N	Weightless -P	nWave
Alcance	2-5km (ciudad); 15km (rural)	10km (ciudad); 30-50km (rural)	5km	2km	10km
Banda de frecuencia	433/868/780/915MHz ISM	Ultra Narrow Band (UNB)	Sub-GHZ ISM	Sub-GHZ ISM	Sub-GHZ ISM
Tasa de envío de datos	300 bps a 50kbps	100 bps, máximo 140 mensajes al día	100 bps	200 bps a 100 kbps	100 bps
Tasa de recepción de datos	300 bps a 50kbps	4 mensajes de 8 bytes al día	No	200 bps a 100 kbps	No
Estándar	LoRaWAN	No	Weightless	Weightless	Weightless

Tabla 3.1. Comparativa entre tecnologías LPWAN.

Fuente: (CNXSoft, 2015).

3.2. LoRa

LoRa es una técnica de modulación usada para crear enlaces de largo alcance, básicamente modulación CSS (Chirp Spread Spectrum) usada para conseguir distintas velocidades de transferencia de datos sobre diferentes canales.

Antiguamente, una de las técnicas de modulación que se usaba era FSK, porque era muy eficiente y conseguía bajo consumo. LoRa es similar a FSK, pero aumenta el rango que puede alcanzar el enlace.

LoRa consigue esto sirviéndose de toda la anchura de banda de los canales para transmitir la señal, de esta manera, consigue una señal mucho más resistente al ruido. El funcionamiento exacto de LoRa se desconoce ya que es tecnología propietaria de Semtech, aun así, se pueden encontrar algunos documentos oficiales con una explicación general (Semtech Corporation, 2015).

Además, LoRa es una implementación para la capa física que no requiere de ninguna implementación concreta para las capas superiores.

3.3. LoRaWAN

LoRaWAN es un protocolo implementado sobre LoRa. Mientras LoRa se encarga de la capa física y habilita conexiones de largo alcance, LoRaWAN se encarga de definir el protocolo de comunicación y la arquitectura de los sistemas.

Como hemos visto anteriormente LoRaWAN es una tecnología LPWAN, en la que LoRa es el encargado de ofrecer la conexión en largas distancias y la implementación de las capas superiores(LoRaWAN) es la encargada de ofrecer el bajo consumo.

A continuación (**Figura: 3.3**), se muestra la pila de funcionamiento LoRaWAN en la que se puede ver la diferencia entre LoRa (capa física) y LoRaWAN.

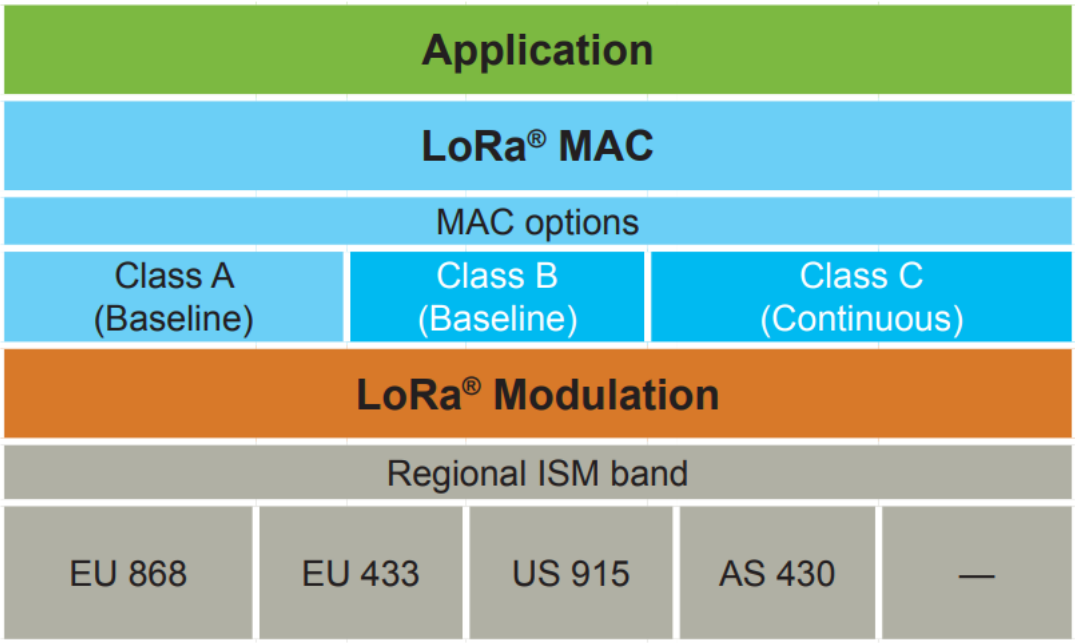


Figura: 3.3. Pila de funcionamiento LoRaWAN.

Fuente: (Lora Alliance, 2015).

Este protocolo ha sido desarrollado y hecho público por la LoRa Alliance, una organización sin ánimo de lucro dedicada a promover las tecnologías LPWAN.

La descripción del protocolo se puede encontrar en el siguiente documento (Sornin , Luis, Eirich, Kramp, & Hersent, 2015).

3.3.1. Topología y componentes de una red LoRaWAN

La topología típicamente usada por una red LoRaWAN es la de estrella (**Figura: 3.5**), o estrella de estrellas, a diferencia de las topologías de tipo malla (**Figura: 3.5**) que usan otras soluciones para el IoT, la topología de estrella ofrece muy buenas latencias y necesidad de mucha menos infraestructura.

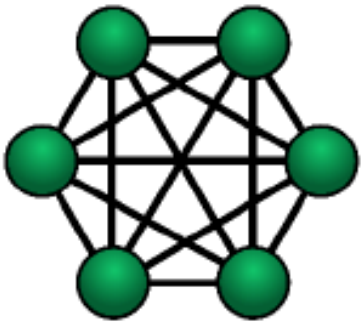


Figura: 3.4: Topología de malla.

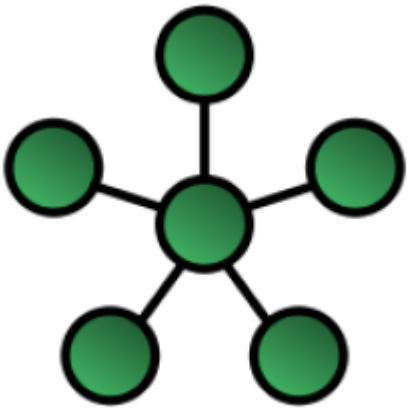


Figura: 3.5. Topología de estrella.

En estas redes existen los siguientes componentes principales:

- **Nodos**: Estos son uno de los dispositivos finales, son los encargados de transmitir mensajes LoRaWAN usando la técnica de modulación LoRa.
- **Gateway**: El gateway o concentrador es el encargado de recibir estos paquetes LoRaWAN y mandarlos al backend mediante una conexión IP.
- **Network Server**: El servidor de red es el encargado de borrar los paquetes que llegan duplicados desde los gateways, reconocer si los paquetes que le llegan pertenecen a su red y en caso de ser así enviarlo al servidor de aplicaciones.

- Application Server: El servidor de aplicaciones es el encargado de gestionar los nodos y los usuarios, para facilitar la gestión de las aplicaciones, también se encarga de descifrar los datos.

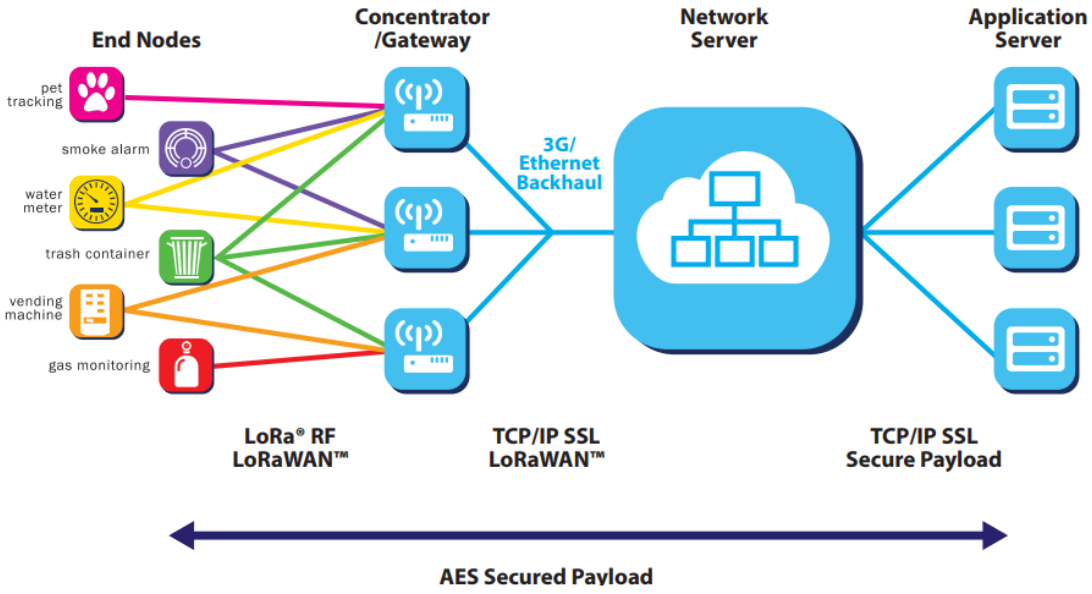


Figura: 3.6. Componentes principales de una red LoRaWAN.

Fuente: (Lora Alliance, 2015).

3.3.2. Métodos de activación

Los dispositivos LoRaWAN tienen un identificador único (*DevEUI*) que se les asigna por el creador del chip, pero las comunicaciones se hacen mediante otro identificador, *DevAddr*, este identificador se asigna al unirse a una red.

Existen dos métodos para que un nodo se una a una red LoRaWAN:

- Activation by Personalization (ABP): Se escriben directamente las claves de sesión, tanto en el nodo, como en el servidor y se da a conocer la dirección del nodo (el *DevAddr,* que se ha escrito directamente en el nodo) al servidor, de esta manera la transmisión de paquetes puede empezar directamente.
- Over-the-Air Activation (OTAA): En este procedimiento el nodo y el servidor negocian las claves de sesión de forma segura y se le asigna un *DevAddr* al nodo de forma dinámica.

3.3.3. Canales y regulación del espectro de radiofrecuencia

LoRa opera en un espacio de radiofrecuencia sin licencia, el ISM (Industria, ciencia y medicina), esto quiere decir que cualquiera puede utilizar esta banda sin pagar licencia.

Estas frecuencias dependiendo de la situación geográfica varían:

Región	Banda de frecuencia
Europa	868-870
EEUU	902-928
China	779-897

Tabla 3.2. Bandas de frecuencia según la región.

Fuente: Creada por el autor.

Por lo tanto, en Europa se usa la banda de 868MHz a 870MHz, este espectro está regulado por la sección 7.2.3 del estándar ETSI EN300.220 (The Things Network, 2017).

En este estándar se definen 5 sub-bandas y se especifica cuál tiene que ser el uso de ciclos:

- **g** (863.0 – 868.0 MHz): 1%.
- **g1** (868.0 – 868.6 MHz): 1%.
- **g2** (868.7 – 869.2 MHz): 0.1%.
- **g3** (869.4 – 869.65 MHz): 10%.
- **g4** (869.7 – 870.0 MHz): 1%.

Un ciclo de uso indica el tiempo que un dispositivo está ocupado, en la lista anterior se puede ver cuánto está permitido por cada sub-banda.

Un dispositivo LoRa cuando envía un mensaje utiliza distintos canales (los va alternando de forma aleatoria), por lo tanto, si se tiene un solo canal y se transmite por 2 unidades de tiempo cada 10 unidades de tiempo el dispositivo tendría un 20% de uso de ciclo (**Figura: 3.7**).

Figura: 3.7. Dispositivo enviando en un solo canal.

Fuente: (The Things Network, 2017)

Si transmite por 2 unidades de tiempo cada 10 pero en tres canales el uso de ciclo sería del 60% (Figura: 3.8).

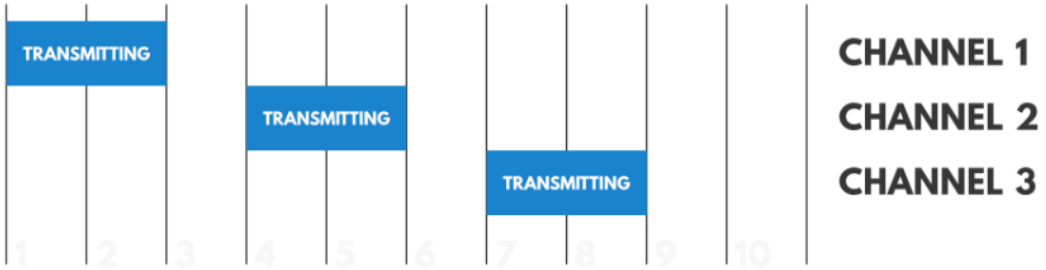


Figura: 3.8. Dispositivo enviando en tres canales.

Fuente: (The Things Network, 2017)

En el plan de frecuencia que existe en Europa los canales están en las sub-bandas, por ejemplo, podría ser que dos canales operasen en la sub-banda g y un canal en la sub-banda g1. Esto es algo a tener en cuenta, por ejemplo, en la Figura: 3.9, se puede ver como cada uno de los canales independientemente tiene un uso de ciclo del 20% pero esto no es relevante en cuanto a la regulación del espectro de radiofrecuencia, lo que hay que mirar es el uso de ciclo en cada una de las sub-bandas. Por lo tanto, se aprecia que en la Figura: 3.9 la banda 1 tiene un 20% y la banda 2 un 40%, mientras el dispositivo tiene un 60%.

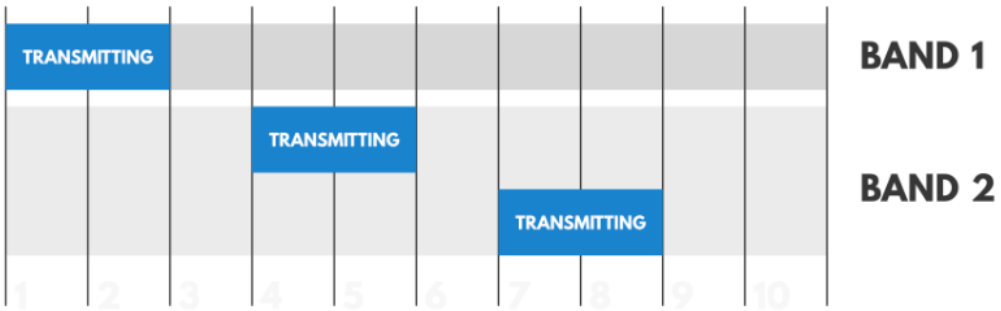


Figura: 3.9. Dispositivo enviando en tres canales dentro de dos sub-bandas.

Fuente: (The Things Network, 2017)

En este caso lo que hay que comprobar en qué sub-banda está cada canal y si los canales de una sub-banda superan el uso de ciclos permitido.

Hacer estos cálculos no es sencillo, pero el usuario final no tiene que preocuparse de esto, ya que todos los dispositivos que operen en esta banda están obligados a cumplir la normativa

actual (sección 7.2.3 del estándar ETSI EN300.220) y en caso de superar los niveles permitidos el dispositivo lanza un error y no permite seguir enviando paquetes.

Por lo tanto, esto sólo interesaría a desarrolladores, para ellos existen multitud de herramientas[3] que facilitan los cálculos de los ciclos de uso.

3.3.4. Clases LoRaWAN

Para los dispositivos finales en LoRaWAN existen 3 clases diferentes A, B y C. Todos los dispositivos tienen las características de la clase A por lo menos, las características específicas de cada clase son las siguientes (The Things Network, 2017):

- **Clase A:** Los dispositivos de clase A permiten una comunicación bidireccional. En esta comunicación se pueden enviar mensajes en cualquier momento. Una vez enviado un paquete se abren dos ventanas de recepción en un tiempo especificado. Si el servidor no responde en esas dos ventanas, se tendrá que repetir el envío por parte del sensor y volver a repetir el proceso.

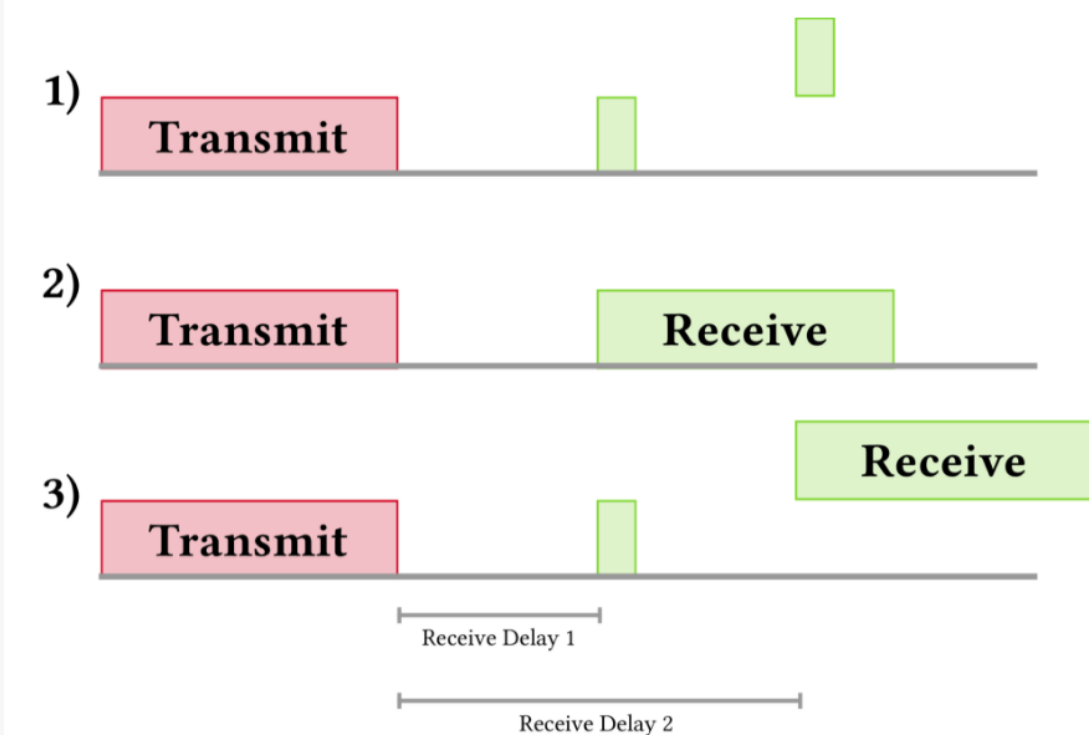


Figura: 3.10. Funcionamiento de los dispositivos de clase A.

Fuente: (The Things Network, 2017).

[3] http://www.semtech.com/apps/filedown/down.php?file=SX1272LoRaCalculatorSetup1%271.zip

En la imagen anterior (Figura: 3.10) se puede apreciar el funcionamiento. En el ejmplo número 1 se ve en color verde las dos ventanas de recepción y como después de transmitir no se recibe nada, en este caso se debería repetir la transmisión.

En cambio, en los ejemplos 2 y 3 (Figura: 3.10) se puede observar como sí se reciben los mensajes en las dos ventanas de transmisión.

- Clase B: Estos dispositivos añaden a las características de los dispositivos de clase A ventanas de recepción programadas. Esto se consigue mediante *beacons* que envía el servidor, usando estos *beacons* el gateway y los sensores pueden negociar tiempos de abertura de ventanas de recepción.
- Clase C: Los dispositivos de la clase C, además de las características de la clase A, tienen las ventanas de recepción abiertas la mayor parte del tiempo, estas solo se cierran cuando se realiza una transmisión.

3.3.5. Seguridad en LoRaWAN

Para la seguridad LoRaWan utiliza 3 claves de 128 bits:

- Network Session Key: Esta clave se usa en la interacción del nodo con la red.
- Application Session Key: Esta clave se usa en la interacción del nodo con el servidor de aplicaciones.
- Application key: Esta clave se usa cuando se activa un dispositivo usando el método OTAA, sirve para crear las claves de sesión necesarias para la activación.

En un escenario normal un nodo se conectaría a la red mediante OTAA y se le asignarían las dos claves de sesión, la clave de red se compartiría por la red y se usaría para validar la integridad de los mensajes (validación MIC). A su vez la clave de sesión de aplicación se mantendría como privada y se usaría para cifrar y descifrar los paquetes

3.4. Redes comunitarias

Las redes comunitarias son un tipo de redes que surgen como alternativa a las redes convencionales, en las que un proveedor cobra por los servicios ofrecidos y establece unas normas.

En contraposición las redes comunitarias son iniciativas de redes que tienen como objetivo proveer a una comunidad de una red para los fines que ellos decidan. Algunos ejemplos pueden ser, proporcionar una red fiable a todos sus miembros, crear una alternativa más democrática de red en la que se acepten a más usuarios (gente con problemas socioeconómicos), promocionar la comunidad que la ha creado…

La anterior es una definición muy abierta de las redes comunitarias, esto se debe a que existen muchas iniciativas a lo largo del mundo que gestionan su red comunitaria de formas diferentes.

3.4.1. Ejemplos de redes comunitarias en el mundo

Algunos ejemplos de redes comunitarias son las siguientes:

- Redbricks Intranet Collective[4]: Es una comunidad creada en Manchester en 1998 cuya finalidad es proveer de acceso a Internet rápido y fiable a sus miembros.
- Athens Wireless Metropolitan Network[5]: Es una iniciativa de red comunitaria surgida en Atenas en el 2002 cuyo objetivo es proveer de una red que compita con los altos precios de las ISP convencionales (Kloc, 2013).
- Freifunk[6]: Una red creada en Alemania en el 2003 con el objetivo de construir una red descentralizada, libre y neutral.
- Guifi.net: Una red comunitaria surgida en Cataluña que será objeto de análisis en este proyecto.

3.4.2. Guifi.net

3.4.2.1. Creación

Como se ha descrito existen multitud de redes comunitarias con distintos objetivos a lo largo del mundo. Pero el ejemplo que más interesa en este proyecto es Guifi.net una red abierta, libre y neutral que fue creada en 2004 en Cataluña, concretamente en una localidad llamada Gurb.

Esta comunidad ha ido creciendo y expandiéndose por el mundo hasta conseguir convertirse en la red comunitaria más grande del mundo, con 33.764 nodos activos en 2017 (Guifi.net, 2017).

[4] https://redbricksnetwork.wordpress.com
[5] http://awmn.net/content.php?s=ef30b0fecf53470c7d23253e242cabe2
[6] https://freifunk.net/

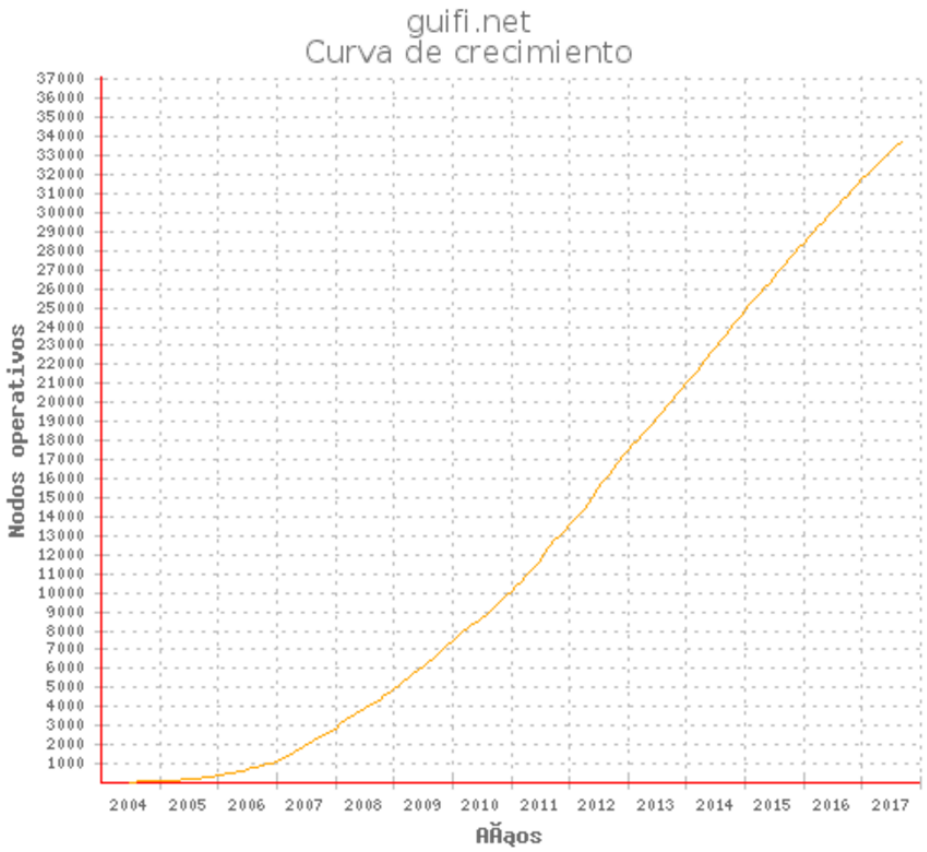


Figura: 3.11. Curva de crecimiento de Guifi.net.

Fuente: (Guifi.net, 2017).

3.4.2.2. Principios

Esta red surgió con el objetivo de llevar servicios de red a zonas en las que los operadores convencionales no tenían interés.

Al principio empezó como una iniciativa pequeña, pero rápidamente se empezó a ver un crecimiento muy pronunciado, que se debe en parte a los valores sobre los que se cimienta esta red.

Estos se resumen en 4 principios generales (Baig, Roca, Freitag, & Navarro, 2015) (Guifi.net, 2010):

- Libertad de uso de la red siempre y cuando no se perjudique el funcionamiento de la red ni al resto de usuarios.
- Libertad de conocer los componentes y el funcionamiento de la red.
- Libertad de prestar servicios a la red en las condiciones que se desee.
- Libertad de formar parte de la red.

3.4.2.3. Estructura de la red

La topología de la red tiene formato de malla y la componen dos partes generales, el *backbone* o parte troncal, que está formada por supernodos conectados entre ellos mediante conexiones de punto a punto. Y la parte de los nodos clientes, que conforman el tramo final de la red.

Para crear la estructura de la red se han usado dos tecnologías: WiFi y fibra óptica.

WiFi

Al principio se empezó a usar WiFi, esto se debe a que era una tecnología barata y sobre todo fácil de implementar. Al ser una tecnología inalámbrica no se necesita ningún tipo de permiso especial para implantarla.

Para construir la red WiFi se utilizaron dos tipos de antenas, por una parte, los supernodos. Estos forman la parte central de la red, están compuestos por un router con radio y antenas conectado en modo bridged a un router normal que hace las funciones de routeo.

Estos supenodos ofrecen conexión a los nodos cliente en un radio aproximado de 2km y se conectan a otros supernodos en una distancia de hasta 20km en línea recta.

Figura: 3.12. Supernodo de guifi.net alimentado por energía solar.

Fuente: (Guifi.net, 2012)

Por otra parte, se encuentran los nodos clientes, estos nodos forman la periferia de la red y son usados por los usuarios finales para conectarse a un supernodo.

Figura: 3.13. Nodo cliente SXT 5Hdn.

Fuente: (Mikrotik, s.f.)

Fibra óptica

En un principio usar WiFI parecía la mejor alternativa, pero a la larga se empezó a ver que requería mucho coste de mantenimiento, en cambio la fibra óptica una vez superada la desventaja inicial de su alto coste a la hora de desplegarse, se convertía en una solución que ofrecía más fiabilidad y muchísima más velocidad.

Por lo tanto, se priorizó la fibra óptica en los casos en los que fuese posible realizar una primera inversión más fuerte y las administraciones locales facilitasen el despliegue de la misma. Sobre todo, es interesante desplegarla en la parte troncal de la infraestructura de red.

3.4.2.4. Servicios disponibles

Dado que todo el mundo es libre de prestar los servicios que desee en la red, la oferta que se puede encontrar es muy variada, en la web se pueden ver todos los servicios que se ofrecen, desde acceso a Internet hasta servicios de VoIP[7],.

[7] https://guifi.net/es/node/17711/view/services

4

4. Análisis del problema

Después de conocer los objetivos del trabajo y los antecedentes tecnológicos que existen se desemboca naturalmente a el análisis del problema. En este apartado se plantea qué es lo que se le va a pedir exactamente a el sistema resultante de este TFG y mediante el análisis y la comparativa de las soluciones existentes se buscará la que más se ajuste a las necesidades del proyecto

4.1. Análisis de requisitos

El sistema resultante de este TFG tiene que ser una infraestructura capacitada para comunicar sensores a Internet en zonas en las que el acceso a Internet es complicado.

Por lo tanto, la conexión entre los sensores y los gateways debería ser de largo alcance, de esta forma se facilitaría que el gateway se pudiese colocar en zonas donde acceder a Internet sea más sencillo.

Además, la solución propuesta debería poder almacenar la información y mostrarla en algún servicio web.

Por último, la solución entera debería estar preparada para en fases posteriores del proyecto ser una infraestructura de red segura y capacitada para dar servicio a grandes cantidades de usuarios.

4.2. Análisis de las soluciones

4.2.1. Cómo comunicar los sensores

Para comunicar los sensores a los gateways el primer requisito que tiene que cumplir esta solución es que la conexión sea de largo alcance, por lo tanto, protocolos como Bluetooth, RFID o los estándares inalámbricos del IEEE 802 quedan descartados.

Otra de las características principales que se le podría pedir al protocolo es que el consumo de energía fuese bajo, esto sería de gran ayuda porque aumentaría la autonomía de los sensores.

En este punto, desembocamos en las tecnologías LPWAN comentadas en el apartado anterior, en el mismo ya se ha hecho una comparativa exhaustiva entre ellas. Como lo que se quiere en este proyecto es hacer una red privada, entre las soluciones posibles se ha escogido LoRAWAN, un protocolo abierto que como podemos ver en Tabla 3.1 de entre las tecnologías LPWAN que ofrecen más alcance es la única que ofrece la posibilidad de crear una red privada, ya que, la única que le supera en rango, Sigfox, pertenece a una empresa privada que alquila la infraestructura de red.

4.2.2. Cómo comunicar el gateway con Internet

En este caso el mismo proyecto establece cuál es la tecnología que se va a usar. La empresa GISA dispone de posiciones ideales para los gateways en las que hay conexión con Guifi.net, por lo tanto, esta es la solución más apropiada para este proyecto ya que, aunque existan otras tecnologías como las LTE, la infraestructura para conectarse a Guifi.net ya está implementada y, por lo tanto, es la solución más económica y apropiada.

A pesar de esto, existen multitud de tecnologías, como las LTE mencionadas anteriormente, que podrían servir de solución en otros escenarios, y el usarlas no modificaría ningún otro aspecto del diseño.

4.2.3. Servidor de red y servidor de aplicaciones

A la hora de elegir un servidor de red y uno de aplicaciones para una red LoRaWAN existen bastantes opciones en el mercado, se ha hecho una lista clasificándolas según si permiten gestionar por uno mismo el servidor, o si son servidores que ofrece una empresa (se muestran con un asterisco las soluciones de código abierto).

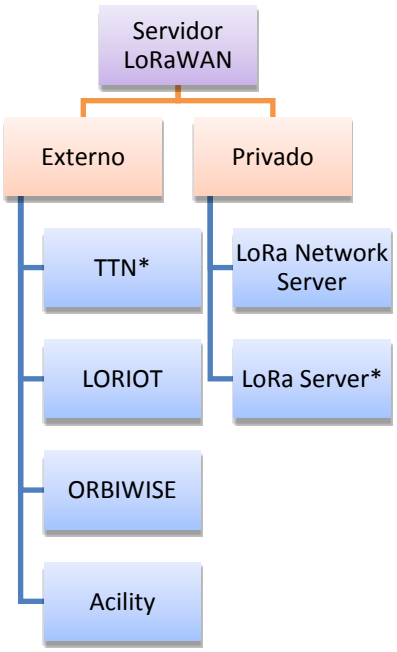


Figura: 4.1. Soluciones de Servidor LoRaWAN.

Fuente: Diseñado por el autor.

En el caso de este proyecto la rama correspondiente a los servidores externos se descarta porque se quiere tener una instancia autogestionada del servidor, por lo tanto, entre los servidores privados existen dos: el LoRa Network server, que es un servidor integrado en los gateways de Multitech y LoRa Server[8], una solución de código abierto para un servidor de LoRaWAN completo (Servidor de red y servidor de aplicaciones).

En caso de adquirir un gateway Multitech es una opción muy viable usar el servidor que integran los propios gateways, a pesar de esto, se ha decidido decantarse por LoRa Server,

[8] https://www.loraserver.io/

entre otras cosas porque ofrece una solución completa, de código abierto y con una gran comunidad por detrás. Además, al tener el servidor de red separado del gateway permite al gateway centrar su trabajo en recibir los paquetes y reenviarlos, y deja la carga de trabajo de un servidor a máquinas más potentes.

4.2.4. Permanencia de datos

Las bases de datos convencionales no están preparadas para el mundo IoT, esto ocurre porque las bases de datos tradicionales no han sido diseñadas para un escenario en el que cantidades muy grandes de sensores van emitir datos continuamente.

Además, se requiere que estos datos lleven marcas de tiempo, porque normalmente se quiere saber a tiempo real los datos de los sensores.

Para suplir esta demanda se crearon las *time series databases*, es decir, bases de datos orientadas al tiempo, estas bases de datos están preparadas para tratar grandes cantidades de datos a tiempo real, existen multitud de ellas, por mencionar las más conocidas, están InfluxDB[9] o Graphite[10].

Ambas son soluciones de código abierto muy populares, aun así, InfluxDB creada en 2013, 7 años más tarde que Graphite parece ser una solución más fácil de configurar y con un lenguaje más similar a SQL.

4.2.5. Mostrar los datos

A la hora de mostrar los datos almacenados, una solución para consultarlos podría ser realizar consultas simples a la base de datos o mostrarlos en un servicio web.

Hoy en día existen soluciones como Grafana[11], que ofrecen una plataforma que permite visualizar todos los datos en un navegador web.

Además, Grafana ofrece integración con la base de datos InfluxDB, de esta manera facilita conectar y visualizar datos almacenados en InfluxDB.

4.3. Solución propuesta

Resumiendo, la solución que se propone es usar el protocolo LoRaWAN para enviar los datos desde los sensores hasta la gateway, conectar la gateway a Internet mediante la red comunitaria Guifi.net y enviar los datos a el servidor LoRa Server, en este punto, los datos se

[9] https://www.influxdata.com/time-series-platform/influxdb/
[10] https://graphiteapp.org/
[11] https://grafana.com/

podrán almacenar en la base de datos InfluxDB y visualizar desde un navegador web mediante Grafana.

5

5. Diseño de la solución

En este capítulo basándose en las decisiones tomadas en la fase de análisis se presentan las herramientas, tanto hardware como software, de las que se ha valido el autor para desarrollar la solución. Además, se expondrá la arquitectura concreta de la solución diseñada.

En este capítulo del proyecto cabe destacar que debido a que el gateway no llegó en la fecha esperada se han planteado dos soluciones.

La primera, para superar esta contingencia plantea un sistema en el que el gateway y los sensores se sustituyen por un simulador. La segunda solución es la solución más natural, en esta se utiliza hardware real.

5.1. Análisis del hardware

5.1.1. Sensor

Como nodo se ha utilizado la placa DM164138 de Microchip, esta es una placa de demostración que proporciona un puente USB a UART al módulo RN2483.

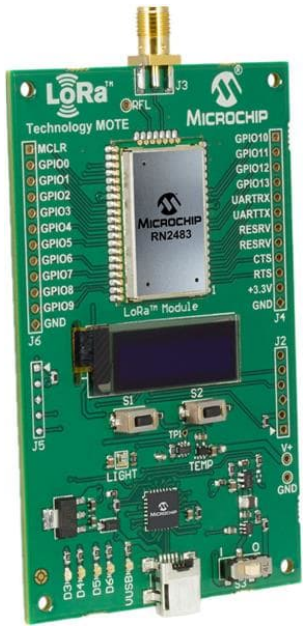


Figura: 5.1. Placa DM164138 de Microchip.

Fuente: (Microchip, s.f.).

Es posible conectarse directamente desde un PC mediante el conector USB Mini-B. De esta manera es posible enviar comandos al módulo RN2483.

Algunas de las características principales de la placa son:

- Panel LCD.
- Alimentación mediante baterías.
- Incluye antena.

Y algunas de las características principales del módulo RN2483 son:

- Opera tanto en 433 MHz, como en 868 MHz.
- Implementa la pila de protocolos de clase A de LoRaWAN.
- Tamaño pequeño.
- Consumo reducido.
- Potencia de señal de hasta 14 dBm.

5.1.2. Gateway

El gateway que se ha utilizado para la implementación es el *MultiConnect® Conduit™ MTCDT-H5-246A-US-EU-GB* (**Error! Reference source not found.**), el cual lleva una tarjeta de LoRa *MTAC-LORA-H-868* (Figura: 5.3).

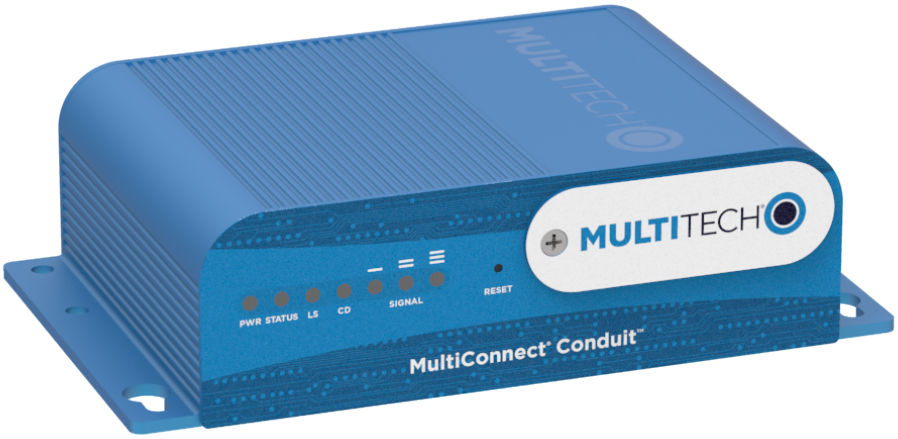


Figura: 5.2. El MultiConnect® Conduit™ MTCDT-H5-246A-US-EU-GB

Fuente: Multitech

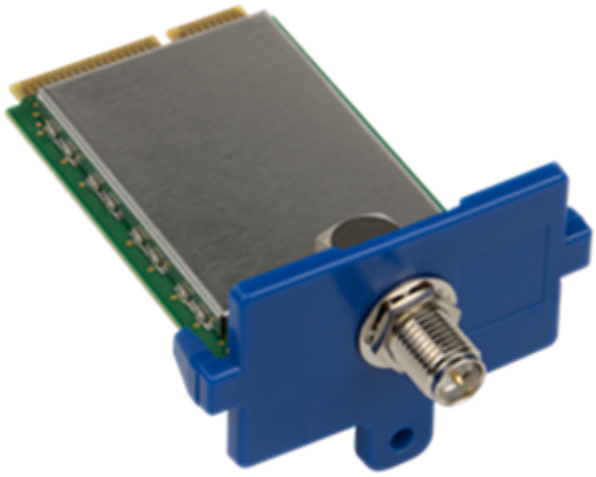


Figura: 5.3. MTAC-LORA-H-868.

Fuente: Multitech

Las características principales de este gateway son las siguientes:

- mLinux como sistema operativo.
- Puerto ethernet.
- Puerto USB Micro-B para debuggear.
- Entrada para tarjeta SIM.
- Integra un servidor de red LoRaWAN.

Las características principales de la tarjeta de LoRa son las siguientes:

- Alcance nominal de hasta 15km con contacto visual directo y 2km atravesando edificios.
- Opera en la frecuencia 868MHz.

5.1.3. Máquina virtual

Todo el software de la parte del servidor está levantado en Vagrant (Explicado en el anexo C), por lo tanto, se ha necesitado una máquina virtual en VirtualBox con las siguientes características:

- Sistema operativo: Ubuntu 16.04 (Xenial Xerus).
- Memoria RAM: 1789 MB.
- Memoria en disco: 10 GB.

5.2. Análisis del software

5.2.1. LoRa Server

El proyecto LoRa Server es un conjunto de aplicaciones que dentro de la estructura de una red LoRaWAN se colocan entre el gateway y las aplicaciones.

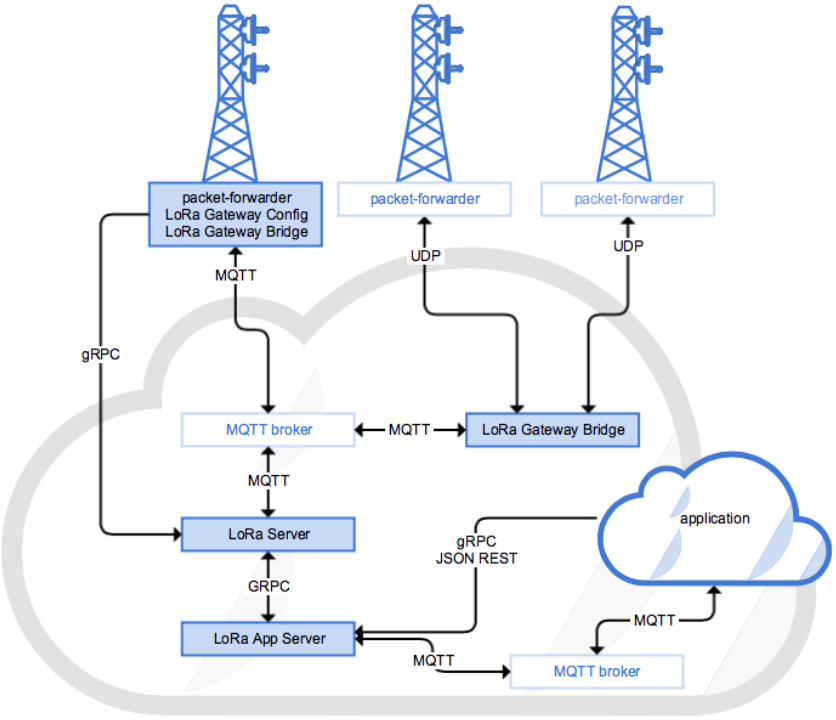


Figura: 5.4. Estructura de LoRa Server.

Fuente: (LoRa Server, s.f.)

Es decir, cumplen todas las funciones necesarias desde que un paquete sale del gateway hasta que lo recibe la aplicación pertinente. Para estas tareas LoRa Server Project se sirve de 3 herramientas (LoRa Server, s.f.):

- **LoRa Gateway Bridge**: Esta herramienta es la encargada de establecer la comunicación con el gateway, se encarga de transformar los paquetes UDP que envía el packet forwarder a JSON usando MQTT.
- **LoRa Server**: Este es el servidor de LoRa, tiene almacenadas las conexiones activas con los sensores y cuando le llega una petición de un nuevo nodo para unirse a la red es el encargado de preguntarle al servidor de aplicaciones si hay que aceptar al nuevo nodo.

Como se ha explicado anteriormente, su función como servidor de red incluye tareas como deduplicar los datos recibidos o hacer forward de los datos encriptados al servidor de aplicaciones.

Además, acepta dispositivos de clase A y C, es capaz de confirmar la recepción de paquetes, acepta ADR, permite activación de nodos mediante ABP y OTAA… entre otras características.

- LoRa App Server: Este componente es el encargado de la gestión de los nodos y los usuarios, para esta tarea ofrece la siguiente interfaz:

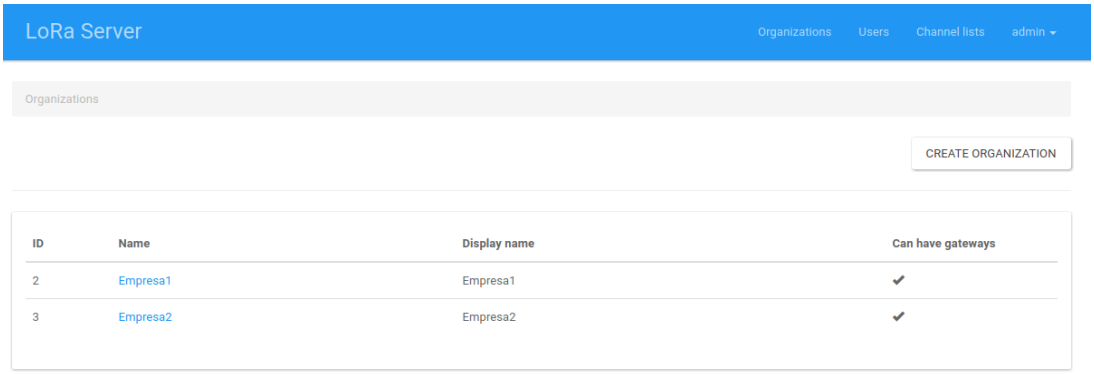


Figura: 5.5. Interfaz principal de LoRa App Server.

Fuente: Captura de pantalla realizada por el autor.

A primera vista se puede ver como el servidor permite añadir organizaciones, además de esto permite añadir usuarios, gateways, nodos y aplicaciones.

Gracias a esto, una vez configurados los datos del servidor de aplicaciones, cada vez que se reciba un paquete de un nodo el servidor comprueba a qué aplicación se corresponde el nodo y publica el mensaje en una cola de MQTT específica para ese conjunto de nodo y aplicación.

De esta manera solo los miembros de esta organización pueden conectarse a esta cola y recibir los datos de sus sensores.

5.2.2. InfluxDB

InfluxDB es una base de datos orientada a las series de tiempo de código libre, se caracteriza por la velocidad de almacenamiento y la velocidad de devolución de datos en consultas.

Además, es una base de datos preparada para analíticas en tiempo real, que te permite acceder a los datos en el momento para que fácilmente puedas identificar patrones y así predecir resultados futuros o controlar sistemas.

InfluxDB también ofrece una solución comercial que incluye clusters distribuidos para almacenar los datos creando de esta manera un sistema más escalable, con alta fiabilidad y capaz de gestionar incluso las redes IoT más grandes.

Su instalación es muy sencilla ya que está completamente escrita en GO y se compila a un simple binario sin dependencias externas.

A la hora de operar con esta base de datos a pesar de que no usa un lenguaje SQL, su lenguaje propio, InfluxQL, es muy similar a SQL así que la adaptación al mismo suele ser rápida.

5.2.3. Grafana

Grafana es una aplicación de código abierto que sirve para visualizar datos online. Además de esto ofrece la posibilidad de crear alertas dependiendo de los datos que se están recibiendo, para ayudar a la monitorización.

Una de las características más relevantes de Grafana para este proyecto es que soporta de forma nativa la integración con InfluxDB, ofreciendo un editor de consultas con consultas predeterminadas que facilitan mostrar los datos de InfluxDB.

Además, es una herramienta perfecta para usarla como servicio para varios clientes, esto se debe a que soporta multitud de métodos de autenticación y es capaz de gestionar organizaciones, asignándoles sus propios administradores, usuarios o fuentes de datos.

Una de las características que se ha mencionado, la posibilidad de visualizar datos online, para poder apreciarla es necesario ilustrarla, en la **Figura: 5.6** se pueden ver las posibilidades que ofrece Grafana a la hora de crear paneles.

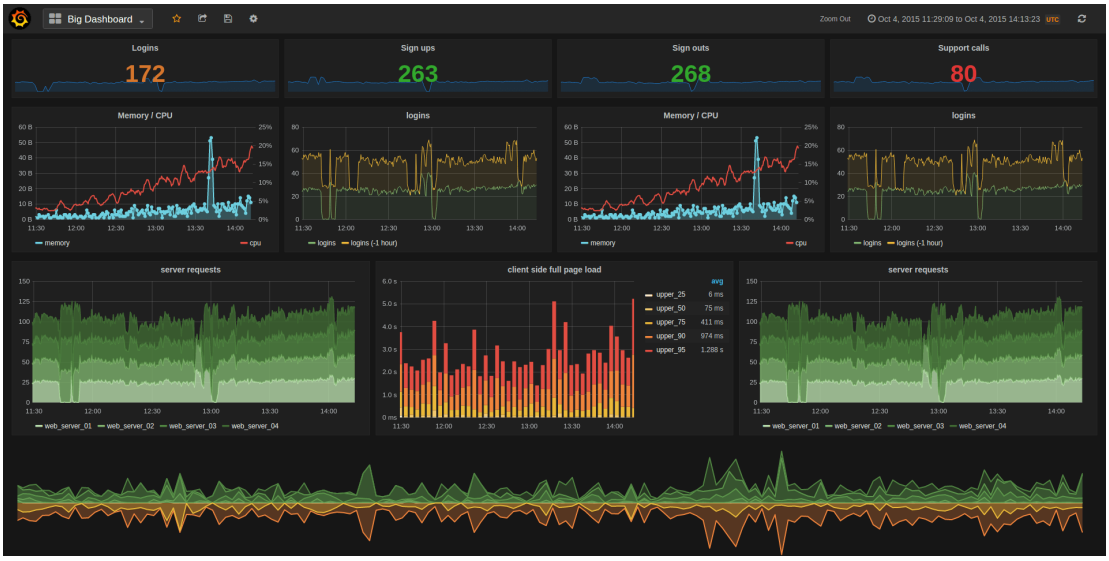


Figura: 5.6. Ejemplo de panel en Grafana.

Fuente: (Rendle, 2016).

5.2.4. Simulador de sensores y gateway

Como se ha mencionado en el inicio, al no tener el hardware disponible en la fecha prevista, se exploró la posibilidad de usar un software que simulase la función del hardware. Para esto es necesario crear paquetes que respeten las especificaciones del protocolo de comunicación entre el gateway y el servidor. Este es el protocolo del packet forwarder[12] y en el caso del backend que se está utilizando en este proyecto, tiene que respetar el protocolo de versiones superiores a la 3.0.

Existen varios generadores de paquetes que respetan este protocolo subidos en GitHub. El utilizado en este proyecto es el creado por el usuario Cambierr (Cambierr, 2016).

El problema de este simulador es que es necesario escribir los parámetros cada vez que se quiere lanzar, y a la hora de ver el funcionamiento del backend era necesario simular un envío periódico de paquetes desde diferentes fuentes, por lo tanto, se creó un script en Python que hacía que se enviasen paquetes periódicamente con datos diferentes cada vez valiéndose del simulador de Cambierr.

5.3. Estructura de red

Todos los componentes explicados en los apartados anteriores quedarían de esta manera en la red global:

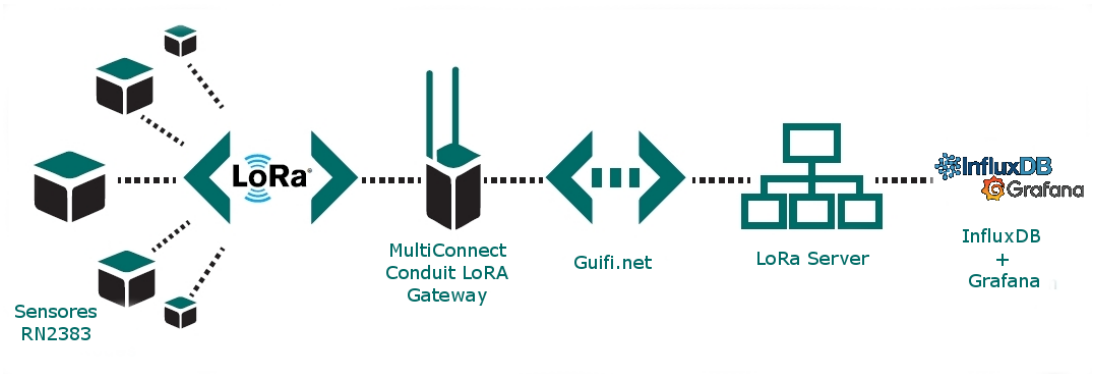


Figura: 5.7. Estructura general de la red.

Fuente: (Kumbhar, 2015). Editada por el autor.

[12] https://github.com/Lora-net/packet_forwarder/blob/master/PROTOCOL.TXT

De esta forma se conecta cada una de las herramientas descritas anteriormente, pero al tener tantas herramientas diferentes, la integración de las mismas en algunos casos puede suponer un reto.

La configuración explicita de cada una de las integraciones que se van a mencionar queda explicada en los anexos, pero merece la pena explicar de forma general las dificultades que se han encontrado.

A la hora de conectar los sensores RN2383 con el gateway ha sido necesario modificar apartados de la configuración ya que con la configuración nativa el Conduit no acepta sensores de otros fabricantes.

El gateway como se ha explicado previamente lleva integrado un servidor de red LoRaWAN, pero en esta solución lo que se había planteado era usar como servidor de red LoRa Server, por lo tanto, es necesario cambiar la configuración del gateway, suprimir sus funciones como servidor y hacer que funcione exclusivamente como packet forwarder.

Como se ve en la Figura: 5.4 el último punto del LoRa server es un broker MQTT, en este caso Mosquitto. Para recibir los datos vale con usar cualquier tipo de cliente MQTT, pero en este proyecto se planteó la posibilidad de asegurar la permanencia de estos datos valiéndose de la base de datos InfluxDB. Para hacer esto se decidió crear un script en Python que automatizase la escritura del broker MQTT a la base de datos InfluxDB y se puso a funcionar el script como servicio en el servidor de red.

Por último, queda la integración de InfluxDB con Grafana, este paso es el más sencillo, ya que, Grafana de forma nativa incluye InfluxDB como base de datos.

6

6. Implantación y Pruebas

En el apartado anterior se menciona la arquitectura concreta que tiene la solución y en este capítulo se explican los retos que conlleva implantar un sistema de estas características en un escenario real. Junto con esto se llevarán a cabo pruebas de alcance y de backend para comprobar el rendimiento del sistema.

6.1. Análisis del entorno

El entorno en el que se va a realizar la implantación es en Guipúzcoa, una provincia de la comunidad autónoma del País Vasco situada al norte de España.

Guipúzcoa se encuentra entre la Cordillera Cantábrica y los Pirineos, esto es muy relevante ya que la convierte en la segunda provincia de España con más desnivel de terreno (Gisbert & Martí, 2010).

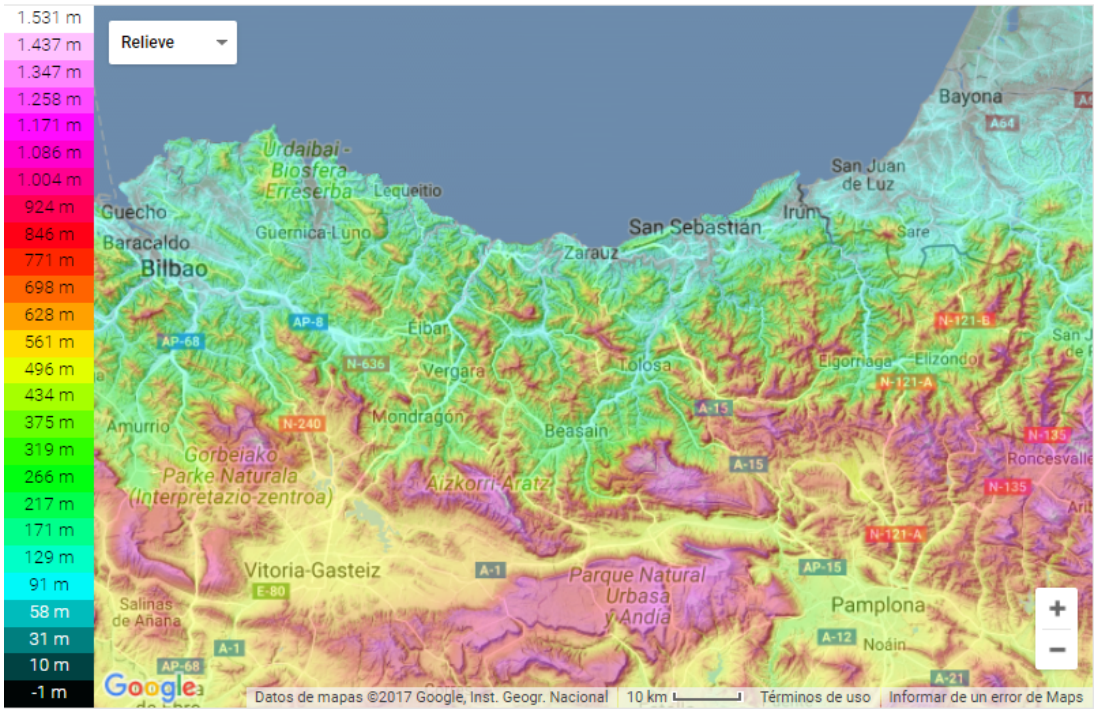


Figura: 6.1. Mapa topográfico de Guipúzcoa.

Fuente: Google Maps.

Esto hay que tenerlo en cuenta dado que la calidad de la señal empeora considerablemente si encuentra obstáculos de por medio, por ello, a la hora de colocar tanto gateways como sensores, para realizar pruebas de alcance, se tienen que utilizar herramientas que ayuden a ver los perfiles de elevación para evitar la mayor cantidad de obstáculos posible.

Una de las herramientas más potentes que proporciona esos perfiles es "*Google Earth*". En esta aplicación se puede dibujar la ruta deseada y el programa ofrece el perfil de elevación.

Otra herramienta de gran utilidad a la hora de calcular la calidad de los radioenlaces es Radio Mobile, este software una vez introducidas las características de los radioenlaces lanza una simulación que determina qué parámetros tendrá la conexión resultante.

6.2. Ubicación de gateways y sensores

En este proyecto se han creado tres situaciones como ejemplo, para todas ellas el gateway se ha colocado en una ventana del tercer piso de la Facultad de Informática de la UPV/EHU. En los perfiles de elevación no es posible tener en cuenta este factor, por lo tanto, se tienen que sumar aproximadamente unos 10 metros de altura a la coordenada en la que se encuentra el gateway.

Figura: 6.2. Posición del gateway.

Fuente: Fotografía tomada por el autor.

6.2.1. Situación 1

La primera situación es la más simple, y su objetivo principal es testar que el sistema funciona, la distancia es de 91,4 metros y hay una línea de visión muy clara entre ambos puntos.

Figura: 6.3. Localización geográfica de la situación 1.

Fuente: Google Earth 7.3.0.3832

6.2.2. Situación 2

La segunda situación tiene un perfil de elevación más variado, a pesar de esto se puede apreciar como existe línea de visión directa entre ambos puntos. La distancia de este trayecto son 216 metros.

Figura: 6.4. Localización geográfica de la situación 2.

Fuente: Google Earth 7.3.0.3832. Editada por el autor.

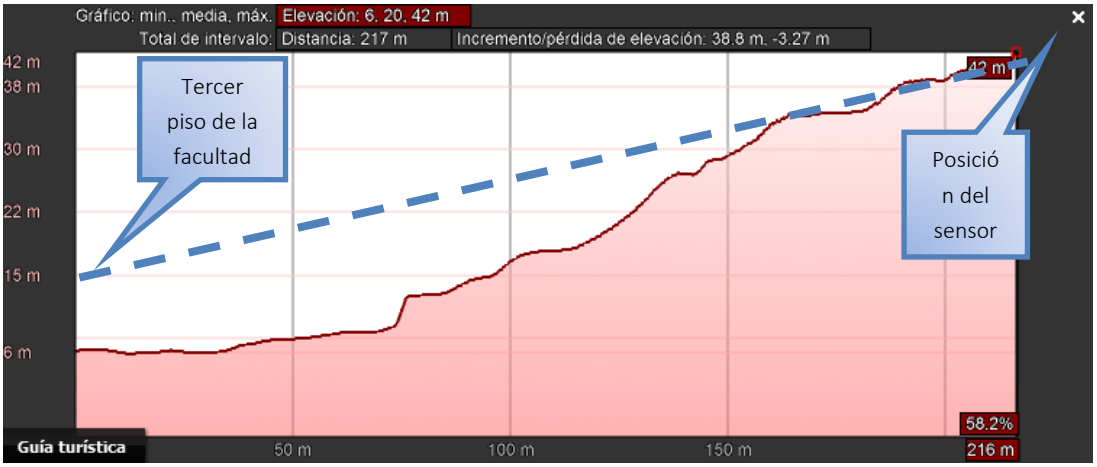


Figura: 6.5. Perfil de elevación de la situación 2.

Fuente: Google Earth 7.3.0.3832. Editada por el autor.

6.2.3. Situación 3

Por último, la situación 3 tiene el perfil de elevación más complejo, a pesar de que se llega a ver el gateway desde la posición del sensor, hay partes de algunos edificios (Chimeneas metálicas) y vegetación que estorban el camino directo. Además, este es el trayecto más largo cubriendo aproximadamente medio kilómetro de distancia.

Figura: 6.6. Localización geográfica de la situación 3.

Fuente: Google Earth 7.3.0.3832.

Figura: 6.7. Perfil de elevación de la situación 3.

Fuente: Google Earth 7.3.0.3832. Editada por el autor

6.3. Pruebas de alcance

Las pruebas de alcance serán 3, se hará una por cada una de las situaciones descritas en el apartado anterior.

Los sensores, se transportarán junto con un portátil y después de una secuencia de pruebas iniciales para verificar que todo está en orden se enviarán 5 paquetes.

Solo se envían 5 paquetes debido a las limitaciones que tiene LoRaWAN a la hora de gestionar el volumen de datos (**3. Estado del arte**).

Cada una de las rutas tiene un objetivo distinto:

- **Situación 1**: Probar que el sistema funciona y se pueden recibir todos los mensajes sin sufrir pérdidas. Para este propósito se colocará la antena del sensor apuntando directamente hacia el gateway.
- **Situación 2**: Comprobar que se reciben todos los mensajes a distancias medias a pesar de que las condiciones no sean las óptimas. Para este objetivo se colocará el sensor apuntando en la dirección contraria a la que se encuentra el gateway.
- **Situación 3**: Probar si a distancias grandes y con ciertos obstáculos se pueden recibir todos los mensajes. Para esta prueba se apuntará directamente a la posición en la que se encuentra el gateway a través de los obstáculos.

Figura: 6.8. Envío de mensajes con el sensor.

Fuente: Fotografía tomada por el autor.

6.3.1. Resultados esperados

Para investigar cuales son los resultados que se pueden esperar en cuanto al alcance se ha usado como base el estudio realizado por Eduardo José Córdoba Peñalver (Peñalver, 2017).

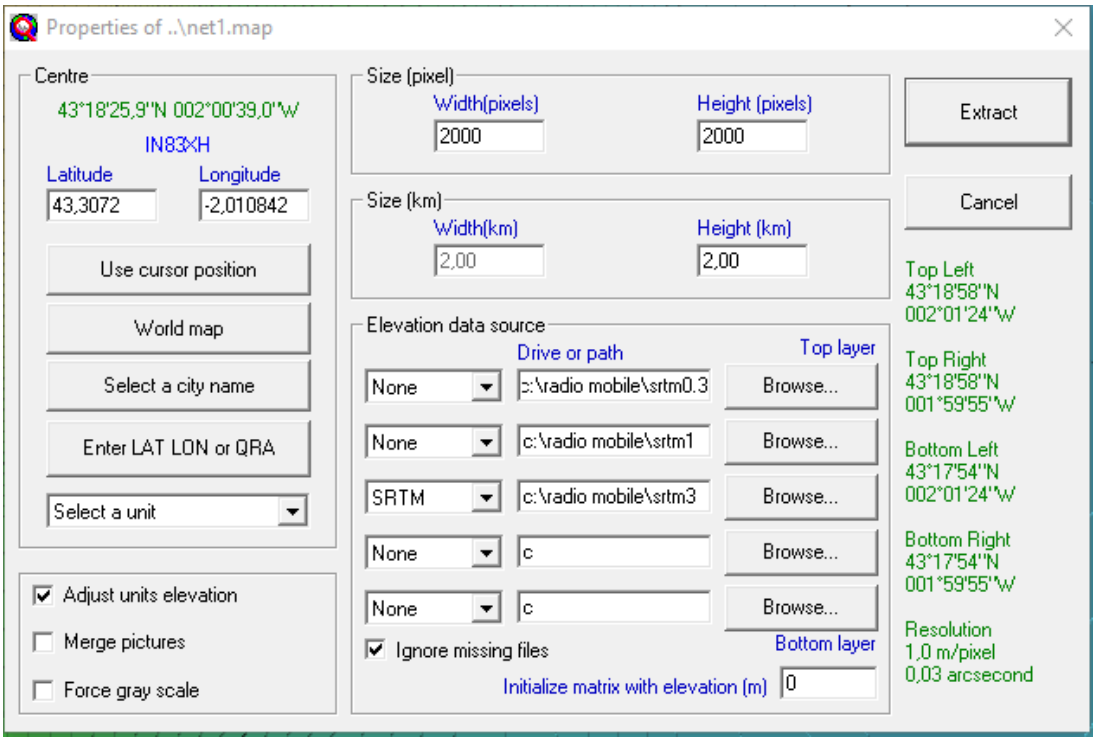
Se utilizará el software Radio Mobile[13] que permite hacer una simulación de la red y calcular las características de los radioenlaces.

Este software es idóneo para estas pruebas, porque permite describir las características de los radioenlaces con mucha exactitud. La aplicación da la posibilidad de seleccionar datos como cuál es el clima en la zona o la dirección exacta en la que apunta una antena. Esto es muy útil por ejemplo para simular la ruta 2 en la que la antena no está mirando directamente al gateway.

Además, cabe destacar que estas pruebas se han podido hacer a pesar de no tener el gateway.

6.3.1.1. Diseño de la red en Radio Mobile

Lo primero que hay que hacer para diseñar la red es conseguir las coordenadas de la zona en la que se van a realizar las pruebas para ubicar el centro del mapa. En este caso la latitud es de 43.307203 y la longitud de -2.010842.



[13] http://radiomobile.pe1mew.nl/?Installation:Download

Figura: 6.9. Configuración general del mapa.

Fuente: Radio Mobile 11.6.5.

El siguiente paso consistiría en describir todos los elementos que conforman la red, para ello lo primero será recopilar sus coordenadas.

Enlaces	Latitud	Longitud
Gateway ruta 1	43,3070556	-2,010411111
Gateway ruta 2	43,3072194	-2,010930556
Gateway ruta 3	43,3070556	-2,010411111
Sensor ruta 1	43,3069694	-2,009380556
Sensor ruta 2	43.3062806	-2,013100000
Sensor ruta 3	43,3076083	-2,004991667

Tabla 6.1. Coordenadas de los enlaces.

Fuente: Creada por el autor.

Una vez recopiladas estas características junto con otras como la altura de los enlaces, se introducen en la interfaz *unit properties.*

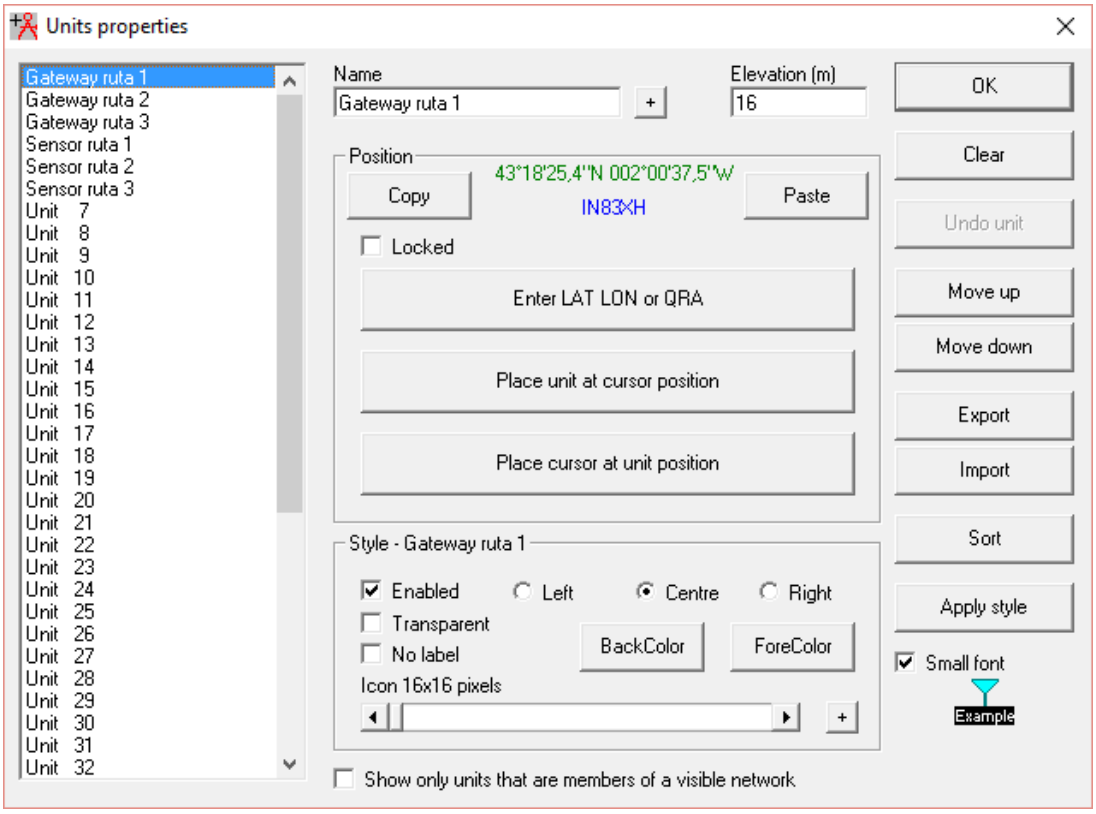


Figura: 6.10. Propiedades de los elementos de la red.

Fuente: Radio Mobile 11.6.5.

Y el programa devuelve un mapa con el posicionamiento de todos los elementos de la red.

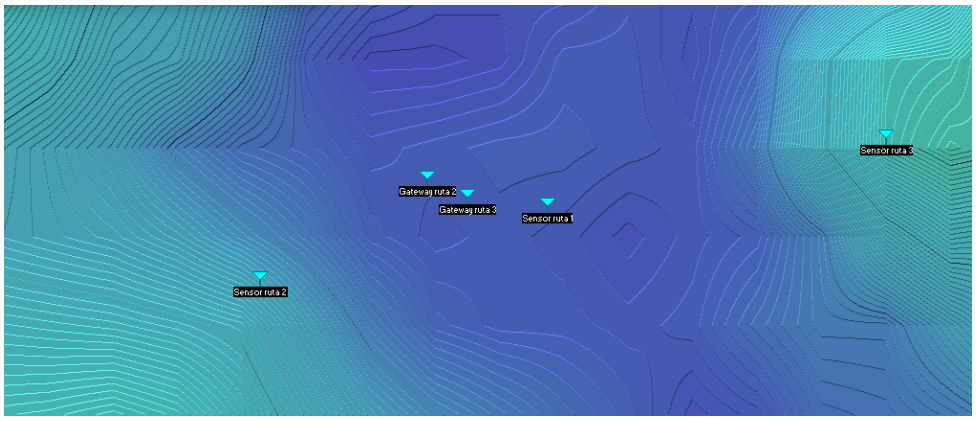


Figura: 6.11. Posicionamiento de los elementos de red.

Fuente: Radio Mobile 11.6.5.

Una vez se describen todos los elementos de la red y el software devuelve un mapa con su posicionamiento, hay que describir las características de los radioenlaces para las 3 rutas.

Para esta tarea Radio Mobile ofrece 3 ventanas principales:

- **Parameters**: En esta ventana se puede designar la frecuencia de los radioenlaces y las condiciones climáticas, entre otras características como la pérdida adicional por encontrarse en condiciones no óptimas (Ciudad o selva).

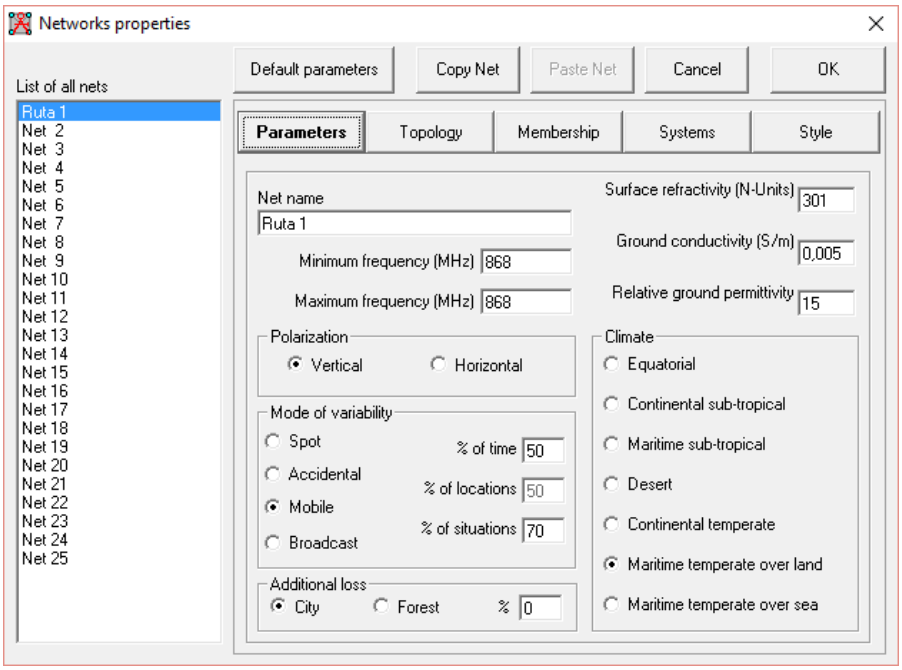


Figura: 6.12. Interfaz de selección de parámetros de la red.

Fuente: Radio Mobile 11.6.5.

- **Topology:** En esta parte de la interfaz se puede seleccionar la topología de la red, en el caso de las tres rutas que se describen en este proyecto la topología que se va a seleccionar es la de estrella de tipo master/slave.

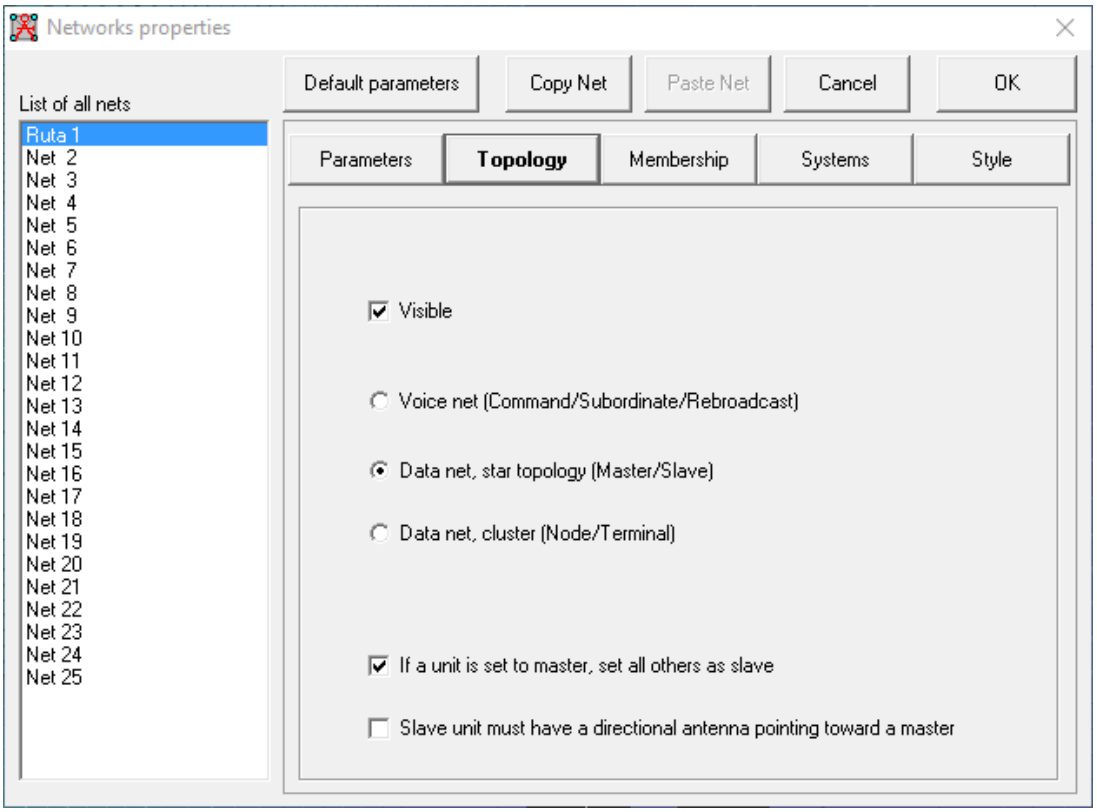


Figura: 6.13. Interfaz de selección de topología de la red.

Fuente: Radio Mobile 11.6.5.

- **Systems**: En esta ventana se describirán los dos tipos de enlaces que existen en la red, es decir, gateways y sensores. Los datos que se piden abarcan desde datos como la potencia de transmisión o la sensibilidad de la antena hasta datos como el tipo de antena y su ganancia. Para saber cuáles son estos datos hay que dirigirse a las páginas de especificación del hardware que ofrecen los fabricantes, (Multitech, 2017) y (Microchip, 2017), para gateways y sensores respectivamente.

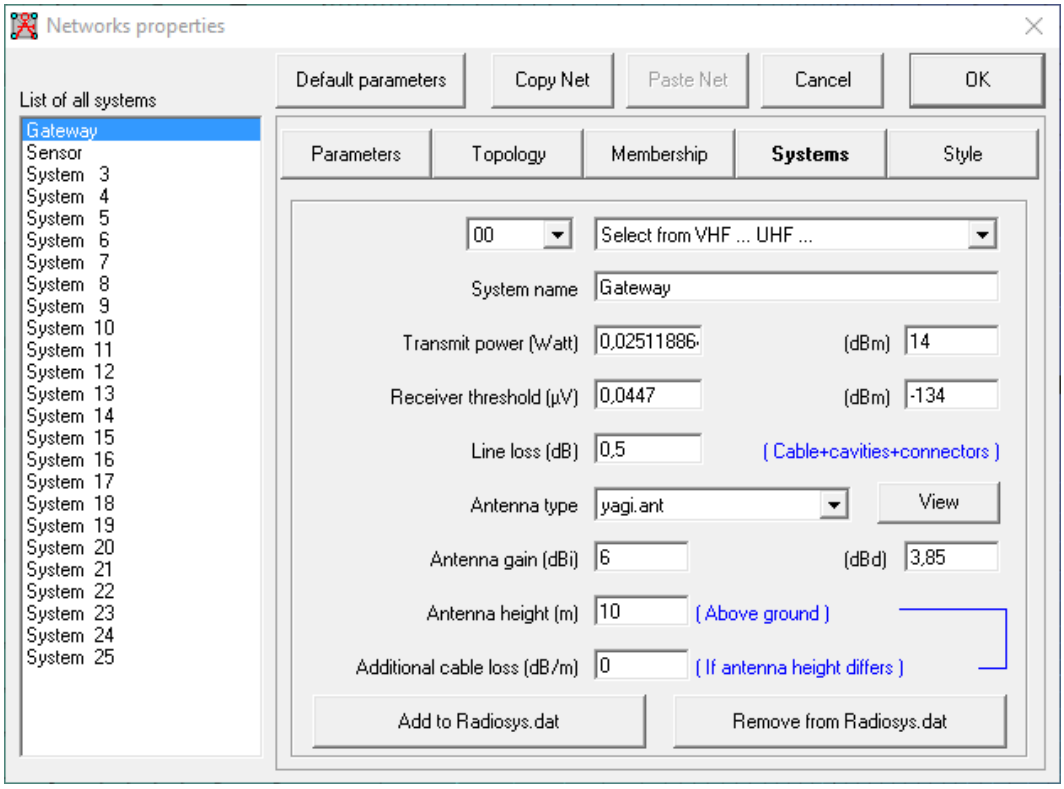


Figura: 6.14. Interfaz de descripción de sistemas (Gateway).

Fuente: Radio Mobile 11.6.5.

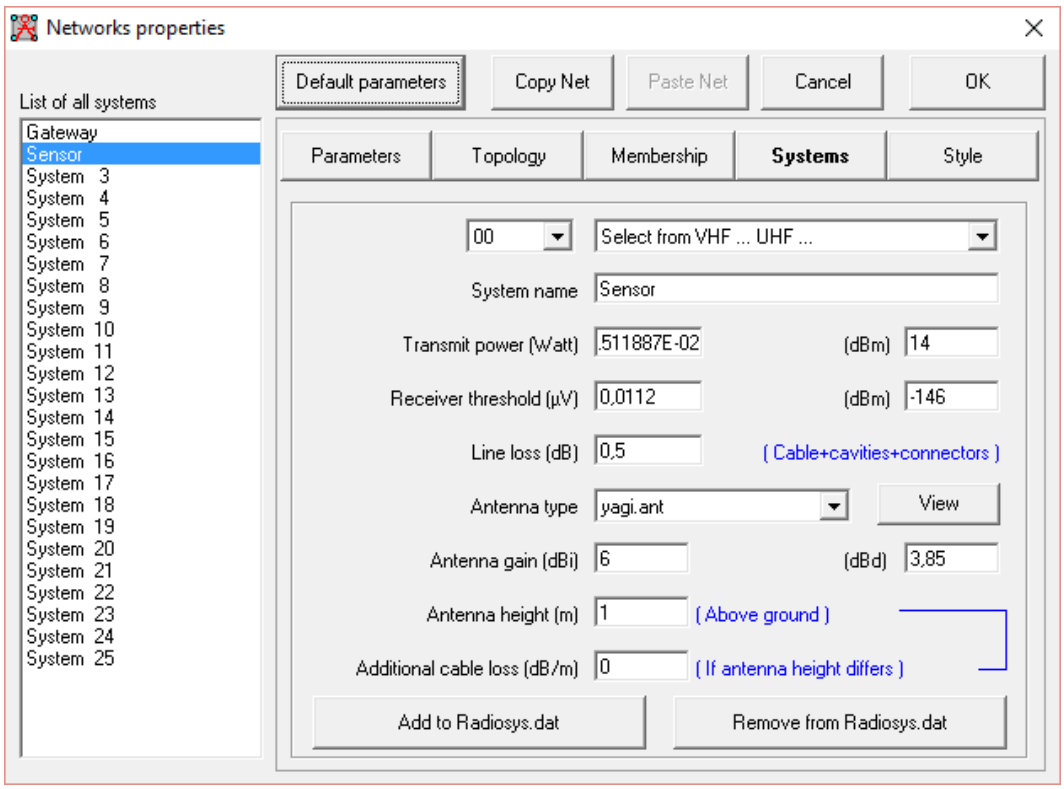


Figura: 6.15. Interfaz de descripción de sistemas (Sensor).

Fuente: Radio Mobile 11.6.5.

6.3.1.2. Resultados de la simulación en Radio Mobile

En este punto, una vez descrita la red en su totalidad, Radio Mobile devuelve un mapa con todos los enlaces. En este caso los dibuja de color verde, lo que quiere decir que la calidad de los radioenlaces es buena.

Figura: 6.16. Mapa de los radioenlaces de la ruta 1-3.

Fuente: Radio Mobile 11.6.5.

Por último, se pueden ver las características específicas de cada uno de los radioenlaces, el programa lo muestra en las interfaces de las dos figuras siguientes.

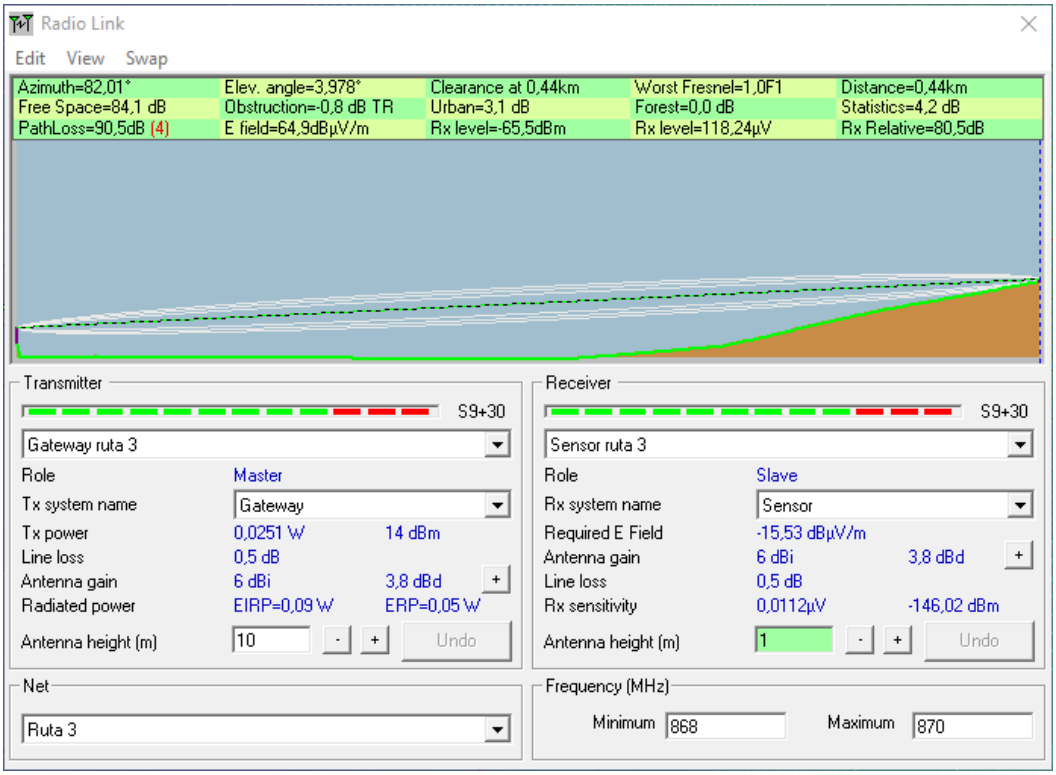


Figura: 6.17. Estadísticas del radioenlace de la ruta 3.

Fuente: Radio Mobile 11.6.5.

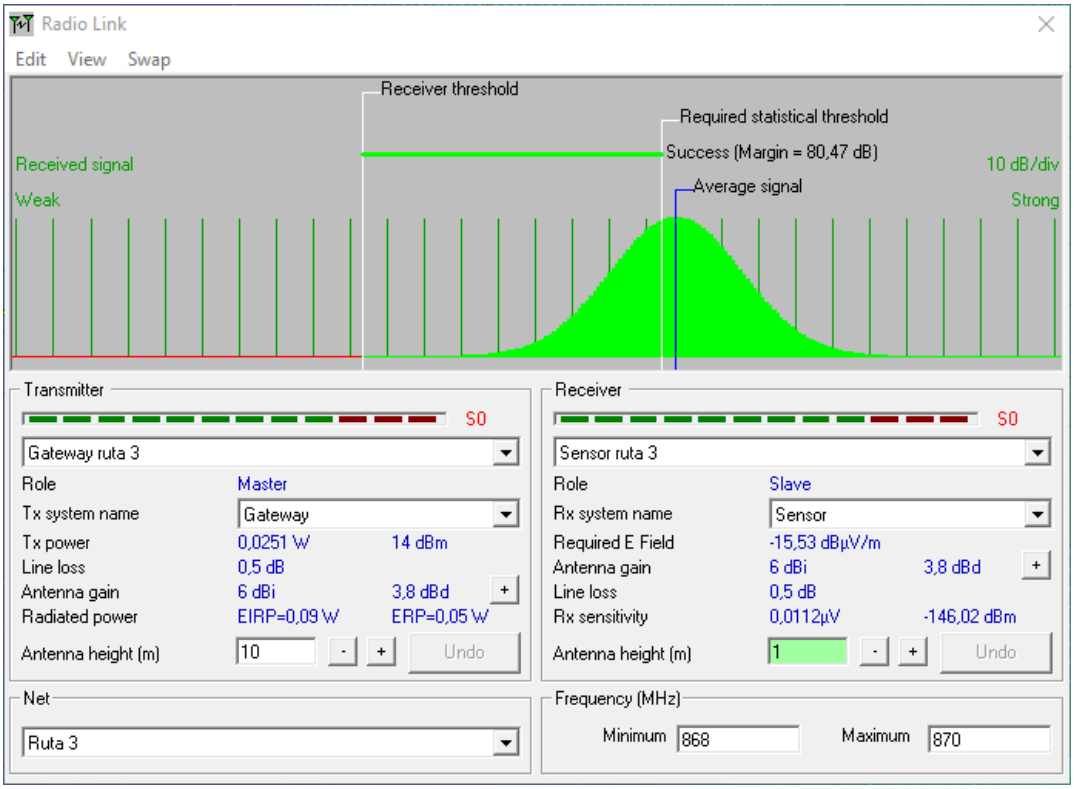


Figura: 6.18. Estadísticas del radioenlace de la ruta 3 (Distribución).

Fuente: Radio Mobile 11.6.5.

Con el fin de resumir los datos se ha creado la siguiente tabla con los más importantes y se extraerán conclusiones de los resultados que se deberían obtener.

Ruta	Distancia	Perdida en el trayecto	Obstrucción	RSSI
Ruta 1	80 m	71,2 dB	0	-46,2 dBm
Ruta 2	200 m	81,2 dB	0	-74,5 dBm
Ruta 3	440 m	84,1 dB	0,8 dB	-65,5 dBm

Tabla 6.2. Datos extraídos de la simulación de los radioenlaces.

Fuente: Radio Mobile 11.6.5.

El dato que más no interesa de la tabla anterior es el RSSI (*Received Signal Strength Indicator*) o Indicador de fuerza de la señal recibida. Este valor determina la calidad de una conexión inalámbrica.

Este dato es algo relativo, pero en (Metageek, s.f.) se puede encontrar una tabla que especifica qué esperar de la calidad de una señal teniendo en cuenta su RSSI.

RSSI	Descripción
-30 dBm	Máximo posible de fuerza de señal, para que se dé este caso el sensor y el gateway tienen que estar muy cerca el uno del otro.
-67 dBm	Mínimo de fuerza de señal para aplicaciones que requieran de poco delay, por ejemplo VoIP o streaming de vídeo.
-70 dBm	Mínimo de fuerza de señal para entrega de paquetes fiable.
-80 dBm	Mínimo de fuerza de señal para conectividad fiable, puede que haya pérdida de paquetes.
-90 dBm	Es poco probable que funcione la comunicación en estos niveles.

Tabla 6.3. Calidad de la señal respecto a su RSSI.

Fuente: (Metageek, s.f.), Editada por el autor.

Comparando esta tabla con los valores obtenidos se puede hacer la siguiente hipótesis: Todos los paquetes que se envíen en las pruebas descritas se tienen que recibir correctamente.

6.3.2. Resultados obtenidos

Como se mencionó, para cada una de las rutas se hizo la comprobación inicial y posteriormente un envió de 5 paquetes. En todos los casos se recibieron todos los paquetes enviados. Por lo tanto, la hipótesis derivada de la simulación: *"Todos los paquetes que se envíen en las pruebas descritas se tienen que recibir correctamente"* queda confirmada.

6.4. Integración en Guifi.net

Los detalles de cómo realizar una conexión con Guifi.net quedan explicados en el anexo, pero cabe destacar en este apartado qué fue necesario para realizar esta tarea en un escenario real, a nivel de hardware.

Para conectarse a un supernodo de Guifi.net existen dos formas principales, por ethernet o usando una antena direccional de 5 Ghz.

En este caso se utilizó una antena como las que se pueden ver en la parte izquierda de la Figura: 6.19.

Figura: 6.19. Antena direccional de 5 Ghz y gateway.

Fuente: Fotografía tomada por el autor.

La antena debe colocarse en vertical y paralela al supernodo, con el que debe tener contacto visual.

Figura: 6.20. Posicionamiento de la antena respecto al SuperNodo.

En el caso de esta implantación que se hizo, que no es la definitiva, no había acceso a la corriente eléctrica así que se tuvo que utilizar un sistema de alimentación ininterrumpida para llevar a cabo las pruebas.

Figura: 6.21. Sistema de alimentación ininterrumpida.

Fuente: Fotografía tomada por el autor.

6.5. Pruebas del backend

La prueba de backend tiene la finalidad de testear el correcto funcionamiento del mismo, para esta tarea se va a crear un escenario en el que una organización (*Empresa1*) tiene una aplicación llamada *Prueba_Memoria* y esta aplicación tiene asignado un nodo *(Sensor_Prueba)*:

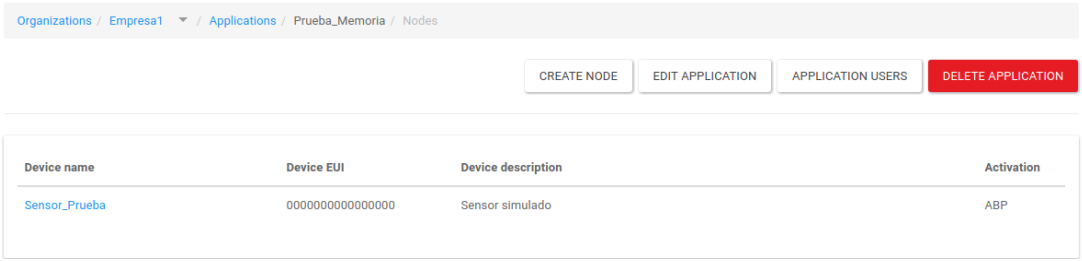


Figura: 6.22. Empresa, aplicación y nodo.

Fuente: Captura de pantalla realizada por el autor.

El nodo tiene la siguiente configuración de red:

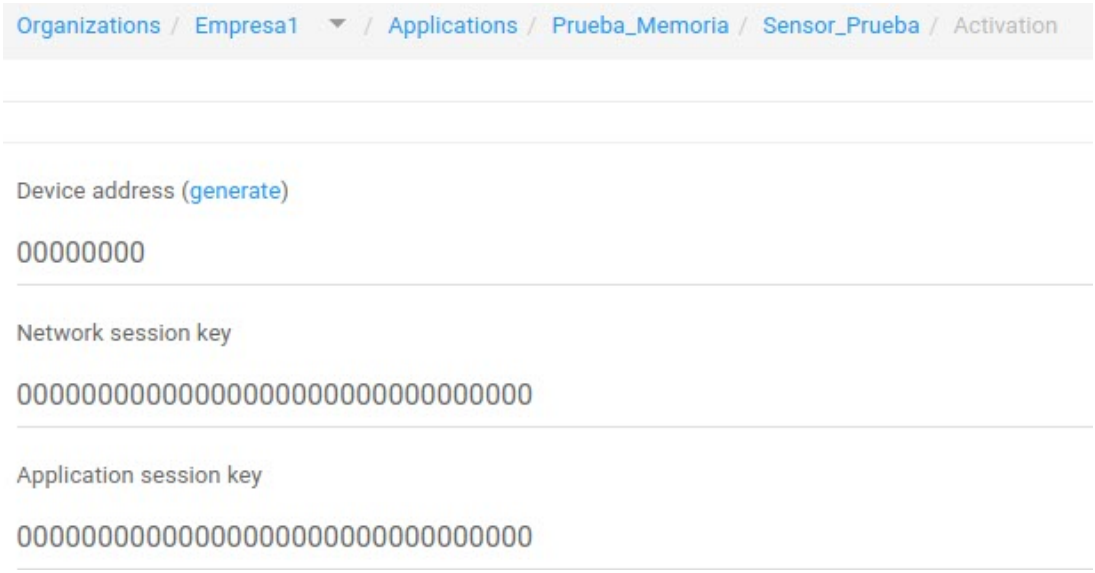


Figura: 6.23. Configuración de red del nodo Sensor_Prueba.

Fuente: Captura de pantalla realizada por el autor.

Como nodo se va a utilizar el simulador, con la siguiente configuración:

```
#!/usr/bin/python

import os
import time
import random
import string
import datetime
n = "java -jar Simulator.jar --routerHost 127.0.0.1 --devAddr
00000000 --nwkSKey 00000000000000000000000000000000 --appSKey
00000000000000000000000000000000 --plain "
i = 0

while True:
    random.seed(datetime.datetime.now())
    i+=1
    os.system(n + ''.join(random.choice(string.ascii_letters) for
_ in range(random.randint(1,50))) + "  --fCnt "+ str(i))
    time.sleep(5)
```

Script de configuración del gateway.

Este script lo que hace es simular ser un nodo con las mismas características que el que se ha descrito en el servidor de aplicaciones y manda mensajes de diferentes tamaños de forma aleatoria.

6.5.1. Resultados esperados

En la prueba de backend el resultado que se espera es que en el servidor de aplicaciones los mensajes del nodo los detecte como los del nodo *Sensor_Prueba*, esto es, que solo los miembros de la *Empresa1* puedan acceder a los datos.

Y también se espera que en grafana se vea a tiempo real los datos que se están recibiendo del sensor.

6.5.2. Resultados obtenidos

Como fue pronosticado en el servidor de aplicaciones se ve como los mensajes que se están enviando se reciben el *Sensor_Prueba,* esto se puede ver porque el parámetro *Uplink frame-counter* se incrementa conforme se mandan mensajes*:*

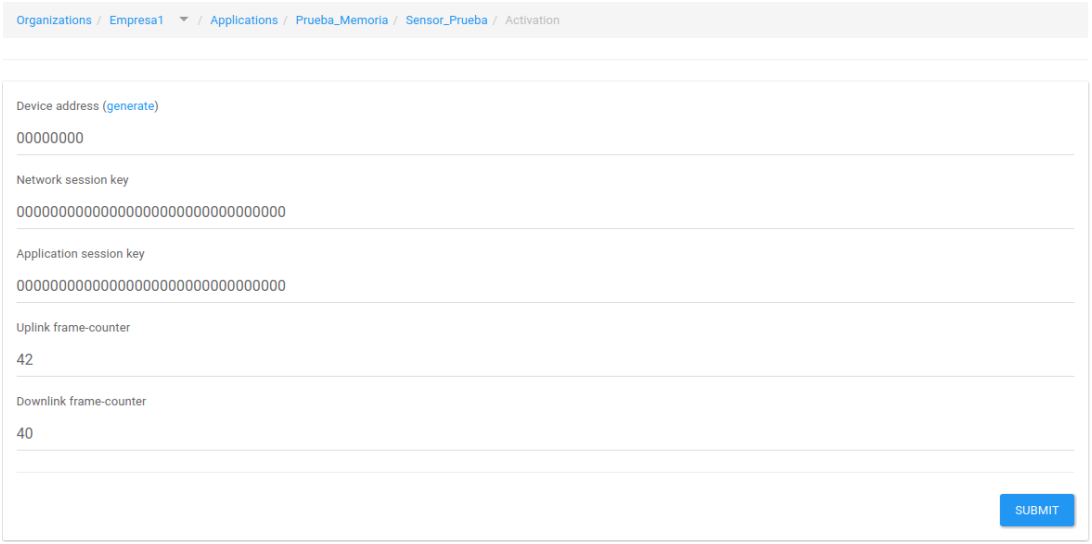


Figura: 6.24. Mensajes enviados por el nodo Sensor_Prueba.

Fuente: Captura de pantalla realizada por el autor.

Además, al conectarse a Grafana y hacer un *dashboard* para ver la base de datos, se puede apreciar como a tiempo real conforme se mandan paquetes con datos de tamaño aleatorio el gráfico va variando.

Figura: 6.25. Dashboard de Grafana para la base de datos de la prueba.

Fuente: Captura de pantalla realizada por el autor.

Por lo tanto, queda demostrado que el backend es capaz de recibir datos de un sensor, localizar correctamente de qué empresa se trata, qué aplicación es y cuál es el sensor que los

está enviando. Y, además, Grafana es capaz de recibir estos datos a tiempo real y mostrarlos en forma gráfica.

7

7. Valoración económica

En este apartado se realiza una valoración económica del proyecto, esto incluye un análisis de los costes que ha conllevado este proyecto y una propuesta de un modelo de explotación de la solución para ayudar a la empresa a valorar el modelo de negocio.

7.1. Análisis de costes

En este apartado se detallará el hardware que ha sido necesario adquirir para realizar este proyecto. Parte de este hardware se ha podido conseguir de forma gratuita porque ya estaba en stock en la universidad, pero en un proyecto real serían gastos a tener en cuenta. El hardware que ha sido necesario es el siguiente:

Adquisición	Coste real	Coste
Gateway MultiConnect® Conduit™	418,46 €	418,46 €
Tarjeta de lora MTAC-LORA-915	150,65 €	150,65 €
Antena para LoRa AN868-915A-1HRA	12,94 €	0 € Estaba en stock en la UPV/EHU
Total	582,05 €	569,11 €

Tabla 7.1: Gestión de adquisiciones.

Fuente: Creada por el autor.

Cabe destacar que a este coste habría que añadirle el coste humano, es decir, las horas que han sido invertidas en el proyecto y estas serán descritas en el capítulo 8. Gestión.

7.2. Modelo de explotación

En el apartado anterior se han visto los costes aproximados del hardware que se necesita, en este se analizan las formas de monetización más rentables.

Existen en el mercado empresas como *Loriot*[14] que se dedican a vender software y servicios relacionados con LoRaWAN. El modelo que elige para monetizar su producto esta empresa es uno basado en el coste por uso.

Es decir, ponen unos límites a la hora de cuantos gateways y sensores se pueden conectar y posteriormente dan libertad de uso y cobran según lo que se consuma.

La monetización del IoT es un tema candente, esto suscita diferentes opiniones sobre el enfoque que se le debería dar. En una entrevista (Shore, 2015), Kyle Hilgendorf,

[14] https://www.loriot.io/

vicepresidente de Gartner especializado en IoT sugiere que una de las mejores formas de monetizar es recogiendo y vendiendo los datos que se recogen, ya que muchas empresas o gobiernos están potencialmente interesados en la información que antiguamente se desechaba.

Seguidamente el entrevistador le pregunta por el método que hemos descrito antes *"pay-as-you-go"* a lo que Hilgendorf le responde: "*Es una posibilidad. No había pensado en ello, pero esa es la belleza del IoT—abre la puerta a la innovación"*[15]. Y tanto es así que la consultora Capgemini en un estudio que publicó en 2014 (Capgemini, 2014) propuso los siguientes modelos para la monetización del IoT, clasificados en un gráfico (Figura: 7.1) que valora la complejidad del modelo respecto a la relación con el cliente:

- Construir un ecosistema: En este modelo se crean plataformas compartidas para que tanto proveedores como consumidores se beneficien.
- Hardware premium: Es un modelo en el que se carga un precio extra por características extra de conectividad.
- Monetizar servicios: Este modelo se basa en convertir un producto en un servicio cobrando periódicamente por unas características específicas, como, por ejemplo, servicio de mantenimiento o cobro de suscripciones a software.
- Monetizar datos: En este modelo se cobra por vender los datos almacenados.

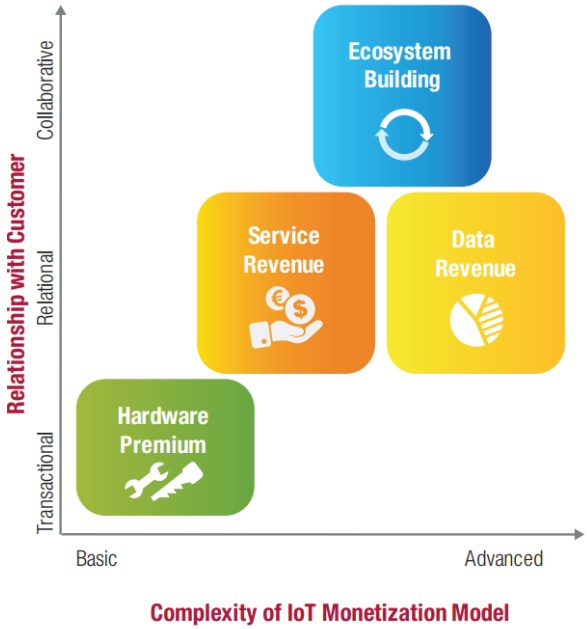


Figura: 7.1: Modelos de monetización para el IoT.

Fuente: (Capgemini, 2014)

[15] Traducido del inglés

Dentro de los diferentes modelos los dos que más se le pueden aplicar a este proyecto son los de monetizar datos y monetizar servicios. En este informe se comenta que las empresas para las que están hechos estos modelos son empresas con clientes muy comprometidos y en posición para recoger grandes cantidades de datos.

Esta descripción es exactamente a la que apunta una empresa que desarrolla una infraestructura. Quiere tener muchos clientes que contraten sus servicios para así sacarle rentabilidad y obtener muchos datos y captar clientes comprometidos al ser pionera a la hora de crear infraestructura.

Los consejos que ofrecen para estos modelos son tener subscripciones a muchos rangos de precios para poder abarcar más tipos de usuarios finales.

En conclusión, una buena forma para monetizar este proyecto consistiría en desplegar la infraestructura lo antes posible, para hacerse un sitio en el mercado e intentar conseguir la mayor cantidad de clientes posible. Para conseguirlo, valerse de suscripciones a diferentes rangos de precios y un modelo basado en el *pay-as-you-go* que ofrezca mucha flexibilidad a los contratantes. De esta manera, una vez conseguido ser una empresa que sea capaz de recolectar grandes cantidades de datos, adentrarse en el mercado de la venta de datos.

8

8. Gestión

Dado que este TFG está desarrollado en empresa la gestión se convierte en un componente central del proyecto.

Es fundamental llevar una gestión del alcance y el tiempo dedicado al proyecto, pero también es de vital importancia gestionar los costes y a los interesados.

En este capítulo se hace un análisis de las áreas más relevantes en lo que a la gestión refiere.

8.1. Gestión del alcance

En lo referido al alcance ha ocurrido un evento principal que ha implicado cambios. El retraso de la adquisición del gateway ha creado la necesidad de añadir nuevos paquetes de trabajo para conseguir poder testear el backend y avanzar en el proyecto mientras se esperaba la llegada del gateway.

El paquete de trabajo que más se ha visto resentido por estos imprevistos ha sido el de pruebas, puesto que se había decidido que era más importante tener un sistema totalmente funcional que hacer unas pruebas completas.

Por lo tanto, se decidió conseguir un sistema que cumpliese todos los objetivos y sustituir parte de las pruebas por una simulación de radioenlaces con el software Radio Mobile.

8.2. Gestión del tiempo

Este proyecto ha transcurrido entre los meses de mayo y julio, en la siguiente tabla se presenta la planificación temporal que se hizo al inicio del proyecto.

Semana	Dedicación	Fase	Tareas	Fecha
1	20	Inicio del proyecto		24 de Abril - 30 de Abril
2	25	Información	Documentarse sobre las tecnologías que se van a usar.	1 de Mayo - 7 de Mayo
3	20	Información		8 de Mayo - 14 de Mayo
4	25	Diseño	Diseñar el escenario que se va a implementar.	15 de Mayo - 21 de Mayo
5	25	Diseño		22 de Mayo - 28 de Mayo
6	20	Implementación	Tener el gateway y los sensores y empezar a configurarlos.	29 de Mayo - 4 de Junio
7	25	Implementación		5 de Junio - 11 de Junio
8	25	Implementación	Implementar el servidor de LoRa.	12 de Junio - 18 de Junio
9	25	Implementación		19 de Junio - 25 de Junio
10	25	Pruebas	Realizar pruebas de alcance	26 de Junio - 2 de Julio
11	25	Pruebas		3 de Julio - 9 de Julio
12	25	Terminar Documentación + Hacer presentación		10 de Julio - 16 de Julio
13	15	Preparar la defensa		17 de Julio - 23 de Julio
TOTAL	300			

Figura: 8.1. Planificación temporal inicial del proyecto.

Fuente: Creado por el autor.

Como se puede apreciar la llegada del gateway estaba prevista para finales del mes de mayo, pero este no llegó hasta mediados de junio, por lo tanto, se tomó la decisión de recortar en el apartado de pruebas y utilizar ese tiempo para investigar la posibilidad de usar un simulador para testear el backend.

8.3. Gestión de las dedicaciones

En este apartado se van a revisar las horas planeadas e invertidas para cada uno de los paquetes de trabajo principales.

PAQUETE DE TRABAJO	TAREA	DEDICACIÓN		
		Estimadas	Reales	Desviaciones
Desarrollo	Documentación	40h	35h	-5h
	Diseño	40h	25h	-15h
	Implementación	65h	95h	+30h
	Pruebas y Resultados	60h	30h	-30h
	Subtotal	205h	185h	-20h
ACADEMICO	Memoria	60h	80h	+20h
	Defensa	20h	20h	0h
	Subtotal	80h	100h	+20h
GESTIÓN	Planificación	6h	8h	+2h
	Seguimiento y Control	4h	4h	0h
	Reuniones	10h	12h	+2h
	Subtotal	20h	24h	+4h
TOTAL		305h	319h	+14h

Tabla 8.1. Gestión de dedicaciones.

8.4. Gestión de riesgos

Para prevenir posibles problemas se ha hecho la siguiente lista en la que se plantean cuáles pueden ser los problemas, la probabilidad de que ocurran, el impacto que tendrían en el proyecto y cómo solucionarlos.

8.4.1. No poder acabar a tiempo

DESCRIPCIÓN: No poder acabar el proyecto en las fechas fijadas.

EFECTO: Muy grave.

PROBABILIDAD: Baja.

SOLUCIÓN: Reducir la cantidad de objetivos a cumplir y si esto significase que el proyecto no tuviese la entidad suficiente para ser un trabajo de fin de grado posponer la fecha de entrega a la siguiente habilitada.

8.4.2. Perder documentación

DESCRIPCIÓN: Perder archivos del proyecto.

EFECTO: Muy grave.

PROBABILIDAD: Media.

SOLUCIÓN: A diario si se ha hecho alguna modificación en la estructura del proyecto se guardará en un servicio en la nube (Drive). Además, una vez a la semana se hará una copia en una memoria externa que se mantendrá en un espacio físico diferente al del portátil (Herramienta de trabajo principal):

8.4.3. Que no se disponga del hardware

DESCRIPCIÓN: Que por alguna razón no se puedan usar las herramientas hardware previstas.

EFECTO: Alto.

PROBABILIDAD: Media.

SOLUCIÓN: Se planteará hacer el proyecto sin usar hardware físico y se sustituirán las pruebas de alcance por otras líneas de investigación. Además, para probar el backend se explorará el uso de alguna herramienta que sea capaz de simular el trabajo de un gateway de LoRa.

8.5. Gestión de los interesados

Al tratarse de un proyecto desarrollado de forma paralela para la empresa GISA y la universidad es necesario identificar a los interesados:

Posición	Nombre
Responsable del proyecto	Alejandro Reyes Díez
Promotora del proyecto	Maider Likona Santamarina
Tutor del proyecto	Julián Alberto Lafuente Rojo

Tabla 8.2. Interesados del proyecto.

Fuente: Creado por el autor.

Además de esto, es necesario hacer un análisis del poder e interés de cada uno de ellos en los diferentes paquetes principales que componen el producto:

	Producto		Académico		Gestión	
	Interés	Poder	Interés	Poder	Interés	Poder
Responsable del proyecto	Alto	Alto	Alto	Alto	Alto	Alto
Promotora del proyecto	Alto	Alto	Bajo	Bajo	Alto	Alto
Tutor del proyecto	Medio	Bajo	Alto	Alto	Medio	Medio

Tabla 8.3. Interés y poder de los interesados en el proyecto.

Fuente: Creado por el autor.

9

9. Conclusiones y Trabajos futuros

En este apartado se muestra el análisis final que el autor ha ido extrayendo a lo largo de este proyecto. Del mismo modo, se revisan los objetivos planteados al inicio del proyecto y se pone de manifiesto cuales se han conseguido alcanzar.

Asimismo, se comentan caminos que al autor le hubiese gustado continuar, pero, debido a la limitación temporal no ha sido posible profundizar y se facilitan herramientas, pautas y recomendaciones para trabajos futuros que quieran hacer mejoras.

9.1. Conclusiones

Viendo la importancia que está adquiriendo el paradigma del IoT en este proyecto se ha planteado el desarrollo de una infraestructura de red para sensores y de un servicio web que permita almacenar los datos y accederlos en tiempo real.

Para ello se han usado dos tecnologías principales, por una parte, para comunicar los sensores se ha decidido usar LoRaWAN, una especificación para redes de bajo consumo energético y largo alcance de cobertura. Y por otra parte, se ha usado *Guifi.net,* una red de telecomunicaciones libre iniciada en Cataluña que ha llegado a convertirse en la red libre más extensa del mundo (David, 2016).

Utilizando estas herramientas, se ha conseguido superar el objetivo principal del proyecto: "*Desarrollar y documentar una prueba de concepto de una red de sensores conectada a Internet mediante redes comunitarias.*" Y se ha creado un servicio web que permite a los clientes acceder a los datos de los sensores a tiempo real.

En lo referido a los objetivos secundarios se han cumplido todos. Se ha hecho un análisis completo de las diferentes tecnologías con las que podría haberse desarrollado el proyecto y se han superado las contingencias que han aparecido, como, por ejemplo, la demora del gateway.

Esta demora ha creado conflicto en dos áreas: las pruebas de alcance y las pruebas del backend. No tener el gateway en un principio impedía testear el backend o hacer pruebas de alcance, dejando el proyecto paralizado. Viendo esto se decidió solucionar el problema. Por una parte, se automatizó un simulador que cumplía las especificaciones del protocolo que usaba el gateway para comunicarse con el servidor. De esta manera, usando el simulador, se pudo comprobar el funcionamiento del backend e ir avanzando en su desarrollo.

Por otra parte, para solucionar el problema con las pruebas de alcance, que a falta de hardware real no se podían hacer, se simularon los radioenlaces en Radio Mobile. Este software permitía introducir las características exactas de los gateways y sensores, junto con los datos de la zona geográfica en los que se iban a implantar, y devolvía el rendimiento que se podría esperar de ellos según la distancia.

Posteriormente el hardware llegó, por lo tanto, se complementaron las pruebas del backend con él y la simulación de los radioenlaces que se había hecho también se comprobó con el hardware real.

Una vez con el prototipo funcionando y testeado, al ser un prototipo para una empresa se decidió realizar una valoración económica en la que se proponen formas de explotación económica de la solución propuesta.

De esta forma cabe concluir que tanto LoRaWAN, debido a ser un estándar abierto, que posibilita la creación de redes de bajo consumo y largo alcance, como *guifi.net*, que posibilita la conexión a Internet en lugares en los que los medios tradicionales de acceso a Internet no llegan, son tecnologías idóneas para los objetivos que se planteaban en este proyecto.

En el transcurso de este proyecto se han aplicado conocimientos de la carrera. Particularmente conocimientos de redes y de servicios web, pero desde un prisma más general se han empleado muchos conocimientos de gestión de proyectos y nuevas aptitudes que el autor ha ido adquiriendo a lo largo de su carrera universitaria y que le han sido de gran utilidad a la hora de enfrentar los nuevos retos que le lanzaba este proyecto.

Tanto es así que en este proyecto pudiéndose haber elegido tecnologías ya conocidas se ha decidido apostar por tecnologías de vanguardia (InfluxDB y Grafana) para resolver el problema planteado, ya que, eran más apropiadas para este fin.

Otra de las enseñanzas valiosas que se ha extraído de este proyecto es el valor de una buena planificación y una buena gestión de riesgos. Gracias a ellas se han podido superar todos los problemas surgidos durante la fase de desarrollo y se han acabado cumpliendo la totalidad de los objetivos.

Es importante destacar que el resultado de este proyecto es un prototipo, pero es un prototipo muy cercano a una red real y hay pocos pasos desde este prototipo hasta una infraestructura de red, que permita convertir a San Sebastián en una *smart city* o que modernice sus industrias convirtiéndolas en la industria del futuro.

9.2. Trabajos futuros

Los trabajos futuros deberían seguir lo propuesto en este trabajo, mejorando los aspectos que menos se han trabajado por falta de tiempo, siempre teniendo en cuenta que este proyecto es una PoC y, por lo tanto, tiene que verse como lo que es, un prototipo. Hay que ser consciente de sus limitaciones y no pedirle las funcionalidades que se esperarían de un sistema comercial.

Por ello, a la hora de valorar los trabajos futuros existen dos caminos claros: la mejora de esta PoC y el despliegue de la red en un escenario de producción. Dentro del primer camino, una de las líneas que al autor más le hubiese gustado continuar es el relacionado con las pruebas de alcance.

En este documento se han detallado claramente los pasos a seguir a la hora de probar el alcance, pero las pruebas realizadas son insuficientes para extraer conclusiones relevantes. A pesar de que hay estudios que avalan las distancias que puede alcanzar LoRa en un escenario real (Reibot, 2017) (IOTpreneur, 2016), sería de alto valor llevar a cabo pruebas similares en el lugar en el que se va a implantar la infraestructura, usando diferentes hardwares y en distintas condiciones atmosféricas. Para, de esta manera, ser totalmente conscientes de las posibilidades y limitaciones a la hora de dar cobertura.

Dentro de este camino, otra forma de mejorar el prototipo consistiría en documentarse de buenas prácticas en seguridad a la hora de poner un servicio en producción, porque a pesar de que en este documento se han tratado cuestiones de seguridad, el aumento de los

ataques a los dispositivos IoT es un tema con el que hay que ser precavidos. Un dato que ilustra esto es el ofrecido en el *Global Threat Intelligence Report* (NTT Security, 2017) , que dice que los ataques de DDoS se han doblado de un 3% en 2016 a un 6% en 2017 debido a la falta de controles de seguridad en los dispositivos IoT.

Por último, una mejora que el autor considera muy necesaria es hacer pruebas de rendimiento al servicio. Para ello, se deberían realizar pruebas de estrés, y analizar el comportamiento de la solución.

Debería ser una prioridad ofrecer un servicio de alta fiabilidad a los clientes y para ello hay que crear un sistema escalable y redundante. Además, serían necesarias herramientas de monitorización para asegurar el correcto funcionamiento de los servicios.

El otro camino consiste en realizar el despliegue de la red en un entorno real. Esta tarea traerá nuevos retos, aun así, debería ser un paso natural una vez se haya conseguido un prototipo de más calidad, basándose en el trabajo logrado en este proyecto y las mejoras propuestas en este apartado.

Bibliografía

Roger Baig, Ramon Roca, Felix Freitag, Leandro Navarro: guifi.net, a crowdsourced network infrastructure held in common. Computer Networks 90: 150-165 (2015)Cambierr. (2016). *LoRaWAN Simulator*. Obtenido de GitHub: https://github.com/cambierr/LoRaWanSimulator.

Burgos, U., Gamecho, B., Gardeazabal, L., Gómez-Calzado, C., & Lafuente, A. (2016). Wireless Sensor Networks for Bird Tracking. Actas de las XXIV Jornadas de Concurrencia y Sistemas Distribuidos, 61-76. Granada, 15-17 junio 2016. ISBN: 978-84-16478-90-3.

Capgemini. (2014). *Monetizing the Internet of Things.* Obtenido de https://www.capgemini-consulting.com/resource-file-access/resource/pdf/iot_monetization_0.pdf

Casey, K. (4 de Agosto de 2015). *Network Computing*. Obtenido de http://www.networkcomputing.com/internet-things/10-leaders-internet-things-infrastructure/1612927605

Cisco. (Abril de 2011). *Cisco IBSG.* Obtenido de https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

CNXSoft. (21 de Septiembre de 2015). *Comparison Table of Low Power WAN Standards for Industrial Applications*. Obtenido de CNXSoft: https://www.cnx-software.com/2015/09/21/comparison-table-of-low-power-wan-standards-for-industrial-applications/

David. (7 de Junio de 2016). *Guifi.net una red de telecomunicaciones libre, abierta y neutral*. Obtenido de Cuéntame algo bueno: http://www.cuentamealgobueno.com/2016/06/guifi-net-una-red-de-telecomunicaciones-libre-abierta-y-neutral/

Deloitte Consulting LLP. (2016). *2016 Global Outsourcing Survey.* Obtenido de https://www2.deloitte.com/content/dam/Deloitte/nl/Documents/operations/deloitte-nl-s&o-global-outsourcing-survey.pdf

Egli, P. (2015). *Overview of emerging technologies for low power wide area networks in Internet and M2M scenarios.*

Gartner. (7 de Febrero de 2017). *Press Release*. Obtenido de http://www.gartner.com/newsroom/id/3598917

Gisbert, F. J., & Martí, I. C. (2010). *Rugosidad del terreno.* Obtenido de Fundación BBVA: https://w3.grupobbva.com/TLFU/dat/dt10_2010.pdf

Guifi.net. (20 de Enero de 2010). *Guifi.net*. Obtenido de https://guifi.net/es/ProcomunXOLN#ProcomunXOLN

Guifi.net. (2012). *Supernodo solar*. Obtenido de guifi.net: http://es.wiki.guifi.net/wiki/Archivo:Supernodo_solar.png

Guifi.net. (31 de Julio de 2017). *Estadísticas de nodos*. Obtenido de https://guifi.net/guifi/menu/stats/nodes

Human Rights Council. (27 de June de 2016). *The promotion, protection and enjoyment of human rights on.* Obtenido de https://www.article19.org/data/files/Internet_Statement_Adopted.pdf

IOTpreneur. (06 de Febrero de 2016). *LoRa / LoRaWan: pruebas de alcance con Multitech mDot (sx1272) y Conduit (sx1301)*. Obtenido de http://www.iotpreneur.com/lora-lorawan-pruebas-de-alcance-con-multitech-mdot-sx1272-y-conduit-sx1301/

Kloc, J. (2013). *P2PF Wiki*. Obtenido de http://wiki.p2pfoundation.net/Athens_Wireless_Metropolitan_Network

Kumbhar, S. (2 de Diciembre de 2015). *LoRa looks good to go*. Obtenido de IotNow: https://www.iot-now.com/2015/12/02/39616-lora-looks-good-to-go/

Lora Alliance. (2015). *A technical overview of LoRa® and LoRaWAN™.* Obtenido de https://docs.wixstatic.com/ugd/eccc1a_ed71ea1cd969417493c74e4a13c55685.pdf

LoRa Server. (s.f.). *LoRa Server documentation*. Obtenido de https://docs.loraserver.io

Metageek. (s.f.). *Understanding RSSI*. Obtenido de Metageek: http://www.metageek.com/training/resources/understanding-rssi.html

Microchip. (2015). *RN2483 LoRa Technology Module Command Reference User's Guide.*

Microchip. (2017). *Low-Power Long Range LoRa® Technology.* Obtenido de http://ww1.microchip.com/downloads/en/DeviceDoc/50002346C.pdf

Microchip. (s.f.). *868MHz RN2483 LoRa(TM) Technology Mote*. Obtenido de http://www.microchip.com/DevelopmentTools/ProductDetails.aspx?PartNO=dm164138&utm_source=&utm_medium=MicroSolutions&utm_term=&utm_content=DevTools&utm_campaign=RN2483+LoRa+Mote

Mikrotik. (s.f.). *Mikrotik*. Obtenido de https://mikrotik.com/product/RBSXT

Multitech. (2017). *MTAC-LoRa-H Antenna Specifications and Connector*. Obtenido de http://www.multitech.net/developer/products/multiconnect-conduit-platform/accessory-cards/mtac-lora/mtac-lora-h-antenna-specifications-and-connector/

nhimf. (2015). *Python script to transfer data from mqtt to InfluxDB*. Obtenido de GitHub: https://gist.github.com/nhimf/462c37ef78999c538e9f

NTT Security. (2017). *Global Threat Intelligence Report.* Obtenido de http://www.dimensiondata.com/Global/Downloadable%20Documents/2017%20Global%20Threat%20Intelligence%20Report%20as%20released%20by%20NTT%20Security.pdf

Organización de las Naciones Unidas. (1948). *Declaración Universal de los Derechos Humanos.* París.

Peñalver, E. J. (2017). *Análisis y diseño de una red de sensores en un parque natural.*

Pew Research Center. (2016). *The Modern News Consumer.* Obtenido de http://www.journalism.org/2016/07/07/the-modern-news-consumer/

PNUD. (2015). *Objetivos de Desarrollo Sostenible.* Obtenido de http://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/

Reibot. (23 de Abril de 2017). *Lora Range Test*. Obtenido de https://reibot.org/2017/04/23/lora-range-test/

Rendle, M. (2016). *On telemetry spy scandals*. Obtenido de Blog Rendle: https://blog.rendle.io/on-telemetry-spy-scandals/

Semtech Corporation. (2015). *LoRa™ Modulation Basics.* Obtenido de http://www.semtech.com/images/datasheet/an1200.22.pdf

Shore, J. (18 de Agosto de 2015). *Sell IoT data to generate revenue*. Obtenido de Internet of Things Agenda: http://internetofthingsagenda.techtarget.com/news/4500251939/Sell-IoT-data-to-generate-revenue

Sornin , N., Luis, M., Eirich, T., Kramp, T., & Hersent, O. (2015). *LoRaWAN™ Specification.* Obtenido de https://www.rs-online.com/designspark/rel-assets/ds-assets/uploads/knowledge-items/application-notes-for-the-internet-of-things/LoRaWAN%20Specification%201R0.pdf

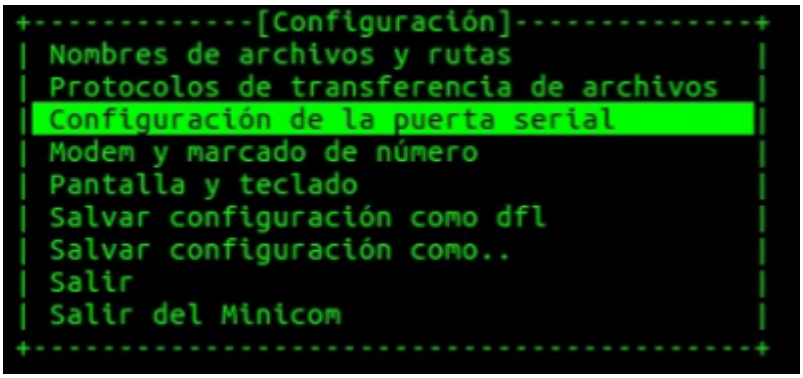
The Things Network. (2017). *LoRaWAN*. Obtenido de The Things Network: https://www.thethingsnetwork.org/wiki/LoRaWAN/Home

Anexo A: Configuración del sensor

En este apartado se explica la configuración básica del sensor descrito en el apartado **5.3. Error! Reference source not found.,** el RN2483 de Microchip.

Lo primero que se necesita es un programa para comunicarse con el sensor, en este caso se usará Minicom[16].

Una vez instalado hay que configurar la puerta serial.



Ventana principal de Minicom.

Fuente: Captura de pantalla realizada por el autor.

Los valores más importantes son los siguientes:

- Dispositivo Serial: Hay que seleccionar cuál de los dispositivos en el directorio */dev* se corresponde con el sensor.
- BPS: Este dato se corresponde a la tasa de baudios, en el caso del RN2483 es de 57600 bps.
- Paridad: No hay paridad en el RN2483.
- Bits: En este apartado se describe la estructura de los mensajes, que es de 8 bits de datos y un bit de parada.

[16] https://linux.die.net/man/1/minicom

Configuración de la puerta serial.

Fuente: Captura de pantalla realizada por el autor.

Después, es importante activar el *echo (ctrl+a -> ctrl+e)*, de esta manera se puede ver los comandos que se están escribiendo.

En este punto ya está configurado para que se puedan escribir comandos, es de vital importancia recordar que después de cada uno de los comandos que se quieran enviar hay que pulsar Enter seguido de Ctrl+j que es el comando que hace que se envié un CR LF en Minicom.

Es ahora cuando podemos ejecutar el comando:

```
Sys factoryRESET
```

Para asegurar que tiene una configuración limpia, posteriormente, se introducen todos los comandos para configurar los parámetros de red.

```
mac set nwkskey INTRODUCIR_AQUÍ_TU_CLAVE_DE_SESIÓN_DE_RED
mac set appskey INTRODUCIR_AQUÍ_TU_CLAVE_DE_SESIÓN_DE_APLICACIÓN
mac set devaddr INTRODUCIR_AQUÍ_TU_DEVADDR
(opcional)mac set adr on
(opcional)mac set pwr 14
mac save
mac join abp
```

En este caso se introducen las claves de sesión junto con la dirección del sensor (*devaddr*). Opcionalmente se puede activar el ADR que si la red lo permite optimiza la frecuencia de envío de datos y el uso de la energía. Otro de los parámetros opcionales es el pwr, que representa la potencia de envío de la señal, en el caso del RN2483 el máximo es 14 dBm (Microchip, 2015) así que se puede seleccionar esa.

En este punto ya es posible mandar paquetes LoRaWAN mediante el siguiente comando:

```
mac tx uncnf 1 A
```

Si se ha hecho todo correctamente el programa devuelve la siguiente respuesta:

```
ok
mac_tx_ok
```

Anexo B: Configuración del gateway

A la hora de configurar el gateway de Multitech es importante saber que por defecto funciona como un servidor de LoRA, es decir, puede recibir paquetes y publicarlos por MQTT, pero es un servidor poco potente, otra de las opciones que existen es quitarle las funciones de servidor y hacer que funcione de packet forwarder y envíe paquetes a un backend más potente como el que se explicará en el siguiente anexo.

Configurar el gateway como servidor

La configuración básica para conectarse al gateway y asignarle valores como la fecha o la dirección IP está en el siguiente link:

http://www.multitech.net/developer/software/mlinux/getting-started-with-conduit-mlinux/

Posteriormente, se debería instalar la última versión disponible de mLinux como se explica en la siguiente web:

http://www.multitech.net/developer/software/mlinux/using-mlinux/flashing-mlinux-firmware-for-conduit/

Y por último sería necesario poner al día las versiones del servidor de red y el packet forwarder, esto está explicado en el siguiente link:

http://www.multitech.net/developer/software/mlinux/using-mlinux/upgrade-lora-server/

En este proyecto las versiones que se han usado de mlinux y de los distintos paquetes que se han instalado son las siguientes:

```
root@mtcdt:~# opkg list | grep lora
lora-gateway-utils - 4.0.1-r9.0
lora-network-server - 1.0.36-r1.0
lora-packet-forwarder - 3.0.0-r9.0
lora-packet-forwarder-usb - 1.4.1-r10.0
lora-query - 1.0.2-r1.0
```

Versión de mLinux y de los paquetes instalados en el gateway.

Fuente: Captura de pantalla realizada por el autor.

El siguiente paso es editar la configuración de lora, para ello hay que crear la carpeta de lora y añadirle la configuración, en el caso de este proyecto es la siguiente:

```
mkdir /var/config/lora
```

```
nano /var/config/lora/lora-network-server.conf
```

Este archivo que se ha creado lleva la configuración del servidor de red, los parámetros que se pueden editar están especificados en el siguiente link:

http://www.multitech.net/developer/software/lora/conduit-mlinux-lora-communication/conduit-mlinux-advance-lora-configuration/

Y la configuración exacta que se usó para este anexo es la siguiente:

```
{
        "lora": {
                "netID": "010203",      /* netID for beacon packets */
                "frequencyBand": "868", /* "915" or "868" */
                "channelPlan": "EU868", /* AU915, US915, EU868, AS923 or KR920 */
                "frequencySubBand": 7,  /* Sub-band for US operation, 1-8 */
                "rx1DatarateOffset": 0, /* Datarate offset for mote rx window 1 sent in join respons$
                "rx2Datarate": 8,               /* Datarate for mote rx window 2 sent in join respon$
                "maxTxPower": 14,       /* Max Tx power (dBm), -6 to 26 */
                "frequencyEU": 867500000, /* center freq for extra EU channels (Hz) */
                "frequencyAS": 922600000, /* center freq for extra AS channels (Hz) */
                "frequencyKR": 922900000  /* center freq for extra KR channels (Hz) */
        },
        "udp": {
                "appPortUp": 1784, /* port for user-developed application use */
                "appPortDown": 1786 /* port for user-developed application use */
        },
        "addressRange": {
                "start": "00:00:00:01", /* address range used for mDots */
                "end": "FF:FF:FF:FE"
        },
        "network": {
                "public": true,    /* set to false for private LoRa network with mDots + Conduit */
                "leasetime": 0,  /* time until mDot join expires (minutes) or 0 for no expiration */
                "eui": "aaaaaaaaaaaaaaaa",
                "key": "aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa"
        },
        "log" : {
                "console" : true,
                "syslog" : false,
                "level" : 100, /* error=10, warn=20, info=30, debug=50, trace=60, max=100 */
                "path": "/var/log/lora-network-server.log"
        },
        "mqtt": {
                "enabled": true
        }
}
```

Archivo de configuración del servidor de red de Lora.

Fuente: Captura de pantalla realizada por el autor.

Lo más importante es que se han usado las frecuencias y canales europeos, y que en el apartado *network*, el valor *public* se ha puesto a *true*, esto es muy importante ya que, al ser el sensor de otra marca que no es Multitech si no se pone pública la red no funcionaría.

Quedaría añadir el nodo mediante ABP, mediante el siguiente comando:

```
Lora-query -a [DevAddr] [APPEUI] [DEVEUI] [NwkSKey] [AppSKey]
```

```
root@mtcdt:~# lora-query -a 00000000 C 0123456789ABCDEF 0004A30B001A7E42 00000000000000000000000000000000 00000000000000000000000000000000
root@mtcdt:~# lora-query -n
Net Addr     Dev EUI                 Class  Joined               Seq Num      Up     Down      1st      2nd  Dropped  RSSI min   max   avg  SNR min   max   avg
00:00:00:00  00-04-a3-0b-00-1a-7e-42  C      2017-07-07T15:51:46Z        0       0        0        0        0        0         0     0     0        0     0     0
```

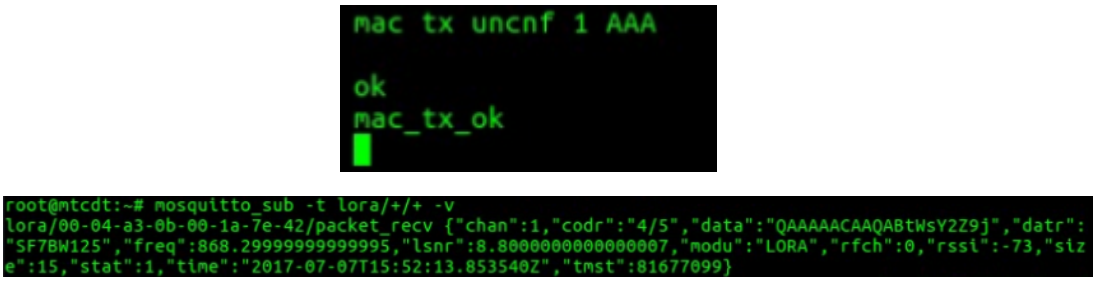
Añadiendo un nodo al servidor de red del gateway.

Fuente: Captura de pantalla realizada por el autor.

En este punto ya se puede iniciar el servidor y poner a funcionar el cliente MQTT:

```
/etc/init.d/lora-network-server restart
mosquitto_sub -t lora/+/+ -v
```

Ya se puede ver como con el comando de envío de paquetes LoRa que se ha explicado en el anexo anterior se pueden mandar mensajes y se reciben correctamente en el gateway.



Envío y recepción de un paquete de LoRa, usando la gateway como servidor.

Fuente: Captura de pantalla realizada por el autor.

Configurar el gateway como packet forwarder

Esta es la parte más interesante ya que se deja que el gateway simplemente haga forward de los paquetes y estos ya se tratan en un backend más potente.

Para implementar esta solución hay que seguir los siguientes pasos:

- Parar el servidor de red del gateway, ya que se va a usar otro diferente:

```
/etc/init.d/lora-network-server stop
```

- Editar la configuración inicial para que deshabilite el servidor del gateway, en el apartado *ENABLED* poner *"no":*

```
nano /etc/default/lora-network-server
```

- Editar la configuración inicial para que habilite el packet forwarder, en el apartado *ENABLED* poner *"yes":*

```
nano /etc/default/lora-packet-forwarder
```

- Editar el archivo *global_conf.json* para que concuerde con los parámetros del nuevo backend:

```
"SX1301_conf": {
    "radio_0": {
        "enable": true,
        "freq": 867500000
    },
    "radio_1": {
        "enable": true,
        "freq": 868500000
    },
    "chan_multiSF_0": {
        "enable": true,
        "radio": 1,
        "if": -400000
    },
    "chan_multiSF_1": {
        "enable": true,
        "radio": 1,
        "if": -200000
    },
    "chan_multiSF_2": {
        "enable": true,
        "radio": 1,
        "if": 0
    },
    "chan_multiSF_3": {
        "enable": true,
        "radio": 0,
        "if": -400000
    },
    "chan_multiSF_4": {
        "enable": true,
        "radio": 0,
        "if": -200000
    },
    "chan_multiSF_5": {
        "enable": true,
        "radio": 0,
        "if": 0
    },
    "chan_multiSF_6": {
        "enable": true,
        "radio": 0,
        "if": 200000
    },
    "chan_multiSF_7": {
        "enable": true,
        "radio": 0,
        "if": 400000
    },
    "chan_Lora_std": {
        "enable": true,
        "radio": 1,
        "if": -200000,
        "bandwidth": 250000,
        "spread_factor": 7
    },
```

```
        "chan_FSK": {

        "enable": true,
        "radio": 1,
        "if": 300000,
        "datarate": 50000,
        "freq_deviation": 25000
    }
},
"gateway_conf": {
    "synch_word": 52,
    "forward_crc_disabled": false,
    "forward_crc_error": true,
    "forward_crc_valid": true,
    "gateway_ID" : "00800000A000062A"
    "keepalive_interval": 120,
    "push_timeout_ms": 120,
    "serv_port_down": 1700,
    "serv_port_up": 1700,
    "server_address": "192.168.2.2",
    "stat_interval": 20
}
```

Archivo global_conf.json

Este es el archivo base al que se le han añadido varios detalles, se ha puesto la IP del servidor “192.168.2.2”, el puerto que usa este servidor “1700” y el ID del gateway, que se ha conseguido mediante el siguiente comando:

```
root@mtcdt:~# mts-io-sysfs show lora/eui
00:80:00:00:A0:00:06:2A
```

Comando para conseguir el ID del gateway.

Fuente: Captura de pantalla realizada por el autor.

- En este punto ya se puede simplemente ejecutar el packet forwarder y ya estaría listo el gateway enviando paquetes al nuevo backend.

```
/etc/init.d/lora-packet-forwarder start
```

Anexo C: Configuración del backend

Ahora queda configurar la solución descrita en este trabajo, para ello lo primero que hay que hacer es instalar el LoRa Server.

Para ello el autor propone varios métodos, el que se ha usado en este proyecto es uno que utiliza Vagrant[17]. Vagrant es una herramienta que facilita la creación y configuración de máquinas virtualizadas.

Lo primero que hay que hacer es descargar las últimas versiones tanto de Vagrant como de VirtualBox[18], y comprobar que el PC que se vaya a usar tenga habilitada la virtualización.

Una vez hecho esto se descarga del siguiente link todos los archivos necesarios para levantar el servicio con Vagrant:

https://github.com/brocaar/loraserver-setup

Una vez estén todos los archivos localizados en una carpeta, hay que acceder al archivo *host_vars/vagrant.yml* y actualizar el valor de *BAND* a EU_433 que es el que corresponde a Europa.

Por otra parte, antes de levantar la máquina hay que editar el archivo *Vagrantfile* y añadir las siguientes líneas:

```
box.vm.network "forwarded_port", guest: 3000, host: 3000, protocol: "tcp"
box.vm.network "forwarded_port", guest: 8086, host: 8086, protocol: "tcp"
```

Comandos añadidos en Vagrantfile.

Fuente: Captura de pantalla realizada por el autor.

Estos comandos hacen forwarding de los puertos 3000 y 8086, que corresponden a Grafana y a InfluxDB respectivamente. Esto es necesario si se quiere acceder a la interfaz web de estos programas desde el PC que ha levantado la instancia de Vagrant.

En este punto, podemos ejecutar los siguientes comandos para levantar la máquina virtual y hacer una conexión ssh:

```
sudo vagrant up
```

[17] https://www.vagrantup.com/
[18] https://www.virtualbox.org/

```
sudo vagrant ssh
```

Una vez conectados a la máquina virtual procedemos a instalar tanto InfluxDB como Grafana, los tutoriales seguidos para realizar esta instalación son los siguientes:

- Para instalar Grafana: http://docs.grafana.org/installation/debian/
- Para instalar InfluxDB: https://docs.influxdata.com/influxdb/v0.9/introduction/installation/

El siguiente paso es editar el firewall para abrir los puertos que usan estas dos aplicaciones y hacer los cambios permanentes, esto se hace mediante los siguientes comandos:

```
sudo iptables –A INPUT –p tcp –dport 3000 –j ACCEPT
sudo iptables –A INPUT –p tcp –dport 8086 –j ACCEPT
sudo iptables-save > /etc/iptables.rules.v4
```

Ahora, con todos los softwares instalados es necesario conectarlos entre ellos, son dos las conexiones que hay que hacer, conectar el broker MQTT con la base de datos para que escriba directamente ahí y la segunda es hacer la conexión entre InfluxDB y Grafana para que se hagan gráficos de los datos a tiempo real.

Para escribir del broker MQTT a InfluxDB se ha usado un script en Python sirviéndose tanto de la librería paho.mqtt.client[19] como de la ofrecida por InfluxDB (nhimf, 2015).

El script que se ha usado es el siguiente:

[19] https://pypi.python.org/pypi/paho-mqtt/1.1

```
#!/usr/bin/env python

import paho.mqtt.client as mqtt
from influxdb import InfluxDBClient
import datetime
import json

def on_connect(client, userdata, flags, rc):
    print("Connected with result code "+str(rc))
    client.subscribe("#")

def on_message(client, userdata, msg):
    current_time = datetime.datetime.now().strftime("%Y-%m-%d
%H:%M:%S")
    d= json.loads(str(msg.payload))
    print d['rxInfo']['size']
    json_body = [
    {
        "measurement": "Size",
        "tags": {
            "host": "sensor",
        },
        "fields": {
            "value": d['rxInfo']['size']
        }
    }
    ]
influx_client.write_points(json_body)

print(msg.topic+"*****************************"+str(msg.payload
))

influx_client = InfluxDBClient('localhost', 8086,
database='datos')
client = mqtt.Client()
client.on_connect = on_connect
client.on_message = on_message

client.connect("127.0.0.1", 1883, 60)

client.loop_forever()
```

InfluxToMQTT.py

En este script se hace lo siguiente, cuando se conecte se suscribe al topic # y cada vez que llega un mensaje se carga en un JSON y se almacena en la base de datos llamada *datos*.

Es un ejemplo simple de como cargar todos los mensajes enviados por los gateways a una base de datos. Cuando el proyecto se ponga en producción este script se debería personalizar teniendo en cuenta de que gateway o de qué sensor se reciben los datos, y teniendo en cuenta esto se podrían almacenar los datos en distintas bases de datos manteniendo así la información más organizada.

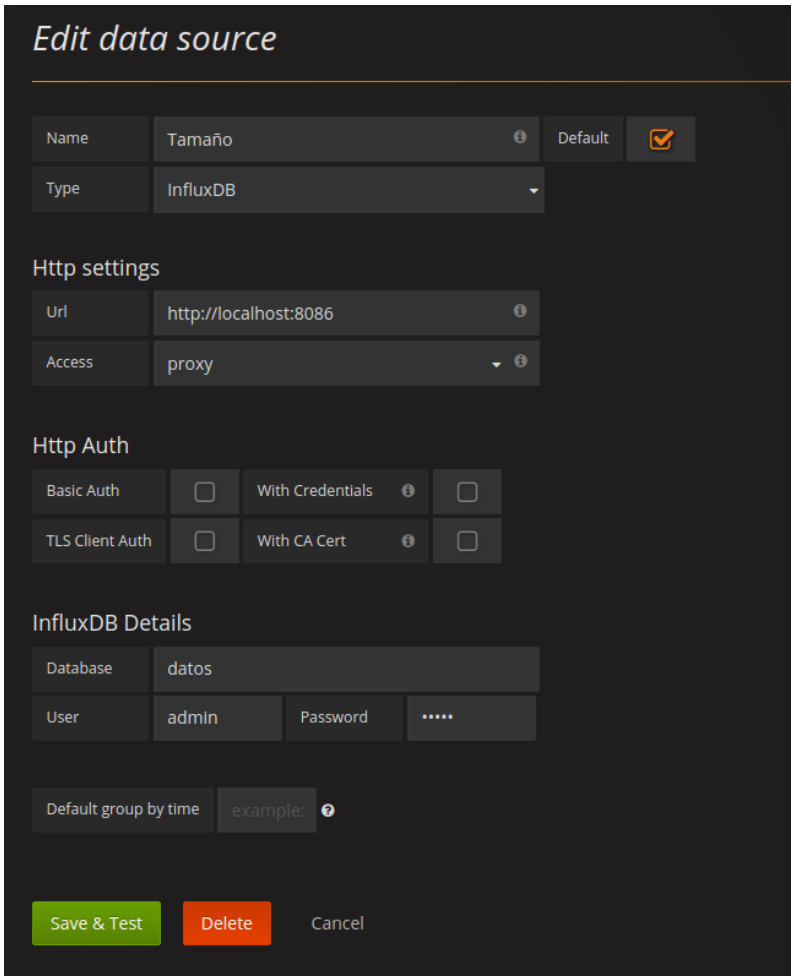
Se puede hacer que este script se ejecute como un servicio en segundo plano cada vez que se levante la máquina virtual, una de las formas para conseguirlo es editando el archivo */etc/rc.local* y añadiendo una línea llamando al script.

Ahora que ya se ha hecho la conexión entre el broker e InfluxDB hay que conectar la base de datos con Grafana para poder ver la información a tiempo real y en gráficos.

Para ello hay que conectarse al puerto de Grafana desde el navegador:

```
localhost:3000
```

Y acceder a la ventana de *Data Sources*, donde es necesario añadir una nueva fuente de datos con los siguientes parámetros:



Configuración de la fuente de datos para Grafana.

Fuente: Captura de pantalla realizada por el autor.

En esta ventana simplemente se le da un nombre a la conexión, se especifica la IP y el puerto en el que está InfluxDB y se concreta cual es la base de datos de la que se va a recoger la información.

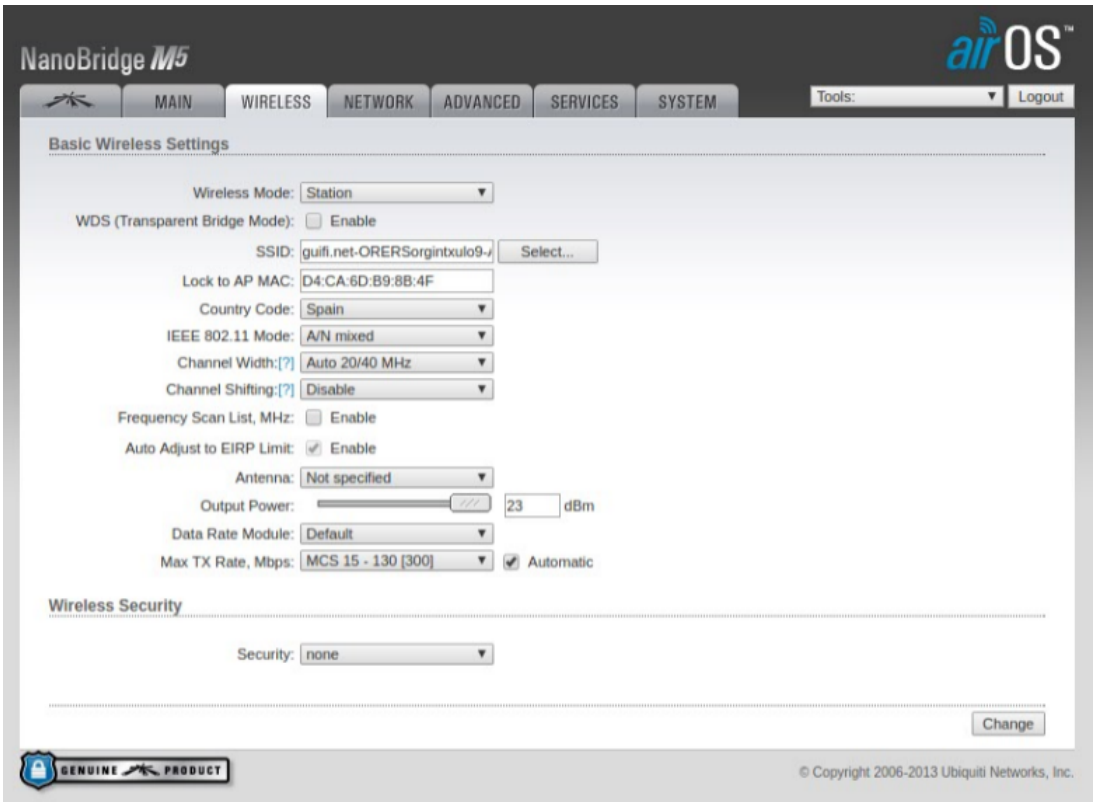
Ahora ya se pueden crear los diferentes *dashboards* para mostrar los datos que se desee. Como se pueden hacer las consultas y los diferentes gráficos queda explicado en el siguiente link:

http://docs.grafana.org/features/datasources/influxdb/

Anexo D: Conectarse a Guifi.net

En este apartado se va a hacer una breve descripción de cómo conectar una antena, en este caso una NanoBridge M5[20], a un SuperNodo de Guifi.net.

Para ello el primer paso consiste en conectarse a la IP de la antena, después de hacer esto se presentará la interfaz para hacer la conexión, aquí hay que dirigirse a la pestaña *Wireless* y seleccionar el SSID de la antena a la que se quiera conectarse.



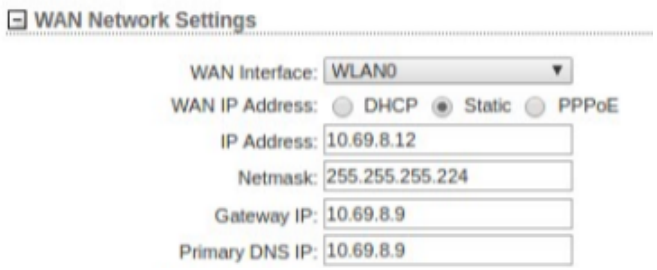
Selección del SSID.

Fuente: Captura de pantalla realizada por el autor.

[20] https://www.ubnt.com/airmax/nanobridgem/

Una vez se realice la conexión hay que hacer la configuración de los parámetros de la red, los parámetros principales que hay que configurar son los siguientes:

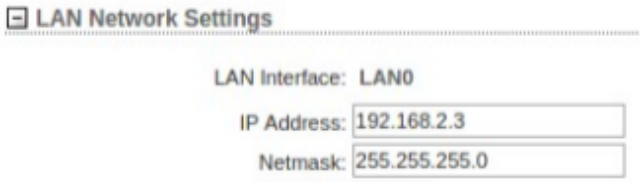
- Network mode: Se usará el modo router para que sea la antena la encargada de las funciones de router (asignar IPs, redireccionar paquetes…)
- WAN IP Address, Netmask y Gateway IP: Estos 3 datos los ofrece la página web de Guifi.net al registrar un nodo.



Configuración WAN.

Fuente: Captura de pantalla realizada por el autor.

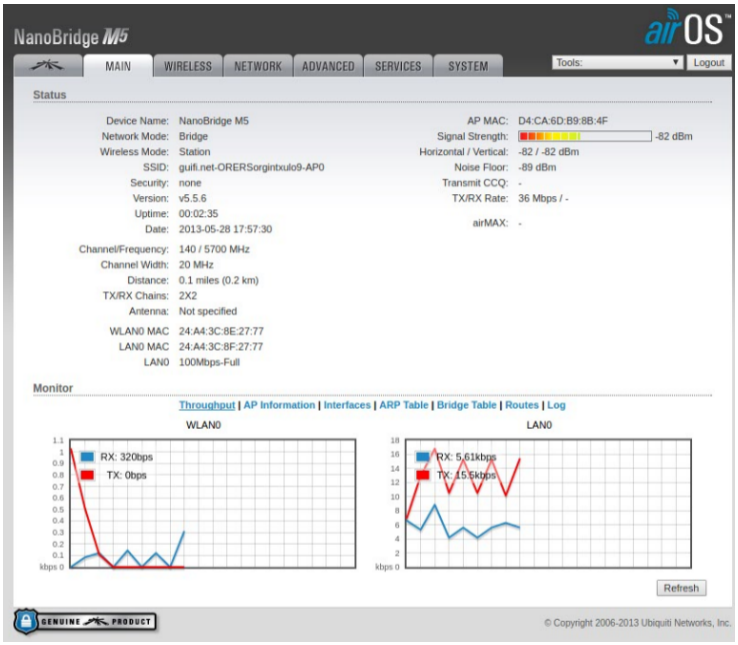
- LAN IP Address y Netmask: Aquí se configuran las propiedades de la red local.



Configuración LAN.

Fuente: Captura de pantalla realizada por el autor.

Si se ha hecho la conexión correctamente se podrá ver las características generales de la conexión en la ventana principal.



Características principales de la conexión.

Fuente: Captura de pantalla realizada por el autor.