



Diseño e implementación de una herramienta para la verificación de cobertura de la red SIGFOX. Estudio de conectividad en una zona geográfica de orografía compleja.

Daniel Cárdenes Tacoronte

Máster Universitario en Ingeniería de Telecomunicación
Sistemas de Comunicación

Raúl Parada Medina

Carlos Monzo Sánchez

9 de junio de 2016



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

*A Maymesseri,
por todos estos años de incondicional apoyo.
Gracias por compartir el camino.*

FICHA DEL TRABAJO FINAL

Título del trabajo:	Diseño e implementación de una herramienta para la verificación de cobertura de la red SIGFOX. Estudio de conectividad en una zona geográfica de orografía compleja
Nombre del autor:	Daniel Cárdenes Tacoronte
Nombre del consultor:	Raúl Parada Medina
Nombre del PRA:	Carlos Monzo Sánchez
Fecha de entrega (mm/aaaa):	06/2016
Titulación:	Máster Universitario en Ingeniería de Telecomunicación
Área del Trabajo Final:	Sistemas de Comunicación
Idioma del trabajo:	Español
Palabras clave:	SIGFOX, cobertura, medidas de campo
Resumen del Trabajo (máximo 250 palabras):	
<p>Entre los diferentes actores que están permitiendo que el Internet de las cosas sea finalmente una realidad, los operadores de redes inalámbricas de largo alcance y bajo consumo, especializados en proveer servicios de conexión a millones de dispositivos, están jugando un papel fundamental.</p> <p>Por otro lado, la capacidad de proporcionar ubicuidad a todos esos dispositivos está supeditada a los niveles de cobertura que dichos operadores sean capaces de dar, siendo necesaria, en la mayoría de los casos, la comprobación previa de la disponibilidad real del servicio en aquellas zonas de interés donde se haya proyectado hacer el despliegue.</p> <p>Es por ello que resulta de gran utilidad disponer de una herramienta que permita realizar dicha verificación de forma sencilla y que pueda servir de apoyo a la optimización de los diseños finales que pretendan usar las redes de esos operadores.</p> <p>El presente trabajo muestra el desarrollo e implementación de una herramienta que permite comprobar el nivel de cobertura a nivel de servicio de la red SIGFOX, el primer operador de red orientado específicamente al Internet de las cosas. Esta herramienta, basada en una plataforma de desarrollo de prototipos de bajo coste, permite además realizar dicha verificación usando diferentes patrones de transmisión, facilitando así la tarea de simular distintos modelos de sensores.</p> <p>El trabajo también recoge los resultados de una campaña de medidas realizada en la isla de Gran Canaria, donde se pone en práctica el desarrollo realizado.</p>	

Abstract (in English, 250 words or less):

Among different players that are making the Internet of Things to be a fact, Low-Power Wide-Area network operators, specialized in providing connectivity services to millions of devices, are playing a prime role.

On the other hand, the capability of providing ubiquity to all of those devices is conditioned to the coverage level that the operators are able to offer, being necessary, in most cases, the previous verification of the real service coverage around the places of interest where the final solution is expected to be deployed.

For all of that, it would be helpful if there were a tool that permitted to perform such verification in an easy way, assisting to the final design optimizations that plan to use those operator networks.

This work shows the development and implementation of a tool for testing the service level coverage of SIGFOX network, the first operator focused on the Internet of Things. This tool, based on a low-cost prototyping platform, can perform this verification using different transmission patterns, thus making it simple the task of simulating different sensor models.

A coverage field trial over Gran Canaria Island is also presented, where the implementation was tested in a real scenario.

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo	1
1.2 Objetivos del Trabajo	2
1.3 Enfoque y método seguido	3
1.4 Planificación del Trabajo.....	4
1.5 Breve resumen de productos obtenidos	7
1.6 Breve descripción de los otros capítulos de la memoria.....	8
2. Estado del Arte.....	9
2.1 Redes LPWAN	9
2.2 Medidas de cobertura sobre redes LPWAN	11
2.3 Herramientas para la verificación del servicio de la red SIGFOX.....	13
2.4 Contribuciones de este trabajo	15
3. Desarrollo del prototipo	17
3.1 Arquitectura general del sistema	17
3.2 Diagrama de bloques del SCAN-device	18
3.3 Tarjeta controladora	21
3.4 Diseño e integración del teclado.....	22
3.5 Integración del módulo GPS	25
3.6 Integración del módulo de comunicaciones SIGFOX	28
3.7 Formato del mensaje transmitido.....	30
3.8 Mapa de la memoria EEPROM.....	32
3.9 Comandos externos.....	34
3.10 Integración mecánica.....	37
3.11 Pruebas funcionales	38
4. Desarrollo de la aplicación	44
4.1 Entorno de desarrollo.....	44
4.2 Diagrama conceptual y formato de ficheros	45
4.3 Importar datos del SCAN-device	48
4.4 Importar datos del SCAN-server.....	50
4.5 Procesado de datos	52
4.6 Presentación de resultados en el mapa.....	54
5. Medidas de Campo	57
5.1 Criterios y planificación de las localizaciones	57
5.2 Análisis y publicación de resultados	61
5.3 Cobertura real medida vs cobertura simulada	68
6. Conclusiones.....	72
7. Glosario	73
8. Bibliografía	74
9. Anexos	76
A.1 Repositorio del código fuente	76
A.2 Manual de usuario del prototipo SCAN-device.....	78
A.3 Manual de usuario de la aplicación SCAN-tool	81
A.4 Coste de componentes.....	83
A.5 Esquema de conexiones	84

Lista de Figuras

Figura 1. Diagrama de Gantt de la planificación	6
Figura 2. Mapa de calor representando el nivel de señal recibido desde diferentes posiciones en Oulu, Finlandia, centrado en la posición de la estación base (Ptx=14 dBm, Grx=2 dBi, R= 293 bps, htx= 2m, hrx=24m) [10].....	12
Figura 3. Demostrador SIGFOX.....	14
Figura 4. Arquitectura propuesta. Se presenta la secuencia típica, desde que se genera un mensaje en el prototipo, hasta que se obtienen los resultados del análisis de cobertura.	18
Figura 5. Placa SmartEverything.....	19
Figura 6. Módulos independientes.	19
Figura 7. Diagrama de bloques del prototipo hardware desarrollado (SCAN-device). Se muestran los diferentes módulos que lo componen, así como su interfaz con la placa controladora.....	20
Figura 8. Detalle de la integración final del teclado. Se representan tanto los elementos que lo forman (LEDs y pulsadores) como la distribución del resto de módulos que forman el prototipo.....	25
Figura 9. Ejemplo de tramas NMEA enviadas por el GPS. En este caso, el GPS ya ha calculado posición válida.....	27
Figura 10. Detalle de las placas que forman el prototipo y del microswitch de configuración del puerto serie en la placa Xbee Shield.	30
Figura 11. Respuesta del SCAN-device a un comando \$GP\$. La respuesta presenta el TXID actual (231), el número de registros de transmisión almacenados en la EEPROM (55) y los valores de potencia asignados a los LEDs Hig, Med y Low (14,7,0) en dBm.	35
Figura 12. Respuesta del SCAN-device a un comando \$GD\$. La respuesta presenta cada uno de los registros almacenados en una línea.....	35
Figura 13. Respuesta del SCAN-device a un comando \$HL:12\$. Inmediatamente después de ejecutarlo, se interroga con un \$GP\$ para verificar que efectivamente el valor del LED Tx Hig ha cambiado al valor 12 dBm.....	36
Figura 14. Respuesta del SCAN-device a un comando \$ML:05\$. Inmediatamente después de ejecutarlo, se interroga con un \$GP\$ para verificar que efectivamente el valor del LED Tx Med ha cambiado al valor 5 dBm.	36
Figura 15. Respuesta del SCAN-device a un comando \$LL:01\$. Inmediatamente después de ejecutarlo, se interroga con un \$GP\$ para verificar que efectivamente el valor del LED Tx Low ha cambiado al valor 1 dBm.....	37
Figura 16. Prototipo final ensamblado.....	38
Figura 17. Configuración de la sección “Downlink data”	39
Figura 18. Configuración de la sección “Display type”	39
Figura 19. Configuración de la sección “Callbacks”	40
Figura 20. Mensaje a 14 dBm recibido correctamente en el backend.	41
Figura 21. Mensaje a 7 dBm recibido correctamente en el backend.	41
Figura 22. Mensaje a 7 dBm recibido correctamente en el backend.	42
Figura 23. Transmisión con ACK. El backend.....	42
Figura 24. Transmisión con ACK. Se fuerza un error en el backend	42
Figura 26. IDE del entorno de desarrollo QT desde donde se desarrolló	45
Figura 27. Diagrama conceptual de la aplicación SCAN-tool. Se presentan los diferentes módulos que la componen así como el formato de ficheros de entrada/salida de cada módulo.	46
Figura 28. Ventana principal de la herramienta SCAN-tool	48

Figura 29. Pestaña del módulo Import-device. Permite importar datos del SCAN-device, así como realizar otras tareas relacionadas con la gestión de ficheros.	48
Figura 30. Diálogo que permite conectarse al SCAN-device a través del puerto serie para la descarga de datos.	49
Figura 31. Formato JSON de salida de un fichero que ha sido importado del SCAN-device.	50
Figura 32. Pestaña del módulo Import-server. Permite importar datos del SCAN-server, así como realizar otras tareas relacionadas con la gestión de ficheros.	51
Figura 33. Diálogo que permite descargar los datos del SCAN-server.	51
Figura 34. Ejemplo de formato JSON de salida de un fichero que ha sido importado del SCAN-server.	52
Figura 35. Contenido parcial de un fichero GeoJSON obtenido al procesar varios ficheros JSON de entrada.	53
Figura 36. Pestaña del módulo Process-data.	54
Figura 37. Ventana principal de presentación del Mapa y resultados.	55
Figura 38. Muestra de algunas de las estructuras topográficas recorridas durante la campaña de medidas de campo en la isla de Gran Canaria.	58
Figura 39. Detalle de las rutas planificadas durante la campaña de medidas. Día 1 en rojo; Día 2 en blanco y día 3 en amarillo. Total 764 Km recorridos.	59
Figura 40. Configuración del panel de medida instalado en la motocicleta.	60
Figura 41. Mapa de nivel de servicio correspondiente a la primera jornada de medidas. Se incluyen también las medidas tomadas en la ciudad de Las Palmas de G.C. durante las pruebas funcionales del prototipo.	62
Figura 42. Detalle de una transmisión que ha sido recibida por 13 estaciones base. El nivel de redundancia en ese punto se puede considerar excelente.	62
Figura 43. Detalle de una transmisión que ha sido recibida por 2 estaciones base.	63
Figura 44. Detalle de una transmisión que ha sido recibida por 1 estación base. En este caso, no existe redundancia en el número de estaciones base que cubren ese punto.	63
Figura 45. Detalle de una transmisión que no ha sido recibida por ninguna estación base. En ese punto se puede considerar que no hay servicio de la red SIGFOX.	64
Figura 46. Mapa de nivel de servicio correspondiente a la segunda jornada de medidas.	65
Figura 47. Mapa de nivel de servicio correspondiente a la tercera jornada de medidas.	66
Figura 48. Mapa global representando el nivel de servicio obtenido una vez concluida la campaña de medidas de campo.	67
Figura 49. Mapa proporcionado por el operador, con la simulación de la cobertura de servicio de la red SIGFOX para la isla de Gran Canaria.	69
Figura 50. Mapa de simulación de cobertura con información de los puntos de servicio obtenidos en la campaña de medidas.	70

Lista de Tablas

Tabla 1. <i>Planificación de tareas</i>	5
Tabla 2. <i>Lista y descripción de los diferentes hitos de la planificación</i>	7
Tabla 3. <i>Comparativa de diferentes tecnologías radio LPWAN [7]</i>	10
Tabla 4. <i>Comparativa entre el demostrador SIGFOX y la herramienta desarrollada en este proyecto</i>	16
Tabla 5. <i>Relación de requisitos básicos de diseño necesarios y capacidades de la placa Arduino Uno</i>	22
Tabla 6. <i>Relación de LEDs y pulsadores empleados en el diseño del teclado y pines de la placa controladora a la que han sido conectados</i>	23
Tabla 7. <i>Características principales del receptor GPS NEO-6M [20]</i>	26
Tabla 8. <i>Detalle de los diferentes campos que forman una trama del tipo \$GPGLL</i>	27
Tabla 9. <i>Características principales del módulo de comunicación SIGFOX empleado en el prototipo</i>	29
Tabla 10. <i>Formato del mensaje transmitido por el prototipo a la red SIGFOX</i>	32
Tabla 11. <i>Mapa de direcciones de memoria de la EEPROM del SCAN-device</i>	34
Tabla 12. <i>Consumo del prototipo medido para diferentes escenarios</i>	43
Tabla 13. <i>Resultado de las estadísticas calculadas por el SCAN-tool para la primera jornada de la campaña de medidas</i>	64
Tabla 14. <i>Resultado de las estadísticas calculadas por el SCAN-tool para la segunda jornada de la campaña de medidas</i>	66
Tabla 15. <i>Resultado de las estadísticas calculadas por el SCAN-tool para la tercera jornada de la campaña de medidas</i>	67
Tabla 16. <i>Estadísticas globales de la campaña de medidas de campo</i>	68
Tabla 17. <i>Cuantificación de coincidencias entre la simulación y las medidas reales obtenidas</i>	71

1. Introducción

1.1 Contexto y justificación del Trabajo

Aunque la aparición por primera vez del concepto “Internet de las cosas” (*IoT, Internet of Things*) se podría situar cronológicamente en el año 1999 cuando, el entonces director ejecutivo del *Auto-ID Center* en el MIT, Kevin Ashton lo introdujo como título de una presentación [1], ha sido en los últimos años cuando el paradigma de IoT está experimentando el impulso definitivo, posicionándose como uno de los segmentos tecnológicos con mayor potencial de crecimiento futuro.

Y es que, si atendemos a las últimas previsiones publicadas por la consultora Gartner, Inc. [2], en 2016 habrá en el mundo 6400 millones de dispositivos conectados, lo que correspondería a un incremento del treinta por ciento respecto al año anterior. La proyección es, según Gartner, que en el año 2020, ese número alcance los 20.800 millones de dispositivos. Fabricantes, habilitadores de soluciones u operadores de red, son algunos de los muchos actores que están permitiendo este crecimiento, y haciendo que el concepto de IoT sea finalmente una realidad. En este sentido, la rapidez con que las empresas, administraciones y usuarios finales puedan sacar partido de todo su potencial, pasa por poder realizar un despliegue de miles de sensores de los que recibir información de manera sencilla.

Como se podrá intuir, las posibilidades que IoT ofrece son innumerables: despliegue de sensores en zonas remotas para monitorizar infraestructuras, detección de incendios, monitorización de cultivos, actividades deportivas y un largo etcétera, incluyendo, entre otras, todas aquellas aplicaciones relacionadas con el concepto de “Ciudades Inteligentes” (*Smart Cities*). Por otro lado, una de las características principales del paradigma de IoT es la ubicuidad, es decir, la capacidad de que los objetos puedan estar conectados en cualquier lugar, y es ahí donde reside la motivación principal del presente trabajo, pues en última instancia, la ubicuidad estará estrechamente relacionada con la cobertura del servicio que el operador de red sea capaz de ofrecer en una región determinada.

Actualmente, SIGFOX es el primer operador que ha desplegado una red de acceso inalámbrica global de larga distancia de tipo LPWAN (*Low-Power Wide-Area Network*) dedicada a IoT [3]. Su tecnología de acceso está basada en modulación de banda ultra estrecha UNB (*Ultra-Narrow Band*) y, aunque ésta difiere de otras soluciones como LoRaWAN [4], en la que se usan modulaciones de espectro ensanchado, tanto la rapidez de penetración en el mercado, como la facilidad para interconectar sensores a la red, están haciendo que SIGFOX se esté convirtiendo en una opción muy interesante para aquellas soluciones que requieran transmitir muy poco volumen de datos de forma energéticamente eficiente.

Por otro lado, aunque la cobertura estimada en una determinada zona puede ser obtenida a partir de las simulaciones ofrecidas por SIGFOX, bien es cierto que dichas simulaciones no dan garantía de servicio, tal como se recoge en la propia web de la compañía [5]:

“This map is only a guide and not a guarantee of service level. Coverage estimation based on computer prediction in outdoor location for device class U0 (maximum output power level as defined in SIGFOX Ready™ certification requirements).

SIGFOX attempts to provide accurate and complete information through this online coverage estimation tool. However, neither SIGFOX nor its Distributors within the regions covered by this map can guarantee the quality, accuracy or completeness of the information. The coverage estimation tool is for information purposes and is provided "as is" without warranties of any kind, either express or implied. SIGFOX and the relevant Distributors within the regions covered by this map assume no liability for any damages resulting from the use or misuse of the coverage estimation tool.”

Antes de planificar por tanto, el diseño y despliegue de los sensores en una zona geográfica determinada para dar solución a una necesidad particular, se debe conocer con detalle si el servicio está realmente disponible en esa zona y con qué calidad.

Es por ello que se hace necesario disponer de una herramienta que permita comprobar con detalle la cobertura real a nivel de servicio existente en la zona bajo estudio y que sirva para extraer conclusiones que ayuden a la optimización del diseño de la solución. En este sentido, si esa herramienta permitiese simular por ejemplo, las condiciones reales de funcionamiento de los sensores así como las antenas a usar, se podría comprobar cuál es la potencia mínima necesaria para establecer el enlace, optimizando tanto la potencia a transmitir, como las localizaciones que permitirían, por ejemplo, maximizar el nivel de redundancia (número de estaciones base que reciben simultáneamente los mensajes enviados por los sensores).

El presente trabajo final de máster pretende por tanto, desarrollar e implementar una herramienta que permita verificar el nivel de servicio real que la red SIGFOX ofrece en una zona geográfica determinada y que pueda servir de ayuda a aquellos proyectos en los que se requiera conocer con exactitud cuál es el nivel de conectividad de la red de acceso. Asimismo, se realizará una campaña de medidas sobre un área geográfica de orografía compleja que permita obtener un mapa de conectividad de la zona bajo estudio.

1.2 Objetivos del Trabajo

Tal como se avanzó en el capítulo introductorio, el presente trabajo tiene como principal objetivo desarrollar una herramienta (*hardware/software*) que permita verificar de forma sencilla el nivel de servicio de la red SIGFOX en un área determinada. En este apartado se describirán los objetivos concretos a cumplir, quedando así definido el alcance del trabajo.

La herramienta a implementar se caracterizará por los siguientes puntos:

- Usando una plataforma de desarrollo de prototipos de bajo coste (tipo Arduino o similar) con módulo de comunicaciones SIGFOX, permitirá la transmisión de mensajes a la red de acceso:
 - Estos mensajes deberán ser transmitidos a demanda por medio de un pulsador o sistema equivalente y deberán contener información de geolocalización.
 - El nivel de señal transmitido podrá ser configurado mediante pulsador o sistema equivalente para simular distintos tipos de dispositivos (0U, 1U, etc. según certificación SIGFOX).
 - Además, se dispondrá de información visual (tipo led o similar) con el estado de la transmisión del mensaje (transmisión *ok/ not ok*) y el estado del GPS (*locked/unlocked*).
 - Los mensajes transmitidos deberán quedar registrados para su posterior análisis.
- Se desarrollará una herramienta software que permitirá el análisis de la conectividad (cobertura a nivel de servicio):
 - La aplicación recopilará los mensajes transmitidos por el módulo y los comparará con los mensajes recibidos por el *backend* de la red SIGFOX.
 - Determinará si, para cada punto geolocalizado, se ha producido recepción o no.
 - Recopilará otra información relevante procedente del *backend*: número de estaciones base que han recibido el mensaje, nivel de señal recibido (RSSI), relación señal/ruido (SNR), etc.
 - Presentará cada uno de estos puntos de forma amigable en un mapa de conectividad mediante un código de colores.
- Como demostrador y aplicación práctica, se realizará un estudio de conectividad sobre un área de orografía compleja (isla de Gran Canaria):
 - Se determinarán las coordenadas de los puntos geográficos que se pretenden medir, haciendo hincapié en aquellos puntos de alto interés (presas, embalses, zonas rurales, montes, etc.).
 - Se realizará una campaña de medidas de campo para verificar la conectividad de los puntos anteriores.
 - Adicionalmente se publicarán esos datos en una web en forma de mapa y que será de acceso público, con el objeto de disponer de una primera aproximación de zonas con servicio SIGFOX verificado y que sirva de ayuda a administraciones, empresas y usuarios finales a la hora de implantar sus soluciones de IoT.

1.3 Enfoque y método seguido

Teniendo en cuenta los objetivos a alcanzar ya expuestos en apartados anteriores, y después de realizar una primera búsqueda sobre posibles soluciones ya existentes que puedan cubrir los requisitos, se llega a la conclusión de que tanto el prototipo hardware como el desarrollo de la aplicación, deben ser implementados desde cero al no encontrar en el mercado soluciones que cumplan con los requisitos.

El proyecto, por tanto, queda desglosado en tres partes claramente diferenciadas con un alcance bien definido. Por un lado, será necesario fabricar un prototipo que cubra las necesidades planteadas y por otro, habrá que desarrollar el software que permita analizar los datos de conectividad. La última etapa del proyecto consistirá en la toma de medidas reales, empleando para ello los dos desarrollos anteriores.

Al tratarse de un trabajo ambicioso en cuanto a la cantidad de objetivos a alcanzar en un tiempo limitado, una correcta planificación inicial, así como una buena gestión de los riesgos son fundamentales para conseguir cumplir los plazos en tiempo y calidad.

En relación a la gestión de los riesgos, ya desde las fases tempranas de la definición del trabajo, se han detectado dos riesgos principales que pueden repercutir directamente sobre el éxito del proyecto.

Por un lado, los plazos de entrega de los proveedores de los componentes *hardware* que formarán parte del prototipo deberán estar perfectamente controlados, existiendo una fecha límite a partir de la cual se hace inviable contar con un determinado proveedor.

Por otro lado, el riesgo de avería de alguno de los componentes hace necesario disponer de una segunda unidad de repuesto de aquellos que sean críticos por la dificultad de disponer de los mismos en un plazo razonable.

Es por ello que en la planificación temporal del trabajo, se deberá tener en cuenta lo anteriormente expuesto, intentando planificar la compra de material en las fases iniciales.

En cuanto al método a seguir, se enfocará el proyecto por objetivos, siendo fundamental alcanzar cada uno de los hitos previstos dentro de plazo. Tal como se verá en el siguiente apartado, donde se desglosa en detalle la planificación, cada uno de los hitos corresponderá al cierre de una fase determinada, siendo por tanto crucial cerrar una etapa para poder avanzar hacia la siguiente.

1.4 Planificación del Trabajo

En este apartado se presenta la planificación inicial propuesta con los diferentes hitos a cumplir durante la duración del trabajo. Esta planificación se ha calculado en base a la disponibilidad real de los recursos, con a una dedicación diaria de cuatro horas durante seis días a la semana.

Las tareas a realizar se han agrupado en tres bloques principales, y como se puede comprobar en la Tabla 1, cada uno de estos bloques termina con un hito que se corresponde con los diferentes objetivos a alcanzar. Los bloques principales del desarrollo son:

Desarrollo hardware

Corresponde al diseño y fabricación del prototipo que será usado para la transmisión de mensajes geolocalizados hacia la red SIGFOX y que será empleado durante la campaña de medidas de campo.

Desarrollo software

Permitirá obtener la aplicación encargada de recuperar los mensajes tanto del dispositivo como del *backend* de la red SIGFOX para su análisis y presentación.

Medidas de campo

Corresponde a la campaña de medidas de campo, así como a la recopilación, presentación de los datos y análisis de los mapas de simulación.

Tabla 1. Planificación de tareas

EDT	Nombre de tarea	Duración	Comienzo	Fin
1	DEFINICIÓN	4 horas	mar 01/03/16	mar 01/03/16
1.1	Alcance y objetivos	2 horas	mar 01/03/16	mar 01/03/16
1.2	Arquitectura	2 horas	mar 01/03/16	mar 01/03/16
2	DESARROLLO HARDWARE	108 horas	mié 02/03/16	vie 01/04/16
2.1	Selección de componentes	4 horas	mié 02/03/16	mié 02/03/16
2.2	Compras y Licencia SIGFOX	4 horas	jue 03/03/16	jue 03/03/16
2.3	Instalación del entorno de desarrollo	1 hora	jue 03/03/16	jue 03/03/16
2.4	Formación y entrenamiento	20 horas	jue 03/03/16	mié 09/03/16
2.5	Integración de Componentes	8 horas	lun 14/03/16	mar 15/03/16
2.6	Desarrollo del Firmware	56 horas	mié 16/03/16	jue 31/03/16
2.6.1	Programación módulo GPS	16 horas	mié 16/03/16	sáb 19/03/16
2.6.2	Programación módulo SIGFOX	24 horas	lun 21/03/16	sáb 26/03/16
2.6.3	Programación E/S	16 horas	lun 28/03/16	jue 31/03/16
2.7	Pruebas funcionales del prototipo	4 horas	vie 01/04/16	vie 01/04/16
2.8	Prototipo terminado	0 horas	vie 01/04/16	vie 01/04/16
3	DESARROLLO SOFTWARE	103 horas	lun 04/04/16	mar 03/05/16
3.1	Selección del lenguaje	1 hora	lun 04/04/16	lun 04/04/16
3.2	Instalación del entorno de desarrollo	3 horas	lun 04/04/16	lun 04/04/16
3.3	Desarrollo de la aplicación	91 horas	mar 05/04/16	sáb 30/04/16
3.4	Pruebas funcionales de la aplicación	8 horas	sáb 30/04/16	mar 03/05/16
3.5	Aplicación terminada	0 horas	mar 03/05/16	mar 03/05/16
4	MEDIDAS DE CAMPO	41 horas	sáb 07/05/16	jue 19/05/16
4.1	Definición de criterios	1 hora	sáb 07/05/16	sáb 07/05/16
4.2	Selección de coordenadas de medida	4 horas	lun 09/05/16	lun 09/05/16
4.3	Campaña de medidas	24 horas	mar 10/05/16	lun 16/05/16
4.4	Resultados y simulaciones	12 horas	mar 17/05/16	jue 19/05/16
4.5	Medidas de campo finalizadas	0 horas	jue 19/05/16	jue 19/05/16
5	DOCUMENTACIÓN	376 horas	lun 07/03/16	jue 23/06/16
5.1	Desarrollo PEC1. Definición	50 horas	lun 07/03/16	lun 21/03/16
5.2	PEC1 entregada	0 horas	lun 21/03/16	lun 21/03/16
5.3	Desarrollo PEC2. Estado del arte	48 horas	mar 22/03/16	lun 04/04/16
5.4	PEC2 entregada	0 horas	lun 04/04/16	lun 04/04/16
5.5	Desarrollo PEC3. Diseño e implementación	156 horas	mar 05/04/16	jue 19/05/16
5.6	PEC3 Entregada	0 horas	jue 19/05/16	jue 19/05/16
5.7	Desarrollo PEC4. Memoria	72 horas	vie 20/05/16	jue 09/06/16
5.8	PEC4 entregada	0 horas	jue 09/06/16	jue 09/06/16
5.9	Desarrollo PEC5. Presentación	48 horas	vie 10/06/16	jue 23/06/16
5.10	PEC5 entregada	0 horas	jue 23/06/16	jue 23/06/16

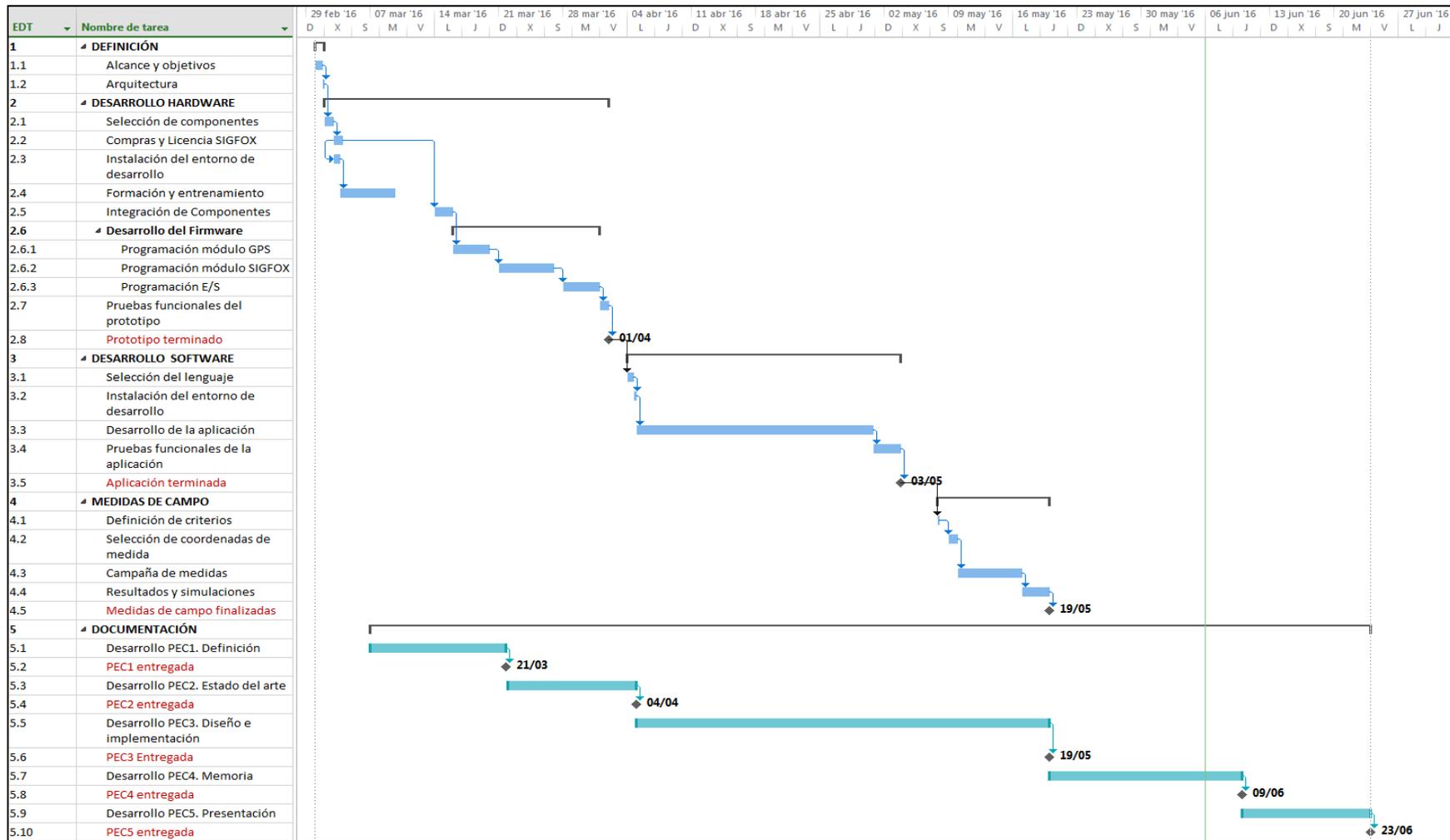


Figura 1. Diagrama de Gantt de la planificación

Como se puede extraer del diagrama de Gantt asociado a la planificación representado en la Figura 1, paralelamente a la ejecución del desarrollo técnico, se ha incorporado un bloque que corresponde con la redacción de los diferentes entregables. Cada una de estas tareas termina por tanto con un hito de entrega de documentación. En este sentido, en las horas asignadas a cada una de las tareas de los bloques de desarrollo técnico, ya se encuentran incluidas las horas destinadas a la preparación de esta documentación.

Cabe también destacar que se ha planificado cierta holgura entre la terminación de un bloque y el comienzo del siguiente para permitir absorber posibles retrasos o replanificaciones que sean detectados durante la fase de seguimiento y control del proyecto.

Finalmente, en la Tabla 2 se muestra resumidamente la lista de hitos del trabajo.

Tabla 2. Lista y descripción de los diferentes hitos de la planificación

FECHA	HITO	DESCRIPCIÓN
21/03/2016	PEC1 entregada	Primer entregable de documentación. Corresponde a la definición del Trabajo
01/04/2016	Prototipo terminado	Este hito corresponde a la finalización del primer bloque de desarrollo. El entregable principal es el prototipo completamente funcional
04/04/2016	PEC2 entregada	Segunda entrega de documentación. Corresponde al "estado del arte"
03/05/2016	Aplicación terminada	Este hito corresponde a la finalización del segundo bloque de desarrollo. El entregable principal es la aplicación software completamente funcional
19/05/2016	Medidas de campo finalizadas	Este hito corresponde a la finalización de la fase de la campaña de medidas. Incluye la recopilación y presentación de resultados
19/05/2016	PEC3 entregada	Tercer entregable de documentación. Corresponde con toda la documentación generada en la fase de implementación
09/06/2016	PEC4 entregada	Cuarto entregable de documentación. Corresponde a la entrega de la memoria
23/06/2016	PEC5 entregada	Quinto entregable de documentación. Corresponde a la entrega de la presentación

1.5 Breve resumen de productos obtenidos

En este apartado se presentarán, de forma resumida, los productos obtenidos una vez finalizado el desarrollo descrito en los diferentes apartados de esta memoria. Estos productos corresponden a la implementación de los diferentes módulos que componen la arquitectura propuesta diseñada y que será explicada en detalle en la primera parte del capítulo 3. Así pues, los productos finales son los siguientes:

- Prototipo hardware, incluyendo el ensamblado y mecanización de los diferentes módulos que lo componen (placa controladora, receptor GPS, módulo de comunicaciones SIGFOX y teclado).

- Programa de control del prototipo (*firmware*). Es el programa que se ha desarrollado y programado en el microcontrolador de la placa de control y que permite el correcto funcionamiento del prototipo hardware, cumpliendo así con los requisitos de diseño.
- Herramienta software de análisis de conectividad. Esta herramienta presenta una interfaz de usuario cómoda desde donde procesar los datos transmitidos por el prototipo, y recibidos por la red SIGFOX, realizando un análisis de cobertura para las localizaciones que se desean estudiar. Presenta los resultados en forma de mapa, además de otras estadísticas calculadas.
- Mapa con puntos de cobertura real de la red SIGFOX en la isla de Gran Canaria. Este es el resultado de las medidas de campo llevadas a cabo en la isla para validar el prototipo y el software desarrollado. Incluye más de 170 puntos de medida y permite conocer si en dichos emplazamientos hay o no servicio, indicando además el número de estaciones base que cubren la zona así como otros datos de interés.
- Análisis comparativo entre el mapa de cobertura simulado proporcionado por SIGFOX, y las medidas reales realizadas durante la campaña. Esta comparación cualitativa permite tener una visión de cuánto se aproxima la simulación proporcionada por el operador, a las medidas reales.

1.6 Breve descripción de los otros capítulos de la memoria

La memoria queda pues estructurada como se describe a continuación.

El capítulo 2 es dedicado al análisis del estado del arte, comenzando con una breve reseña sobre las redes LPWAN y presentando la bibliografía existente sobre medidas de cobertura sobre las mismas. El capítulo finaliza con un apartado dedicado a las contribuciones específicas aportadas con la realización del presente trabajo.

El capítulo 3 entra ya de lleno en el diseño del sistema, comenzando con la presentación de la arquitectura propuesta. Este capítulo se centra en el diseño del prototipo, describiendo en cada apartado cómo se ha realizado la integración de los diferentes módulos que lo componen.

El capítulo 4 se centra en el desarrollo e implementación de la herramienta software de análisis describiendo, en cada apartado, cada uno de los módulos que forman la aplicación. La forma en que ha sido redactado permite seguir la secuencia típica desde que se importan los datos transmitidos y recibidos, hasta que se procesan y se obtiene el mapa de servicio de la zona bajo estudio.

El capítulo 5 cubre la fase de medidas de campo. Describe, desde la planificación de la misma en función de las características particulares de la zona bajo estudio, hasta la presentación de los resultados obtenidos.

Finalmente, en el capítulo 6 se presentan las conclusiones.

El resto de capítulos, del 7 al 9, corresponden al glosario, bibliografía y anexos de la memoria respectivamente.

2. Estado del Arte

Este capítulo abordará la puesta en contexto del presente trabajo, presentando las principales aportaciones encontradas en la literatura relacionadas con el mismo. Se comenzará con una breve introducción de las nuevas redes de acceso de bajo consumo y largo alcance dedicadas a IoT, presentando las diferentes tecnologías existentes y que nos permitirá tener una visión global de dónde quedan enmarcados los objetivos del trabajo. Continuaremos el estudio del estado del arte presentando aquellos trabajos relacionados con los estudios existentes sobre coberturas en redes de bajo consumo y largo alcance LPWAN (*Low-power Wide-Area Network*). Ya en la última parte del capítulo haremos un recorrido por los dispositivos disponibles en el mercado que por sus características puedan presentar funcionalidades similares a las que se proponen en este trabajo.

Cabe destacar, tal como se podrá comprobar durante la lectura del capítulo, que en general, la información publicada en relación a dispositivos específicamente destinados a la comprobación del servicio para redes LPWAN, así como a estudios sobre medidas de campo, es muy reducida. Es por ello que, debido a la escasez de publicaciones dedicadas explícitamente al campo de aplicación de este trabajo, se ha recurrido en algunos casos, a extraer dicha información de secciones determinadas de otros trabajos más generales.

2.1 Redes LPWAN

El constante crecimiento que IoT está experimentado en los últimos años, ha ido acompañado por la aparición de un ecosistema de diferentes tecnologías de comunicación que de una forma u otra, están permitiendo que se puedan dar las premisas fundamentales de ubicuidad, bajo consumo o bajo coste que caracterizan los retos de este paradigma tecnológico. Aunque podemos citar tecnologías que, por ejemplo, han sido diseñadas para cumplir los requisitos de bajo consumo como *ZigBee* o BLE (*Bluetooth Low Energy*), bien es cierto que éstas están enfocadas a comunicaciones orientadas al corto alcance, utilizadas principalmente por redes de sensores o redes personales de área local.

Por otro lado, y después de años sin existir un referente tecnológico claro de conectividad que permitiese el despegue definitivo del Internet de las cosas, actualmente estamos asistiendo a un punto de inflexión con la aparición de nuevas soluciones, como *Low-Power WiFi*, diversas mejoras en las redes celulares, como LTE-M orientada al M2M (*Machine to Machine*) o las LPWANs [6]. En esta sección nos centraremos concretamente en las tecnologías asociadas a esta última, por la rapidez con que están penetrando en el mercado, y por ser la tecnología radio donde encaja la red SIGFOX, objeto de este trabajo.

Tal como su nombre indica, las redes LPWAN, son redes de acceso que permiten dar conectividad a objetos a gran distancia, empleando un bajo consumo energético para ello. Se caracterizan por trabajar en las bandas del espectro sin licencia (*unlicensed*

spectrum) y, en general, están destinadas a aquellas aplicaciones que requieran poca transmisión de información. Dentro de las tecnologías que encajan en las LPWAN, tal como enumera Centenaro et al. [7], podemos citar las siguientes, cada una de ellas con una estrategia en el diseño de la capa física diferente.

Por una lado, la compañía **Ingenu** [8] proporciona una tecnología de acceso patentada llamada *Random Phase Multiple Access* (RPMA) basada en modulación *Direct-Sequence Spread Spectrum* (DSSS). A diferencia de las otras tecnologías, que emplean bandas por debajo del GHz, Ingenu ha apostado por usar la banda de 2.4 GHz, la cual permite, mediante el uso de RPMA alcanzar grandes distancias de cobertura. Uno de sus puntos fuertes frente a otras tecnologías competidoras, según la propia compañía, es la verdadera capacidad de escalar el número de dispositivos conectados a la red a millones de dispositivos.

Por otro lado, el sistema **LoRa** [4] diseñado y patentado por Semtech Corporation está basado en modulación de tipo *Chirp Spread Spectrum* (CSS), en la que se han introducido innovaciones para evitar la complejidad de costosos componentes que mantengan la estabilidad de reloj en cada nodo [7]. La topología de red es de tipo estrella, siendo el *gateway* el encargado de interconectar directamente el nodo que transmite el mensaje hacia el servidor correspondiente de forma transparente. Cabe destacar como particularidad de esta tecnología, que los nodos pueden ser –o pueden ser configurados- de tres formas diferentes. Los dispositivos de clase A, optimizados sólo para la transmisión de información; los de clase B, orientados a aquellas aplicaciones que demanden la necesidad de poder mandar comandos en ventanas temporales prefijadas; y finalmente los de clase C, en los que el nodo no tiene restricciones de consumo permitiendo así la escucha continua para recibir comandos a demanda.

Finalmente, **SIGFOX** [3] es una red operada de comunicaciones dedicada a IoT. La principal diferencia respecto a su modelo de negocio en relación a las dos soluciones anteriores, reside en que la compañía actúa directamente como proveedor de servicio, permitiendo que el despliegue de sensores se pueda llevar a cabo por el usuario de forma sencilla sin la necesidad de instalación y mantenimiento de *gateways* propios para redirigir los mensajes hacia los servidores. En este caso, la capa física está basada en modulación de tipo *Ultra-Narrow Band* (UNB) con una velocidad de transmisión de 100 bps. El hecho de trabajar en la banda sin licencia de 868 MHz hace que la capacidad final de transmisión de los nodos esté regulada por un mecanismo de *duty-cycle*, quedando limitada a los 140 mensajes de 12 bytes como máximo al día.

Tabla 3. Comparativa de diferentes tecnologías radio LPWAN [7]

	SIGFOX	Ingenu	LoRa
Alcance (Km)	Rural: 30-50 Urbano: 3-10	≈15	Rural: 10-15 Urbano: 3-5
Frecuencias (MHz)	868 ó 902	2400	Varios, sub-GHz
Banda ISM	Si	Si	Si
Enlace bi-direccional	Si	No	Si
Velocidad de datos (Kbps)	0.1	0.01-8	0.3-37.5
Nodos por Estación base	≈10 ⁶	≈10 ⁴	≈10 ⁴

2.2 Medidas de cobertura sobre redes LPWAN

A la hora de realizar una campaña de medidas de campo para verificar el rendimiento o la disponibilidad de un determinado servicio, cabe preguntarse cuáles son las cuestiones principales a tener en cuenta para llevarla a cabo y si existe algún método que permita afrontarla de forma sistemática con garantías. Turau et al. [9] aborda esta cuestión, particularizada para el caso de medidas de campo sobre redes de sensores. Aunque no se trata de un caso de uso que específicamente esté dedicado a redes LPWAN, merece la pena enumerar los aspectos claves que allí se describen y que hay que tener en cuenta para preparar una campaña de medidas.

Por un lado, la elección del **hardware** juega un papel primordial a la hora de poder extrapolar los resultados de las medidas a otros escenarios. En otras palabras, si para nuestras medidas empleamos, por ejemplo, un módulo de comunicaciones que únicamente permite transmitir a una potencia fija, los resultados serán válidos sólo para esas condiciones del test y extrapolarlos a otro tipo de hardware requerirá de un análisis riguroso.

Por otro lado, la elección de los **parámetros de interés** que se van a medir constituirán la base del análisis posterior, así como determinarán la capacidad que se tenga para poder extraer conclusiones a partir de las medidas. Algunos ejemplos de estos parámetros pueden ser, el número de mensajes enviados y recibidos, el consumo, el número de retransmisiones o el nivel de señal recibido.

Estos parámetros medidos se deben poder recuperar y es ahí donde entra en juego la estrategia de **almacenamiento**. Turau et al. aquí exponen seis aspectos a considerar:

- Integridad: relacionado con la capacidad del método de medida para no perder ninguna de las medidas realizadas.
- Disponibilidad: relacionado con el momento en que están los datos disponibles para su análisis.
- Escalabilidad: tiene que ver con la capacidad de la estrategia seguida para escalar cuando el número de sensores aumenta.
- Tiempo y esfuerzo: relacionado con la complejidad que supone el despliegue del sistema de almacenamiento.
- Consumo energético: cuál es el consumo empleado para realizar la tarea de guardado de datos.
- Cantidad: cuánta cantidad de datos es capaz de almacenar el sistema.

Otros aspectos destacados son las consideraciones sobre la **duración** de la campaña de medidas o sobre la **estrategia de despliegue** y planificación de las medidas.

Por último, otra de las cuestiones a tener en cuenta, es el nivel de **protección** de las carcasas envolventes de la electrónica del dispositivo empleado para las medidas [9]. Es un hecho que muchos de estos dispositivos forman parte de prototipos y conviene tener en cuenta las condiciones de medida a la que van a ser expuestos -no es lo mismo realizar medidas en un laboratorio que en un ambiente hostil de exterior-.

Centrando el enfoque hacia experiencias de medida de coberturas que se hayan realizado sobre redes LPWAN y, aunque no esté específicamente dedicado al estudio de la red SIGFOX objeto de este trabajo, Petäjajarvi et al. [10] realizan un estudio de

cobertura de la red LoRa, otra de las tecnologías empleadas en redes de bajo consumo y largo alcance ya descrita en el apartado anterior. Es quizá el estudio más completo y explícitamente enfocado a medidas de campo y comprobación de servicio de redes LPWAN publicado hasta la fecha de redacción del presente trabajo. Las medidas de campo se realizaron durante 14 días entre la primavera y verano de 2015 en la ciudad finlandesa de Oulu, caracterizada por presentar una orografía prácticamente plana. Éstas se realizaron tanto en tierra como en regiones de mar circundantes a la ciudad.

Para la realización de la campaña de medidas se instaló una estación base a una altura de 24 metros sobre el nivel del mar, a la que se acopló una antena de 2 dBi de ganancia. Como elemento móvil encargado de transmitir los mensajes a la red, se empleó el dispositivo comercial LoRaMote de Semtech [11], que incorpora una antena plana de tipo PIFA (*Planar Inverted-F Antenna*) integrada en la propia placa. Además de otros sensores, el LoRaMote dispone de un receptor GPS (*Global Positioning System*) que fue empleado para incluir información de geolocalización a cada mensaje transmitido.

La configuración de los parámetros de transmisión se ajustó para poder medir la distancia máxima desde la estación base a la que el sistema era capaz de recibir mensajes, configurando el LoRaMote para que transmitiese la posición proporcionada por el GPS a intervalos regulares e identificando cada una de esas transmisiones mediante un número secuencial.

En cuanto a los resultados, se generó un mapa de calor (Figura 2) a partir de la información de la potencia recibida por parte del *gateway* y la geolocalización de cada paquete. En ese mapa, queda reflejado mediante código de colores, el nivel de señal recibido por la estación base en función de la distancia a la misma.

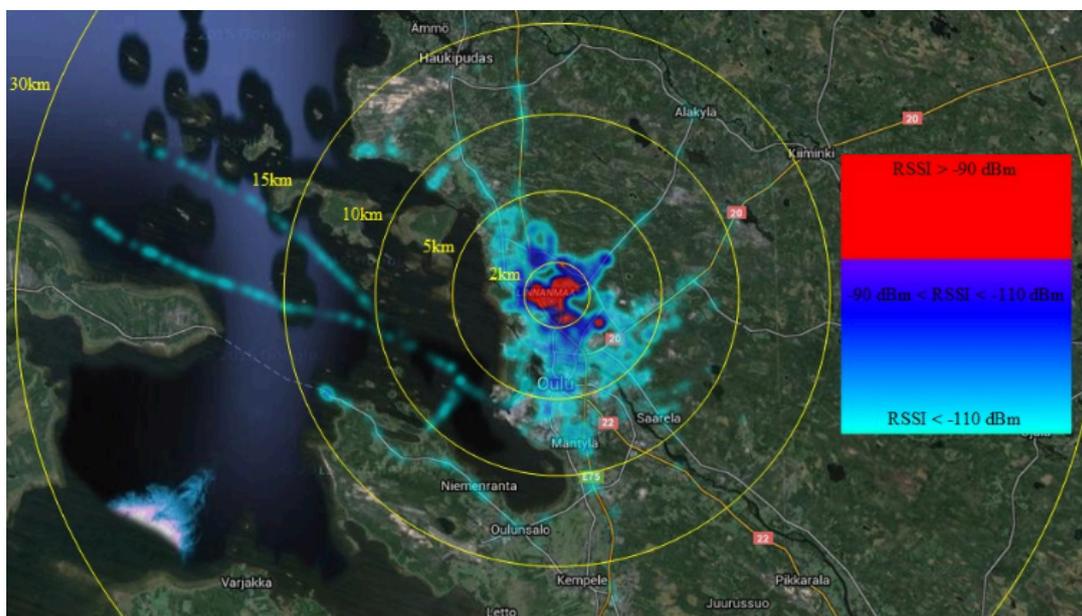


Figura 2. Mapa de calor representando el nivel de señal recibido desde diferentes posiciones en Oulu, Finlandia, centrado en la posición de la estación base ($P_{tx}=14$ dBm, $G_{rx}=2$ dBi, $R=293$ bps, $h_{tx}=2$ m, $h_{rx}=24$ m) [10].

También se presentan en el artículo otros datos de interés, como las estadísticas de paquetes perdidos así como una propuesta de modelo de atenuación del canal que pueda ser usado para estimar la máxima distancia de comunicación en regiones similares a la ciudad de Oulu empleando tecnología LoRa.

Por otro lado, aunque los siguientes dos artículos analizados no están dedicados expresamente a medidas de cobertura, incluyen sendos apartados en los que se evalúa ésta indirectamente mediante el análisis de los datos aportados por la solución concreta que implementan. Cabe destacar por tanto, que para esa evaluación no se ha empleado ningún dispositivo específicamente diseñado para tal efecto, sino que son conclusiones extraídas a partir de los mensajes transmitidos por los sensores bajo estudio. Así, Centenaro et al. [7] incluye una sección en su estudio en la que se explican los resultados de dos pruebas experimentales sobre tecnología LoRa. La primera de ellas consistió en una prueba de concepto para verificar la conectividad de múltiples sensores de temperatura y humedad instalados en diferentes habitaciones de un edificio de 19 plantas. Para ello se instaló un *gateway* LoRa en la novena planta y 32 sensores repartidos por el resto del edificio. Para probar la capacidad de conectividad de la tecnología en entornos hostiles, se hicieron pruebas colocando los sensores dentro de ascensores y en otras localizaciones habitualmente complicadas en cuanto a cobertura de radiofrecuencia. En todos los casos, según el experimento, se consiguió buena conectividad.

El segundo caso estudiado consistió en hacer pruebas de cobertura, también sobre una red LoRa en la ciudad italiana de Padua. El principal objetivo era intentar evaluar la cobertura en las peores condiciones para obtener una estimación conservadora del número de *gateways* que serían necesarios para cubrir la ciudad completa. Para ello se instaló un *gateway* con una antena de 0 dBi de ganancia sobre el tejado de un edificio de dos plantas que se encontraba rodeado por otras construcciones elevadas. En estas condiciones, los resultados mostraron que con esa tecnología se podía cubrir con garantía una celda de aproximadamente 1.2 Km de radio, incluidos los márgenes para posibles interferencias y variaciones en el balance de enlace.

Como último caso, cabe citar los resultados asociados al proyecto llevado a cabo por Llaría et al. [12] para seguir el comportamiento de rebaños en zonas de montaña. En este caso, se usó la red SIGFOX para enviar mensajes geolocalizados usando un dispositivo experimental, específicamente diseñado en el marco de ese proyecto. Las pruebas de campo se realizaron en dos localizaciones de montaña diferentes. La primera de ellas en Prat d'Albis, Ariège, Francia, en orografía montañosa de superficies planas entre 900 y 1600 m de altitud, y la segunda en Urola-Erdia, Guipúzcoa con una orografía de montañas escarpadas entre 400 y 600 metros. Los resultados muestran el rendimiento del sistema a partir de la pérdida de mensajes transmitidos sobre la red SIGFOX para cada una de las zonas geográficas.

2.3 Herramientas para la verificación del servicio de la red SIGFOX

El objeto de este apartado es presentar las soluciones de mercado o prototipos existentes que puedan ser empleados como dispositivos válidos para la comprobación de cobertura a nivel de servicio de la red SIGFOX. Aunque el planteamiento inicial se centraba en la catalogación y análisis de equipos que específicamente estuviesen

diseñados para la tarea concreta de poder comprobar el servicio de las redes de acceso antes de realizar un despliegue de sensores, rápidamente se constató que dichos dispositivos son prácticamente inexistentes como tales, habiéndose comprobado que para los escasos estudios de cobertura reflejados en la literatura, se usaron los propios sensores que iban a ser desplegados.

Es por ello que se decidió ampliar la catalogación para incluir aquellos dispositivos que por sus características o similitudes con el objeto de este trabajo, pudiesen ser empleados como herramientas de comprobación de servicio.



Figura 3. Demostrador SIGFOX [13]

La primera de esas herramientas ha sido desarrollada por la empresa Adeunis RF y se trata de un demostrador específico para probar la tecnología SIGFOX [13]. El objetivo del mismo, aparte de demostrador tecnológico, es el de poder verificar el servicio en la zona de interés. Este dispositivo es, por tanto, de toda la literatura consultada, el único encontrado que por sus características se asemeja al propósito de este trabajo. El equipo se presenta con unas dimensiones reducidas e incorpora un receptor GPS, un sensor de temperatura y un acelerómetro. Dispone de dos pulsadores para interactuar con él, además de dos indicadores de tipo LED que permiten conocer su estado.

El demostrador tiene dos modos de funcionamiento. El modo nominal consta de un ciclo que se repite continuamente cada diez minutos. Cada ciclo está compuesto por las siguientes fases: sincronización del GPS, transmisión del mensaje, activación del modo de recepción esperando la confirmación de los servidores de SIGFOX y confirmación de la recepción. Los datos de la posición desde donde se ha realizado la transmisión se pueden consultar en una dirección web de acceso restringido.

Aparte de este modo, el dispositivo dispone de la posibilidad de transmisión manual, que se puede activar pulsando uno de los botones, o bien agitando el demostrador. Este modo fuerza una transmisión siempre que el dispositivo se encuentre en el intervalo entre dos ciclos del modo nominal.

En cuanto a otros prototipos que incluyan módulo de comunicaciones SIGFOX, además de un sistema GNSS (*Global Navigation Satellite System*) y que puedan ser usados para la tarea objeto de este trabajo -previo desarrollo del correspondiente *firmware*-, podemos destacar el dispositivo *SmartEverything*, fabricado por la compañía Arrow [14]. Se trata de un dispositivo diseñado con tecnología de bajo consumo que, aparte de la conectividad SIGFOX y geolocalización, dispone de otros sensores comúnmente empleados en soluciones IoT (acelerómetros, sensores de temperatura, humedad y proximidad, NFC, BLE). Evidentemente, esta placa de desarrollo de prototipos de uso general, no se puede considerar como una herramienta específica de verificación de servicio de la red SIGFOX, pero sus características técnicas la proponen como un candidato de cara a poder desarrollar soluciones para tal efecto.

2.4 Contribuciones de este trabajo

A lo largo del capítulo hemos esbozado el contexto actual donde se enmarcan los objetivos de este trabajo final de máster, se ha realizado un recorrido por las publicaciones más relevantes relacionadas con medidas de campo sobre redes LPWAN y se han presentado aquellos dispositivos que por su similitud o funcionalidad podían ser comparados con el desarrollado aquí. También se ha constatado que, en general, escasean las publicaciones sobre medidas de campo realizadas sobre la red SIGFOX, así como las herramientas disponibles para poder verificar el servicio de dicha red.

Tal como se ha comentado ya en otros apartados introductorios, a la hora de diseñar y planificar un despliegue de sensores dentro de un área determinada, además de ser necesario comprobar que el servicio efectivamente se encuentra disponible en los emplazamientos proyectados, resulta de gran utilidad para el diseñador conocer otros parámetros importantes para optimizar la solución final propuesta, que faciliten pues la tarea de encontrar localizaciones que permitan alcanzar la mejor implementación. Cuestiones como el número de estaciones base simultáneas que reciben los mensajes transmitidos, o el nivel de señal recibido por las mismas, son parámetros de gran utilidad que permiten, por ejemplo, saber si los dispositivos a emplear necesitan transmitir a la máxima potencia permitida o, por el contrario, dicha potencia puede ser reducida para optimizar el consumo.

Por otro lado, poder realizar las pruebas de verificación del servicio empleando un dispositivo que sea capaz de simular las condiciones reales a las que van a funcionar los sensores previstos - como por ejemplo el tipo de antena a emplear, o la potencia de transmisión a la que funcionan-, permite obtener unos resultados realistas sobre cómo va a ser el rendimiento del diseño final.

Este trabajo propone pues el desarrollo de una solución completa que permite cubrir los aspectos anteriormente citados, y que actualmente no se encuentran disponibles como solución integral en el mercado. De esta forma, aparte de las funcionalidades comunes a otras soluciones disponibles, como son la capacidad de geolocalizar la medida, el prototipo desarrollado permite configurar hasta tres potencias diferentes para simular cualquier dispositivo en el rango de 0 dBm hasta 14 dBm. Por otro lado, las medidas de campo son almacenadas en la memoria interna, lo que permite, mediante la descarga y análisis posterior, comprobar no sólo aquellas zonas donde hay servicio sino poder georeferenciar también aquellas en las que no se ha conseguido establecer el enlace.

Cabe mencionar también, que el hecho de disponer de una antena desmontable con interfaz de conexión estándar, permite realizar las pruebas empleando cualquier otro elemento radiante que pueda simular más fielmente las condiciones reales de funcionamiento.

En cuanto al software desarrollado asociado al prototipo, y que en conjunto proporcionan una herramienta integral para la verificación del servicio, éste permite realizar el análisis y la presentación gráfica de las medidas de campo de una forma semiautomática y sencilla.

Finalmente, se muestra a continuación una tabla comparativa de las capacidades principales que aporta la solución aquí desarrollada, frente al demostrador SIGFOX de Adeunis. En este caso, no se ha incluido la placa *SmartEverything* debido a que no es directamente comparable, pues se trata de una placa de desarrollo de prototipos de propósito general que, si bien contiene todos los elementos necesarios para poder ser usada como herramienta de comprobación de servicio, sería necesario desarrollar el firmware de control que permitiese hacerlo.

Tabla 4. Comparativa entre el demostrador SIGFOX y la herramienta desarrollada en este proyecto

Funcionalidad	Demostrador SIGFOX de Adeunis	Herramienta desarrollada en este proyecto
Capacidad de transmitir a la máxima potencia permitida por la regulación (14 dBm, dispositivo clase 0U según certificación SIGFOX)	Sí	Sí
Transmisión automática a intervalos regulares	Sí	No
Capacidad de realizar transmisión manual	Sí	Sí
Transmisión de mensaje con petición de ACK (<i>Acknowledgment</i>)	Sí	Sí
Transmisión de mensaje sin petición de ACK	No	Sí
Indicador de estado del GPS y de la transmisión	Sí	Sí
Capacidad de modificar la potencia transmitida	No	Sí
Antena desmontable	No	Sí
Capacidad de almacenamiento interno de los datos transmitidos para posterior análisis	No	Sí
Presentación de puntos con cobertura en mapa	Sí	Sí
Presentación de puntos sin cobertura en mapa	No	Sí
Capacidad de programar la potencia a transmitir	No	Sí
Herramienta software para análisis de puntos de cobertura y generación de mapa	No	Sí
Presentación del número de estaciones base que cubren el punto de transmisión (nivel de redundancia)	No	Sí
Presentación de niveles máximo/mínimo de la relación señal a ruido y de la potencia recibida por las estaciones base	No	Sí
Cálculo de datos estadísticos sobre una campaña de medidas (porcentaje de puntos con servicio, redundancia, total de puntos sin cobertura, etc.)	No	Sí

3. Desarrollo del prototipo

Tal como se avanzaba en la sección dedicada a la planificación, el desarrollo de este trabajo se puede descomponer en tres etapas perfectamente diferenciadas: desarrollo del prototipo hardware, desarrollo de la herramienta software de análisis y, finalmente, la descripción y resultados de la campaña de medidas de campo. Es por ello que, para contribuir a la claridad del documento, se hace conveniente separarlas en tres secciones.

Este primer capítulo comprende pues la descripción detallada del diseño e implementación del prototipo encargado de realizar las transmisiones geolocalizadas hacia la red SIGFOX. Se comenzará presentando la arquitectura de la solución general propuesta y que servirá de base para tener una visión de contexto de dónde quedan encajados los diferentes desarrollos dentro del proyecto. Posteriormente, en los siguientes apartados se tratará en detalle la integración de los diferentes módulos hardware que forman el prototipo. Finalmente se presentarán las pruebas funcionales a las que se ha sometido el prototipo ya ensamblado y que será usado en la campaña de medidas.

3.1 Arquitectura general del sistema

A la hora de representar la arquitectura del sistema, se deben tener en cuenta las siguientes premisas:

- El prototipo hardware a desarrollar transmitirá mensajes a demanda, conteniendo cada mensaje la información de la posición desde donde se ha efectuado la transmisión. Estos mensajes deben ser almacenados en la memoria interna para el posterior análisis.
- Estos mensajes, en caso de que haya cobertura de servicio de la red SIGFOX en la zona de transmisión, serán recibidos por una o varias estaciones base, y serán retransmitidos hacia sus servidores. Desde los servidores de SIGFOX (*backend*), estos mensajes deben ser rutados hacia un servidor de usuario para su posterior análisis.
- La herramienta software a desarrollar será capaz de:
 - Importar los datos almacenados en el prototipo
 - Importar los datos recibidos del *backend* de SIGFOX
 - Analizar los datos y presentar los resultados

La siguiente figura muestra la arquitectura propuesta siguiendo lo expuesto en los párrafos anteriores.

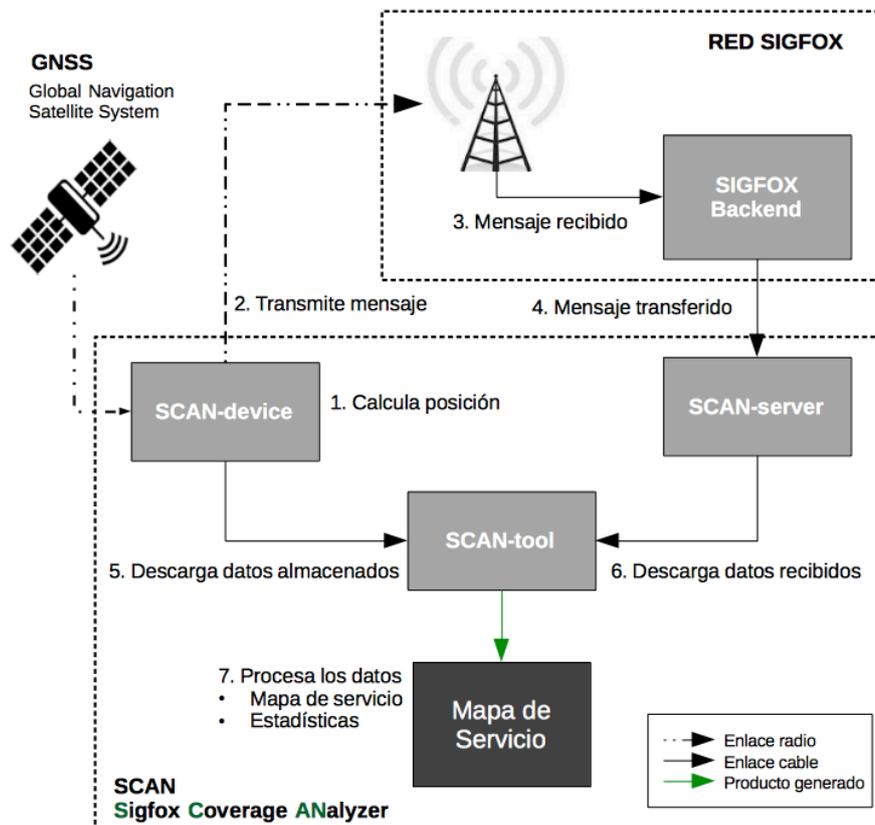


Figura 4. Arquitectura propuesta. Se presenta la secuencia típica, desde que se genera un mensaje en el prototipo, hasta que se obtienen los resultados del análisis de cobertura.

Como se puede observar en el diagrama, la herramienta a desarrollar, que por claridad en el documento será llamada de aquí en adelante SCAN (**Sigfox Coverage ANalyzer**), consta de tres partes: el prototipo hardware (SCAN-device), la herramienta software (SCAN-tool) y el servidor donde se recibirán los datos del *backend* de SIGFOX (SCAN-server). Cabe mencionar que este último módulo, no se refiere al desarrollo de un servidor propiamente dicho, sino al programa alojado en él que será el encargado de ir almacenando los datos recibidos.

Finalmente, la configuración del *backend* necesaria para el correcto funcionamiento del sistema, será tratada en el apartado correspondiente al desarrollo del SCAN-tool.

3.2 Diagrama de bloques del SCAN-device

Teniendo en cuenta los requisitos necesarios a cumplir por el prototipo, y que ya han sido expuestos en el apartado 1.2, se puede derivar que éste debe contener, al menos, los siguientes elementos:

- Módulo receptor GPS que permita obtener la posición del prototipo.
- Módulo de comunicaciones SIGFOX que permita enviar mensajes hacia la red.
- Capacidad de almacenamiento interna que permita la persistencia de los datos transmitidos.

- Puerto de entrada/salida que permita la comunicación con el exterior para su configuración y extracción de datos.
- Capacidad de proporcionar una interfaz de usuario mediante pulsadores e indicadores que permitan interactuar con él.

Se plantean pues dos posibles soluciones que satisfacen estas necesidades:

Placa compacta con hardware integrado

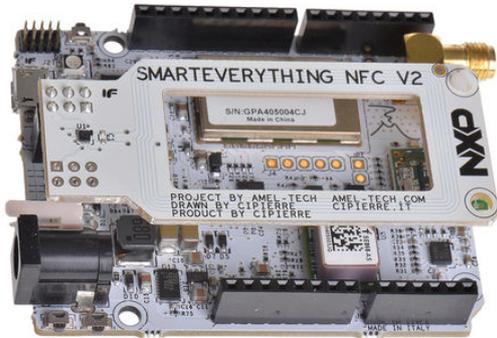


Figura 5. Placa SmartEverything [15].

Esta solución pasa por emplear una placa comercial que integre todos esos elementos. Sin duda sería la opción más simple, pues el desarrollo se limitaría a la programación del microcontrolador. Dentro de las posibilidades existentes en el mercado, se valoró la placa *SmartEverything*, de la compañía Axel Elettronica [15]. Se trata de una placa para realizar prototipos orientados a IoT, siendo sus principales características las siguientes:

- Está basado en un microcontrolador de bajo consumo Atmel D21.
- Incluye módulo de comunicaciones SIGFOX y GPS con antena integrada.
- Incorpora sensores de proximidad, humedad, temperatura y presión, además de acelerómetros.
- Dispone de NFC y BLE, además de varios pulsadores y LEDs de propósito general.
- Compatible con el entorno de programación Arduino IDE.

Como se puede comprobar, además de los elementos estrictamente necesarios, la placa incorpora otros muchos sensores que quedarían sin usar en este proyecto.

Integración de módulos individuales

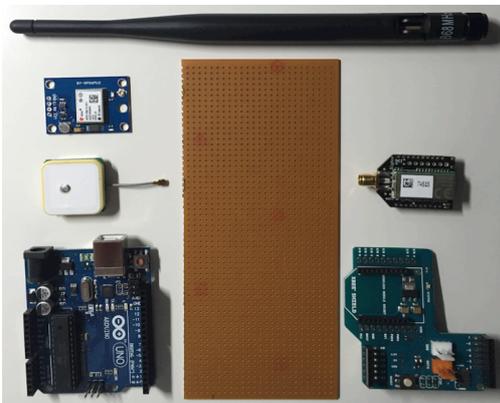


Figura 6. Módulos independientes.

La segunda opción consistiría en adquirir los diferentes módulos por separado, además de una placa controladora y proceder a la integración tanto del hardware como al desarrollo del firmware de control. Esta opción, aunque más laboriosa, permite una mayor flexibilidad a la hora de escoger los diferentes módulos.

Cabe destacar que un factor determinante a la hora de elegir una u otra opción, ha sido el plazo de entrega previsto de los componentes.

En este sentido, tal como se explicaba en el apartado dedicado a la planificación, existía un límite temporal crítico a partir del cual se hacía inviable contar con un determinado proveedor, pues el riesgo de incumplir los plazos planificados, y por tanto de no poder alcanzar los objetivos del proyecto, eran muy elevados.

Es por ello que, estudiadas ambas posibilidades, se comprobó que la primera opción (*SmartEverything*) quedaba automáticamente descartada debido a que el plazo de entrega rondaba el mes y medio (principalmente debido a falta de stock), frente al plazo máximo de dos semanas que suponía la adquisición de los elementos por separado.

En la Figura 7 se representa por tanto, el diagrama de bloques final sobre el que se basa el diseño del SCAN-device. Como se puede observar, está compuesto por una placa controladora principal que se comunica con el resto de módulos, así como con el usuario a través del teclado. En los siguientes apartados se describirá la integración de cada uno de estos módulos, de tal manera que durante la evolución del capítulo se irá visualizando paso a paso cómo se desarrolló el prototipo hasta obtener el producto final.

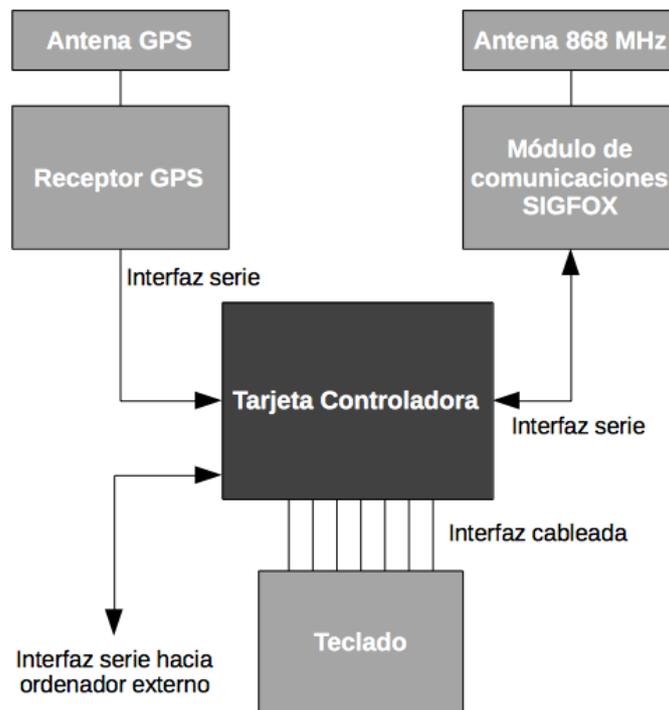


Figura 7. Diagrama de bloques del prototipo hardware desarrollado (SCAN-device). Se muestran los diferentes módulos que lo componen, así como su interfaz con la placa controladora.

3.3 Tarjeta controladora

Dentro de las opciones disponibles que pudiesen ser empleadas como tarjeta controladora de bajo coste y cuya curva de aprendizaje fuese lo suficientemente rápida como para cumplir los plazos planificados, se han valorado las placas Raspberry PI 2 (Model B) [16] y Arduino UNO [17]. Dentro de cada una de estas marcas, se encuentran varios modelos con especificaciones y tamaños ligeramente diferentes, y es por ello que se ha optado por acotar los modelos a estudiar a aquellos que permitiesen encontrar módulos de comunicaciones SIGFOX y GPS compatibles, y fáciles de integrar en ellas. Por otro lado, aunque las dos plataformas están ligadas al concepto de *hardware/software* libre, conceptualmente son totalmente diferentes como veremos a continuación.

Raspberry PI

Se trata básicamente de un ordenador embebido en una pequeña placa de aproximadamente 86x54 mm disponiendo, por tanto, de salida de video/audio así como que de conectividad *Ethernet* y periféricos (teclado y ratón). A continuación se muestran sus características principales:

- CPU: 900 MHz quad-core ARM Cortex A7
- Memoria SDRAM: 1GB
- Puertos USB: 4
- Almacenamiento: MicroSD
- Conectividad de red: 10/100 Ethernet
- Puertos E/S: 40 pines GPIO
- Consumo: 800 mA
- Tensión de alimentación: 5V
- Sistema operativo: GNU/Linux

Arduino UNO

Consiste básicamente en una placa de desarrollo que incorpora un microcontrolador Atmel, además de un entorno IDE donde programar y cargar el firmware. La integración con sensores externos, módulos, pulsadores, etc. es extremadamente sencilla. Sus especificaciones principales son las siguientes:

- Microcontrolador: ATmega328P
- Memoria Flash: 32 KB
- SRAM: 2KB
- EEPROM: 1 KB
- Pines de E/S: 14
- Entradas analógicas: 6
- Velocidad de reloj: 16MHz
- Tensión de alimentación: 5V
- Consumo: dependiendo del diseño (típico: 50 mA min)

El lenguaje de programación del microcontrolador está basado en *Wiring* [18], un *framework* de programación *open-source* que permite el rápido desarrollo de firmware de forma simple. Su sintaxis se asemeja mucho a C/C++, con lo que la curva de aprendizaje es realmente rápida.

Como se puede comprobar, aunque en el mercado encontramos módulos SIGFOX y GPS que se pueden conectar a cada una de estas placas, conceptualmente la integración sería bastante diferente. Mientras que en el primer caso necesitamos tener ejecutándose un sistema operativo completo y la mayor parte de sus recursos estarían desaprovechados (video, audio, memoria, etc.), la placa Arduino nos permitiría tener un diseño mucho más optimizado, activando solo aquellas funciones que son necesarias para interconectar los diferentes componentes.

Por otro lado, aunque en las especificaciones iniciales no existe ninguna restricción en cuanto al consumo energético, no cabe duda que la eficiencia de la placa Arduino, supera en más de diez veces a la Raspberry PI, factor importante a tener en cuenta en el caso de que queramos alimentar el prototipo mediante baterías.

Finalmente, realizando un análisis previo de las especificaciones necesarias y basándonos en lo anteriormente expuesto, se ha optado por emplear la placa Arduino UNO como controladora del prototipo. En la siguiente tabla se resumen las principales especificaciones analizadas que validan el uso de esta placa.

Tabla 5. *Relación de requisitos básicos de diseño necesarios y capacidades de la placa Arduino Uno*

FUNCIONALIDAD	REQUISITO	DISPONIBLE EN ARDUINO
Capacidad de controlar 5 LEDs	5 salidas	Total disponibles: 14 Usadas: 5
Capacidad de lectura de 4 pulsadores	4 entradas	Total disponibles: 9 Usadas: 4
Puerto serie RS232 módulo GPS	1 entrada Rx 1 salida Tx	Puerto serie emulado por software usando pines de E/S: Total disponibles: 5 Usados: 2
Puerto serie RS232 módulo SIGFOX	1 entrada Rx 1 salida Tx	UART puerto nativo RS232
Almacenamiento interno mínimo	576 bytes ¹	1 KB EEPROM

3.4 Diseño e integración del teclado

Como primer paso en el desarrollo del SCAN-device, se optó por diseñar e integrar el teclado para poder empezar a interactuar con el microcontrolador. De esta forma, cualquier actualización y progreso en el desarrollo del firmware podría ser probado directamente con el interfaz de usuario final. Cabe destacar que, aunque aquí se haga referencia únicamente al teclado, realmente éste comprende tanto los pulsadores para comandar al microcontrolador, como los indicadores LEDs que devuelven el estado del prototipo.

¹ Cálculo estimado para almacenar una jornada de 8 horas de medidas, transmitiendo a la máxima capacidad permitida (6 mensajes/hora de una longitud de 12 bytes cada uno)

Estos elementos se ensamblaron en una placa de desarrollo perforada estándar de dimensiones 110 x 53 mm distribuidos de tal manera que permitiesen la integración posterior del módulo de comunicación SIGFOX y GPS. La interconexión con la placa controladora se realizó mediante la conexión directa de pines hacia los puertos de expansión del Arduino. En la Tabla 6 se resumen los pulsadores y LEDs empleados, así como el pin de la placa a la que fueron conectados.

Tabla 6. Relación de LEDs y pulsadores empleados en el diseño del teclado y pines de la placa controladora a la que han sido conectados

PIN	LED/PULSADOR	DESCRIPCIÓN
4	Pulsador: Erase EEPROM	Permite borrar la EEPROM cuando se ha llenado la misma. Como medida de seguridad, y para evitar borrar los datos almacenados de forma accidental, este pulsador deber estar presionado durante 5 segundos para proceder al borrado
5	Pulsador: Select Power	Permite seleccionar la potencia a la que vamos a realizar la transmisión
6	Pulsador: Transmit	Permite realizar una transmisión hacia la red SIGFOX. Si cuando se activa, también está pulsado el botón "Remote", la transmisión se realizará con petición de ACK (Acknowledgment)
7	Pulsador: Remote	Habilita la escucha en el puerto serie. Si se conecta un PC a dicho puerto serie, se pueden enviar comandos para descargar los datos almacenados, comprobar los registros de la EEPROM o reprogramar los diferentes valores de potencias a transmitir
8	LED: TX Low (Rojo)	Indica que se ha seleccionado transmitir a la potencia más baja
9	LED: TX Med (Amarillo)	Indica que se ha seleccionado transmitir a la potencia media
10	LED: TX Hig (Verde)	Indica que se ha seleccionado transmitir a la potencia más alta
11	LED: TX On (Amarillo)	Se enciende cuando se ha iniciado una transmisión. Cuando la transmisión ha terminado, se apaga
12	LED: GPS Lock (Amarillo)	Parpadea cuando el GPS no tiene posición válida. Queda fijo cuando el GPS tiene posición válida

Cabe destacar que en la mayoría de los LEDs se ha implementado una doble funcionalidad, para indicar algún otro estado no contemplado en la tabla anterior y que sea relevante para el usuario. En este sentido, los siguientes estados también se pueden dar.

EEPROM llena

La EEPROM del microcontrolador ATmega328 permite almacenar hasta 1KB de datos. Como se verá en apartados posteriores, y teniendo en cuenta la cantidad de bytes que

se almacenan por transmisión, así como los bytes que se han dejado para futuros usos, esta capacidad permite almacenar hasta 90 registros de transmisión. De esta forma, cuando se ha alcanzado ese número de transmisiones, los LEDs “TX Low” y “TX Hig” se iluminarán simultáneamente, mientras que el LED “Tx Med” permanecerá apagado. En estas condiciones, no se podrá hacer ninguna transmisión más hasta que no se proceda al borrado de la EEPROM.

EEPROM borrada satisfactoriamente

Para poder borrar la memoria interna, se debe presionar, durante al menos 5 segundos, el pulsador “Erase EEPROM”. Cuando esto sucede, el registro contador que mantiene el número de transmisiones realizadas se pone a cero y, en ese momento, los LEDs “TX Low”, “TX Med” y “TX Hig” se apagarán sea cual sea su estado anterior. Esta situación indica que se puede volver a transmitir en condiciones normales.

Error en la transmisión

Cuando presionamos el pulsador “Transmit”, se inicia una secuencia que comprende los siguientes pasos: preparar los datos que van a ser transmitidos, encender el módulo de comunicaciones SIGFOX, configurar su potencia de transmisión en función del valor seleccionado por el usuario, proceder a la transmisión y, si todo ha ido bien, almacenar los datos transmitidos en la EEPROM y actualizar los contadores. Por otro lado, cada vez que se interactúa con el módulo de comunicaciones, se procede a verificar que el comando se ha ejecutado correctamente y en caso de no haya sido así -por ejemplo, ha habido algún error al encender el módulo, al transmitir, etc.-, los LEDs “TX Low”, “TX Med” y “TX Hig” se iluminarán simultáneamente. Este estado permite informar al usuario de cualquier error que se haya producido a la hora de transmitir el mensaje.

SCAN-device en modo remoto

Si queremos que el prototipo acepte comandos por el puerto serie que permitan interactuar con él, se debe presionar el pulsador “Remote”. En este estado, el dispositivo, quedará pendiente de los comandos que reciba por el puerto serie. La indicación de que el dispositivo se encuentra en modo remoto se realiza a través del LED “TX On”, que quedará parpadeando. Para volver al modo normal de funcionamiento basta con volver a presionar el pulsador “Remote”.

ACK recibido correctamente

El dispositivo permite dos modos diferentes de transmisión. El modo de transmisión nominal, en el que se envía el mensaje sin esperar respuesta, y el modo de transmisión con ACK. Cuando se realiza este último modo de transmisión (para ello hay que pulsar el pulsador “Transmit”, manteniendo presionado simultáneamente el botón “Remote”), una vez transmitido el mensaje, el módulo de comunicación queda a la espera de recibir la respuesta por parte del *backend*, confirmando la recepción correcta del mensaje. Para indicar que se ha recibido correctamente el ACK, el LED “Tx On” parpadeará durante 10 segundos con una frecuencia de 2 Hz.

En el Anexo A.5 se muestra el esquema de interconexión de los elementos que forman el teclado con la placa Arduino. Como se puede ver, se han conectado resistencias de

470 ohms en serie con los LEDs, lo que permiten tener una corriente de polarización lo suficientemente alta como para mantener un brillo adecuado en los mismos (aproximadamente 10 mA) y al mismo tiempo cumplir con las especificaciones de corriente máxima tolerable en las salidas digitales del Arduino (20 mA). Por otro lado, los pulsadores serán detectados por transición a nivel alto, habiéndose conectado resistencias de valor 10 Kohm que fuerzan un estado alto cuando el pulsador se encuentra en reposo.

Finalmente, la siguiente figura muestra la integración final del teclado sobre la placa perforada, donde se puede observar la distribución de los diferentes elementos del teclado, así como del resto de módulos.

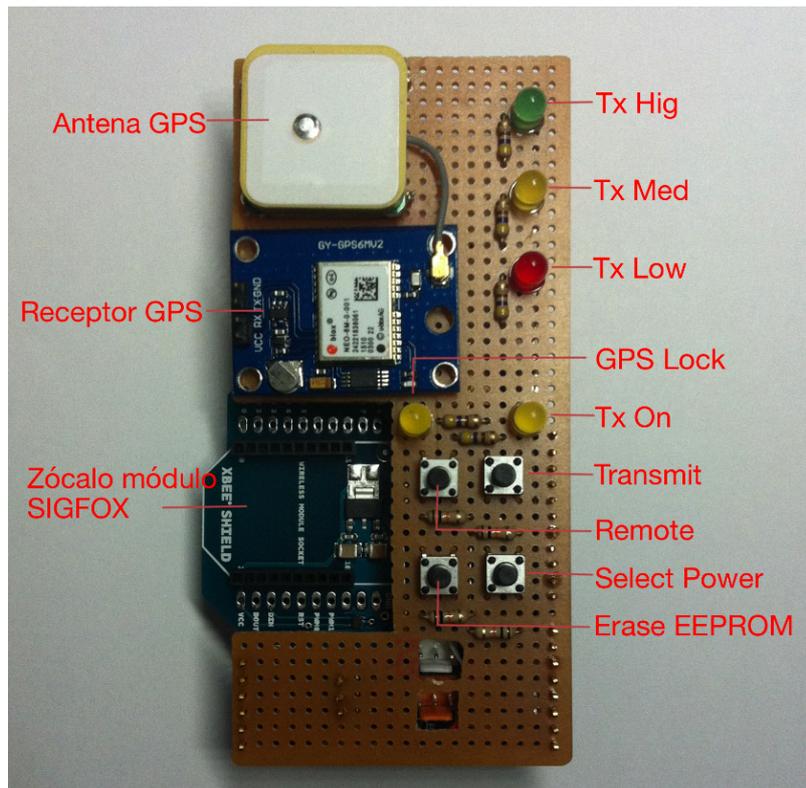


Figura 8. Detalle de la integración final del teclado. Se representan tanto los elementos que lo forman (LEDs y pulsadores) como la distribución del resto de módulos que forman el prototipo.

3.5 Integración del módulo GPS

Una de las primeras cuestiones a decidir para incorporar la capacidad de proporcionar posición a cada una de las transmisiones es la de si emplear un dispositivo interno, integrado en la propia placa del prototipo, o si usar un dispositivo externo. No cabe duda de que ésta última opción es, a priori más cara, y que la mayor parte de las funcionalidades de cualquier GPS comercial quedarían sin utilizar (*waypoints*, velocidad, mapas, etc.). Es por ello que la opción más lógica pasa por intentar encontrar un modelo de GPS que pueda ser integrado con facilidad en la propia placa.

En el mercado se encuentran múltiples módulos GPS de reducido tamaño que pueden ser empleados para tal fin aunque, atendiendo básicamente a cuestiones de dimensiones así como de disponibilidad y precio, finalmente se ha optado por el modelo GY-GPS6MV2 [19], que incorpora un receptor GPS modelo NEO-6M [20] del fabricante Ublox. Las principales características se detallan en la Tabla 7.

Tabla 7. Características principales del receptor GPS NEO-6M [20]

Canales	50
Tiempo para la primera posición	Cold/Warm Start: 27 s Hot start: 1 s
Sensibilidad	Tracking y navegación: -161 dBm Readquisición: -160 dBm Cold Start (sin asistencia): -147 dBm Hot Start: -156 dBm
Precisión de posición horizontal	GPS: 2.5 m SBAS: 2.0 m
Interfaces/Protocolos	UART, USB, SPI/ NMEA, UBX, RTCM
Alimentación	3.0 V max
Consumo	40 mA típico a máximas prestaciones
Temperatura operacional	-40 a 85 °C

Este receptor GPS se encuentra ensamblado en una pequeña PCB de 26 x 35 mm que permite una rápida integración sobre el prototipo final. Los pines de conexión de dicha placa son VCC, GND, Tx y Rx, además del conector de RF de tipo MMBX, donde irá acoplada la antena de tipo cerámica. El protocolo por defecto configurado en la placa es NMEA (*National Marine Electronics Association*) [21], a 9600 baudios, 8 bits de datos, sin paridad y con 1 bit de stop.

La comunicación se realizará por tanto a través de la UART del mismo, conectando los pines de recepción y transmisión del GPS, a los pines de transmisión y recepción correspondientes de la placa Arduino. Cabe destacar que, como se verá en el siguiente apartado, los pines nativos de la UART del Arduino se usarán para la comunicación con el módulo de comunicaciones SIGFOX, además de la comunicación con el PC de usuario, con lo que los pines asignados al puerto serie que conecta el Arduino al GPS se realizará mediante emulación software. Para ello se hará uso de la librería estándar "SoftwareSerial", incluida ya en el propio IDE de Arduino. Esta librería permite la comunicación serie mediante otros pines digitales para replicar la funcionalidad del puerto nativo hardware (pines 0 y 1).

En el esquema de conexiones mostrado en el Anexo A.5 se detalla el conexionado entre el módulo GPS y la placa Arduino, donde se pueden ver los pines que han sido asignados para la comunicación entre ellos.

Como se ha mencionado anteriormente, la comunicación entre el módulo GPS y la placa controladora se realizará mediante el protocolo NMEA, de tal manera que, nada más proporcionar alimentación al receptor GPS, éste comenzará a enviar tramas. A continuación entraremos en detalle sobre dicho protocolo y en cómo se extraerá la información de interés de dichas tramas.

El protocolo NMEA es un estándar creado y mantenido por la *National Marine Electronics Association*. Existen varias versiones (NMEA 0183 v2.0, NMEA 3.01, etc.) siendo implementado en el GPS elegido el protocolo NMEA 0183 V2.3, compatible con

la versión V3.0. A modo de resumen, podríamos decir que es un protocolo que permite el intercambio de datos entre diferentes equipos electrónicos, orientado inicialmente a equipos marinos, pero que es ampliamente aceptado e implementado por diferentes fabricantes.

Los mensajes transmitidos bajo NMEA siguen un patrón repetitivo de tramas en formato ASCII y, por tanto, pueden ser leídas directamente por un usuario. Cada trama comienza con el carácter "\$", y finaliza con un carácter de retorno de carro y alimentación de línea (CRLF). Hay diferentes tipos de tramas en función del valor contenido en el primer campo identificativo (entre el primer carácter \$ y la primera coma que separa el siguiente campo). Cada campo de datos dentro de una trama determinada viene separado por comas.

La siguiente figura muestra un ejemplo de la salida del GPS que se ha integrado en el prototipo, una vez implementada en el firmware la rutina para poder leer las tramas NMEA recibidas por el puerto serie.

```
$GPVTG,,T,,M,0.039,N,0.072,K,D*29
$GPGGA,211142.00,2806.82402,N,01525.50622,W,2,08,0.99,28.8,M,36.1,M,,0000*76
$GPGSA,A,3,31,26,21,16,27,18,10,29,,,,,1.69,0.99,1.36*0C
$GPGSV,4,1,14,04,60,025,36,08,07,250,,10,05,154,34,16,38,323,32*7D
$GPGSV,4,2,14,18,13,120,36,20,06,066,,21,58,074,36,23,05,296,*72
$GPGSV,4,3,14,25,04,106,26,26,59,357,39,27,33,264,40,29,12,049,21*7F
$GPGSV,4,4,14,31,61,178,43,33,57,180,39*71
$GPGLL,2806.82402,N,01525.50622,W,211142.00,A,D*7B
$GPRMC,211143.00,A,2806.82405,N,01525.50621,W,0.028,,150316,,D*6D
```

Figura 9. Ejemplo de tramas NMEA enviadas por el GPS. En este caso, el GPS ya ha calculado posición válida.

De todos los tipos de tramas disponibles, en la implementación se hará uso de la trama \$GPGLL (*Geographic Latitude and Longitude*). Los diferentes campos que componen esta trama son los que se muestran en la Tabla 8.

Tabla 8. Detalle de los diferentes campos que forman una trama del tipo \$GPGLL

Ejemplo: \$GPGLL,2806.82391,N,01525.50379,W,210535.00,A,A*7D							
\$GPGLL	2806.82391	N	01525.50379	W	210535.00	A	A*7D
Tipo de trama: GPGLL	Latitud: 28 deg 06.82391 min	Norte	Longitud: 15 deg 25.50379 min	Oeste	Hora: 21:05:35 UTC	Dato válido: A No válido: V	Checksum

Aunque los detalles de la rutina para extraer, de entre todas las tramas, solamente las del tipo \$GPGLL, pueden ser consultados en el código fuente proporcionado (ver Anexo A.1), cabe detenerse al menos en las diferentes acciones que realiza el

software. Esto ayudará a la interpretación correcta del código para aquellos lectores que precisen entrar en los detalles del mismo.

De este modo, las fases para procesar las tramas NMEA recibidas del GPS son las siguientes: por un lado, en el bucle principal se irá leyendo cada una de las tramas recibidas y se comparará con aquella que resulta de interés (\$GPGLL). Si se encuentra, se pasa a procesar dicha trama donde se extraerán los diferentes campos (en el caso de que tenga datos válidos) que son: la latitud, longitud así como si es norte, sur, este u oeste. Los pasos resumidos son:

Rellena una trama

- Comprueba si hay datos para ser leídos en el puerto serie
- Mientras no se haya completado la cabecera va rellenando con nuevos caracteres. Cuando se rellena la cabecera se compara con la que se está buscando (\$GPGLL).
- Si es la cabecera buscada, rellena el resto de la trama.
- Cuando se completa la trama, se llama a la función que la procesa y extrae los datos.

Procesar una trama

- Primero, comprueba si la posición es válida (carácter "A"). En este caso, actualiza la variable *gpsLocked* a *true*.
- En ese caso va recorriendo cada uno de los campos en el siguiente orden para extraer el resto de datos: latitud, N/S, longitud, E/W.
- En caso de que el GPS aún no esté dando posiciones válidas, la variable *gpsLocked* se mantiene a *false*, para indicar al resto del programa que no se puede realizar la transmisión y que el LED de indicación de GPS Locked debe permanecer intermitente.

3.6 Integración del módulo de comunicaciones SIGFOX

El módulo de comunicaciones elegido para ser integrado en el SCAN-device es de la empresa Libelium, y está basado en el chip TD1207 de TELCOM Design [22]. Dentro de los pocos proveedores que proporcionaban un módulo independiente certificado por SIGFOX para ser usado directamente con su red y que fuese directamente conectable a una placa de control Arduino, el proporcionado por Libelium permitía obtener un plazo de entrega de tan solo 2 semanas, lo que entraba dentro de los márgenes definidos en la planificación.

Hay que destacar que, para facilitar la integración se adquirió, junto con el módulo, la placa Xbee Shield recomendada por el fabricante. De esta manera, el módulo podía ser insertado directamente, a través de dicha placa sobre el Arduino, sin necesidad de cableado exterior. En la Figura 10 se muestra el perfil del diseño completo, donde se pueden observar la posición de la placa Xbee Shield entre la placa Arduino y la placa del teclado.

Hay que destacar que el módulo, junto con la antena suministrada, proporcionan una certificación SIGFOX del tipo 0U, es decir, capaz de transmitir a la máxima potencia

permitida por la regulación y por tanto lograr el máximo alcance. Las características principales del módulo se presentan en la Tabla 9.

Tabla 9. Características principales del módulo de comunicación SIGFOX empleado en el prototipo

Frecuencia	ISM 868 MHz
Tx Power	14 dBm
Consumo Tx	49 mA @ 14 dBm
Consumo Rx	13 mA
Sensibilidad	-126 dBm
Limitación ETSI	140 paquetes de 12 bytes por módulo/día
Certificación SIGFOX	Clase 0U (máxima potencia TX)
Antena	4.5 dBi $\lambda/2$
Alimentación	3.3 V

Por otro lado, la comunicación con la placa Arduino se realizará empleando su puerto serie nativo. Este puerto corresponde a los pines 0 (Rx) y 1 (Tx). Puesto que dicho puerto es también compartido para la comunicación con un ordenador de usuario -para enviar comandos y recibir el estado cuando el prototipo se encuentra en modo remoto- se hace necesario poder aislar dichos pines entre el módulo de comunicaciones SIGFOX y el puerto serie del ordenador.

Esta tarea se realiza mediante un *microswitch* situado en la placa Xbee Shield, de tal manera que cuando se encuentra en posición “USB”, el ordenador será quien pueda comunicarse con la placa Arduino, mientras que cuando se encuentra en posición “XBEE”, será el módulo SIGFOX quien se pueda comunicar con él.

Es, por tanto, de suma importancia que dicho *microswitch* se encuentre en la posición correcta en función del modo de funcionamiento del SCAN-device:

- En modo normal: el *microswitch* debe estar en la posición “XBEE”.
- En modo remoto: el *microswitch* debe estar en la posición “USB”. En este modo se puede proceder a presionar el pulsador “Remote” para comenzar la comunicación entre el ordenador y el prototipo.

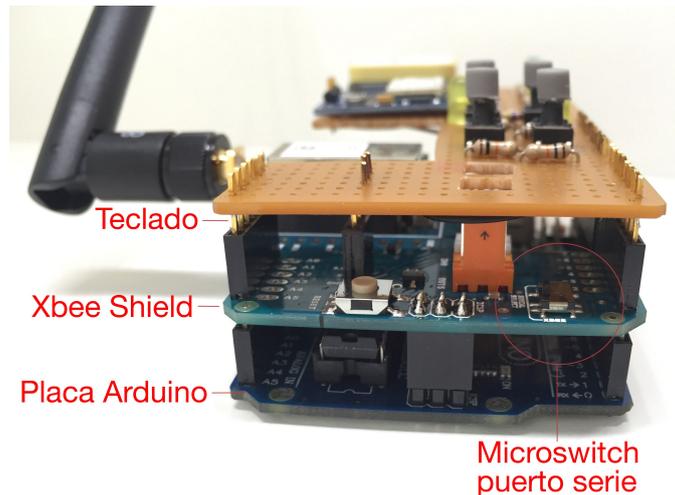


Figura 10. Detalle de las placas que forman el prototipo y del microswitch de configuración del puerto serie en la placa Xbee Shield.

Por otro lado, para la comunicación con el módulo de comunicaciones, se ha hecho uso de las librerías proporcionadas por el propio fabricante. Estas librerías facilitan la tarea de enviar y recibir comandos AT al módulo TD1207, mediante el uso de diferentes funciones. Aunque aquellos lectores que deseen profundizar en el código implementado que controla el módulo de comunicaciones SIGFOX pueden consultar directamente el repositorio proporcionado (ver Anexo A.1), cabe reflejar aquí una breve reseña de las tareas realizadas por el programa para facilitar la comprensión del mismo.

Así pues, el bucle principal consta de los siguientes pasos:

- Comprueba si se ha pulsado el botón TX On (transmitir), si el GPS está en estado Lock (hay posición válida) y si la EEPROM no está llena.
- Si es así prepara el mensaje que se ha de enviar a la red SIGFOX.
- Enciende el módulo y configura la potencia a transmitir en función del valor que haya seleccionado el usuario.
- Envía el mensaje, quedando a la espera de respuesta en caso de que la transmisión se haya realizado con ACK. Si no ha habido ningún error al transmitir el mensaje, incrementa los contadores y guarda el mensaje transmitido en la EEPROM.
- Si la EEPROM está llena, activa los *flags* que indican dicha circunstancia al usuario e impide volver a transmitir.

3.7 Formato del mensaje transmitido

En este apartado se entrará en el detalle de cómo se ha decidido diseñar el mensaje que será transmitido a la red SIGFOX. Tal como se ha mencionado en capítulos previos, la máxima cantidad de información que se puede transmitir en un mensaje es de 12 bytes, con un máximo de 140 mensajes por día y por nodo, siendo el formato de esos 12 bytes totalmente libre. Por otro lado, la información que se deberá incluir en dicho mensaje es la siguiente:

- Número identificador de la transmisión (TXID): éste es un número único y secuencial que identifica inequívocamente cada una de las transmisiones que se realizan. Se podría entender como una especie de número de serie que está asociado a cada una de las transmisiones realizadas durante la campaña de medidas y, por tanto, no se puede repetir durante el transcurso de la misma.
- Nivel de potencia transmitida: indica la potencia en dBm a la que se ha realizado la transmisión. Como ya se ha comentado, cada una de las transmisiones puede ser efectuada hasta con tres potencias diferentes. El valor de dicha potencia debe ser incluido en el mensaje para poder determinar en qué condiciones se realizó dicha transmisión.
- Latitud del punto de transmisión: determina la latitud geográfica en la que se ha realizado la transmisión. El formato será en grados y minutos, codificados de la forma que se describirá en los siguientes párrafos.
- Longitud del punto de transmisión: determina la longitud geográfica en la que se ha realizado la transmisión. El formato será en grados y minutos, codificados de la forma que se describirá en los siguientes párrafos.

A continuación describiremos cuáles son los valores límites de cada uno de estos valores y que permitirán asignar la mejor codificación posible para adaptar la longitud total del mensaje dentro de los 12 bytes máximos permitidos.

En cuanto al identificador de transmisión, se ha decidido emplear un valor entero sin signo de 2 bytes que permitirán codificar hasta 65536 valores diferentes. Se ha estimado oportuno emplear 2 bytes para dicha codificación, pues el número máximo de transmisiones permitidas dentro de una misma campaña de medidas alcanzaría un número lo suficientemente elevado como para que no se repitiesen los identificadores de transmisión. De hecho, si cumpliésemos con el número máximo de transmisiones diarias que la regulación permite para esta banda de trabajo (140 mensajes máximos), se observa que se tendría que estar transmitiendo cada 10 minutos ininterrumpidamente durante aproximadamente 468 días hasta alcanzar el límite.

Por otro lado, el rango de potencias que el módulo de comunicaciones permite configurar van desde 0 hasta 14 dBm. Este rango se codificará usando los 4 bits menos significativos del byte asignado, dejando los 4 bits más significativos para posibles ampliaciones futuras del formato.

En lo que respecta a la latitud y longitud, se debe encontrar un método lo más eficiente posible para poder transmitir la misma usando el mínimo número de bytes posibles, y es que hay que tener en cuenta que, codificarlas usando directamente una representación ASCII no sería viable, puesto que necesitaríamos más de un mensaje para poder incluir toda la precisión necesaria.

En este sentido, una forma óptima de codificar la latitud y la longitud consiste en emplear un entero con signo de tipo Long y componer el valor de la variable uniendo los grados con la parte entera de los minutos, seguidos de la parte decimal de los minutos sin incorporar el separador decimal. De esta forma, teniendo en cuenta que los valores máximos y mínimos de una variable de tipo Long con con signo (4 bytes) en Arduino puede almacenar unos valores que van desde 2147483647 hasta -2147483648, vemos que pueden almacenar perfectamente todo el rango de latitudes

y longitudes posibles (-900000000 hasta 900000000 para la latitud; -1800000000 hasta 1800000000 para la longitud).

La siguiente tabla muestra un resumen de cómo queda distribuida finalmente la información en el mensaje transmitido, siendo **11 el número total de bytes que serán enviados por mensaje**.

Tabla 10. Formato del mensaje transmitido por el prototipo a la red SIGFOX

TIPO	POSICIÓN BYTE	DESCRIPCIÓN
Unsigned Int	0	TXID. Usado para la identificación de la transmisión. Valores posibles entre 0 y 65535
	1	
Byte	2	Potencia transmitida. Almacena el valor de potencia con que se ha realizado la transmisión en los 4 bits menos significativos
Long	3	Latitud. Cuatro bytes que representan un entero con signo. El rango máximo es: 900000000 a -900000000. Los últimos 7 dígitos corresponden a los minutos con 5 decimales. Ejemplo: 754334543 representa Lat 75° 43.34543' N
	4	
	5	
	6	
Long	7	Longitud. Cuatro bytes que representan un entero con signo. El rango máximo es: 1800000000 a -1800000000. Los últimos 7 dígitos corresponden a los minutos con 5 decimales. Ejemplo: -153414367 representa Lon 15° 34.14367' W
	8	
	9	
	10	

3.8 Mapa de la memoria EEPROM

La persistencia de datos se consigue mediante el uso de la memoria EEPROM del propio microcontrolador, que permite mantener información aún cuando la alimentación del dispositivo ha sido interrumpida. Tiene una capacidad de almacenar hasta 1KB de datos y necesita ser mapeada para poder realizar un correcto acceso a la dirección de memoria de la misma a la hora de escribir o recuperar datos.

Los datos que se guardan en dicha memoria son:

- TXID. Se trata del identificador de transmisión ya descrito anteriormente y ocupa los 2 primeros bytes de la memoria. Tiene sentido mantener el valor de

- esta variable, pues aunque el equipo se apague, cualquier nueva transmisión posterior debe continuar a partir del último valor TXID almacenado.
- Número de registros guardados en la EEPROM. Este valor ocupa el siguiente byte en la memoria y mantiene el número actual de registros de transmisión que están guardados en la EEPROM. Es de suma importancia, pues permite direccionar correctamente la memoria a la hora de guardar nuevas transmisiones o recuperar los datos.
 - Valor de potencia máxima. Ocupa 1 byte, y guarda el valor programado para la potencia máxima a transmitir. Puesto que los valores de las tres potencias disponibles pueden ser programados mediante comandos externos, en esta posición de memoria, se almacena dicho valor.
 - Valor de potencia media. Ocupa 1 byte, y guarda el valor programado para la potencia media a transmitir.
 - Valor de potencia mínima. Ocupa 1 byte, y guarda el valor programado para la potencia mínima a transmitir.
 - Registros de datos de transmisión. Cada registro de transmisión guardado ocupa 11 bytes y ocupan las direcciones de memoria desde la dirección 6 hasta la 995.

Cabe destacar que, teniendo en cuenta el tamaño de la memoria EEPROM, así como los datos anteriormente expuestos que también se almacenan en ella, el número máximo de transmisiones guardadas antes de proceder a la descarga de datos, es de 90. Esto permite tener 28 bytes de reserva ($1024-6-11*90$) para futuros usos.

A modo de ejemplo, considerando una jornada de campaña de medidas de 8 horas y, teniendo en cuenta que según la regulación ETSI se pueden transmitir como máximo 6 mensajes a la hora, el llenado de la EEPROM se produciría en casi dos jornadas (unas 15 horas), con lo que es importante proceder a la descarga de los datos y posterior borrado de la memoria para evitar alcanzar la capacidad máxima en medio de una campaña de medidas.

Por otro lado, a la hora de guardar el registro en la memoria, una vez que se ha comprobado que la transmisión se ha realizado correctamente, se debe incluir otra información adicional necesaria para el posterior procesado. Esta información corresponde a si la transmisión se ha realizado esperando confirmación de ACK o no, y si realmente se recibió dicho ACK. Esta información se añadirá a los bits 5 y 6 respectivamente, del byte asignado a la potencia transmitida.

La siguiente tabla resume cómo queda el mapa final de memoria.

Tabla 11. Mapa de direcciones de memoria de la EEPROM del SCAN-device

DIRECCIÓN	BYTES	TIPO	DESCRIPCIÓN
0	0-1	int	Entero de 2 bytes que almacena el número disponible a usar para incluir en la identificación de la transmisión (TXID)
2	2	Byte	Byte que almacena el numero N actual de registros enviados y que se encuentran grabados en la EEPROM
3	3	Byte	Byte que almacena el valor de potencia correspondiente a la potencia máxima (LED verde)
4	4	Byte	Byte que almacena el valor de potencia correspondiente a la potencia media (LED amarillo)
5	5	Byte	Byte que almacena el valor de potencia correspondiente a la potencia mínima (LED rojo)
6	6-16	Struct sigfoxData	Registro #1 de datos. Corresponde a una estructura de 11 bytes que almacena una transmisión
17	17-27	Struct sigfoxData	Registro #2 de datos. Corresponde a una estructura de 11 bytes que almacena una transmisión
6+11(N-1)	6+11(N-1)- 16+11(N-1) <i>siendo N el número de registros</i>	...	Registro #N de datos. Corresponde a una estructura de 11 bytes que almacena una transmisión
985	985-995	Struct sigfoxData	Registro #90 de datos. Corresponde a una estructura de 11 bytes que almacena una transmisión

3.9 Comandos externos

No tendría sentido almacenar los datos en el prototipo si posteriormente no es posible recuperar los mismos. Para poder realizar dicha tarea de recuperación de datos para su posterior análisis, se ha procedido a diseñar una interfaz de comunicación con el “exterior” que permita recibir comandos y poder interactuar así con él. Es lo que anteriormente se ha definido como el modo “Remoto”.

Cabe mencionar que, para dotar de mayor flexibilidad al diseño, y con el objeto de poder configurar las diferentes potencias de transmisión que están asociadas a los diferentes LEDs (Hig, Med, Low), se han implementado también varios comandos que permiten modificar dichos valores.

A continuación se detallarán los 5 comandos disponibles que permiten interactuar con el SCAN-device. Estos comandos han de ser enviados a través del puerto serie (usando configuración 9600-N-8-1), conectando directamente el prototipo a un ordenador y empleando para la comunicación cualquier programa que actúe de terminal. Como se verá en el capítulo dedicado al SCAN-tool, en el software de análisis desarrollado se ha implementado una interfaz que permite comunicarse directamente con el SCAN-device para la descarga de datos en el propio programa sin necesidad de usar un software externo.

Comando \$GP\$

Este comando permite obtener los datos del TXID actual, el número de registros actualmente almacenados en la EEPROM, así como el valor de potencia configurado para cada uno de las tres posibles potencias a transmitir (Hig, Med, Low). Al enviar la cadena de caracteres \$GP\$ por el puerto serie, el SCAN-device responderá con una línea indicando cada uno de los parámetros anteriores separados por comas, y terminará la transmisión con una línea que contiene el código de control "EOF", tal como se puede apreciar en la Figura 11.

```
$GP$  
231,55,14,7,0  
EOF
```

Figura 11. Respuesta del SCAN-device a un comando \$GP\$. La respuesta presenta el TXID actual (231), el número de registros de transmisión almacenados en la EEPROM (55) y los valores de potencia asignados a los LEDs Hig, Med y Low (14,7,0) en dBm.

Comando \$GD\$

Al enviar esta cadena, el prototipo responderá devolviendo todos los registros de transmisión almacenados en la EEPROM. Se presentarán uno por línea y terminará el volcado de datos con un "EOF". Cada línea contendrá por tanto un registro, incluyendo los campos TXID, potencia de transmisión, latitud (formato Long), longitud (formato Long). En el segundo campo, a la potencia de transmisión (4 bits menos significativos), se le combina, mediante una operación OR, los bits 5 y 6 para indicar si se realizó la transmisión con ACK y si el ACK fue recibido correctamente.

```
$GD$  
176,14,275974068,-152843855  
177,14,275973800,-152529584  
178,14,275685131,-152543223  
179,14,275509199,-152632103  
180,14,275674388,-152826460  
181,14,275720860,-153084367  
182,14,275765651,-153213048  
183,14,275811694,-153391972  
184,14,275756414,-153564270  
185,14,275703306,-153674252  
186,14,275623068,-153481656  
.   
.   
.   
.   
230,14,280664535,-152538046  
EOF
```

Figura 12. Respuesta del SCAN-device a un comando \$GD\$. La respuesta presenta cada uno de los registros almacenados en una línea.

Comando \$HL:XX\$

Este comando permite configurar la potencia asociada al LED Hig. El campo XX del comando puede tomar un valor del 00 al 14, indicando éste la potencia deseada en dBm que se quiere programar para la transmisión asociada a ese LED. Si el rango no está entre 00 y 14, el comando se ignora. El prototipo no devuelve ninguna respuesta a este comando, por tanto, si se quiere verificar que la potencia ha sido correctamente programada, se puede interrogar al SCAN-device con un \$GP\$.

```
$HL:12$  
$GP$  
231,55,12,7,0  
EOF
```

Figura 13. Respuesta del SCAN-device a un comando \$HL:12\$. Inmediatamente después de ejecutarlo, se interroga con un \$GP\$ para verificar que efectivamente el valor del LED Tx Hig ha cambiado al valor 12 dBm.

Comando \$ML:XX\$

Este comando permite configurar la potencia asociada al LED Med. El campo XX del comando puede tomar un valor del 00 al 14, indicando éste la potencia deseada en dBm que se quiere programar para la transmisión asociada a ese LED. Si el rango no está entre 00 y 14, el comando se ignora. El prototipo no devuelve ninguna respuesta a este comando, por tanto, si se quiere verificar que la potencia ha sido correctamente programada, se puede interrogar al SCAN-device con un \$GP\$.

```
$ML:05$  
$GP$  
231,55,12,5,0  
EOF
```

Figura 14. Respuesta del SCAN-device a un comando \$ML:05\$. Inmediatamente después de ejecutarlo, se interroga con un \$GP\$ para verificar que efectivamente el valor del LED Tx Med ha cambiado al valor 5 dBm.

Comando \$LL:XX\$

Este comando permite configurar la potencia asociada al LED Low. El campo XX del comando puede tomar un valor del 00 al 14, indicando éste la potencia deseada en dBm que se quiere programar para la transmisión asociada a ese LED. Si el rango no está entre 00 y 14, el comando se ignora. El prototipo no devuelve ninguna respuesta a este comando, por tanto, si se quiere verificar que la potencia ha sido correctamente programada, se puede interrogar al SCAN-device con un \$GP\$.

```
$LL:01$
$GP$
231,55,12,5,1
EOF
```

Figura 15. Respuesta del SCAN-device a un comando \$LL:01\$. Inmediatamente después de ejecutarlo, se interroga con un \$GP\$ para verificar que efectivamente el valor del LED Tx Low ha cambiado al valor 1 dBm.

Finalmente, resulta importante mencionar que, tal como se avanzaba en apartados anteriores, para poder establecer la comunicación con el SCAN-device de forma satisfactoria, hay que seguir los siguientes pasos, debido a que el puerto serie se encuentra compartido con el módulo de comunicaciones:

- Colocar el microswitch “Serial Select” en posición USB.
- Presionar el pulsador “Remote”.

Para aquellos lectores interesados en profundizar en el código del firmware asociado a la interpretación y respuesta de los comandos externos, éste se puede consultar en el repositorio correspondiente (ver Anexo A.1).

3.10 Integración mecánica

Para la integración mecánica del SCAN-device se ha empleado una carcasa envolvente estándar de plástico para prototipos con unas dimensiones exteriores de 130x70x45 mm.

Para alojar correctamente la electrónica en su interior, se ha procedido a realizar tres aberturas en los laterales. Por un lado, en la parte izquierda se realizó un corte vertical de unos 15x10 mm para permitir la conexión de la antena con el módulo de comunicación SIGFOX. Por otro lado, en las partes laterales anterior y posterior, se realizaron perforaciones de 15 mm de diámetro para permitir la conexión de la alimentación, así como para poder actuar sobre el *microswitch* de configuración del puerto serie.

Finalmente, la tapa frontal contiene las aberturas que permiten visualizar los distintos LEDs así como interactuar con los pulsadores.

La figura 16 muestra cómo quedó el ensamblado final, listo para las pruebas funcionales y de campo.



Figura 16. Prototipo final ensamblado

3.11 Pruebas funcionales

En este apartado se describirán las pruebas realizadas para verificar que el prototipo funciona correctamente. El objetivo de las mismas es poder declarar el SCAN-device totalmente operativo y listo para ser usado en la campaña de medidas.

Hay que resaltar que, para poder comprobar que el prototipo transmite correctamente y que los mensajes son recibidos en la red de SIGFOX, como paso previo se necesita dar de alta el módulo de comunicaciones en la propia red. Una vez tramitado el alta con el operador y se tenga acceso al *backend*, habrá que proceder a su configuración.

Se comenzará este apartado pues presentando la configuración mínima que hay que realizar en el *backend* para poder usar correctamente el sistema. Posteriormente, se presentarán las diferentes pruebas funcionales realizadas con sus respectivos resultados.

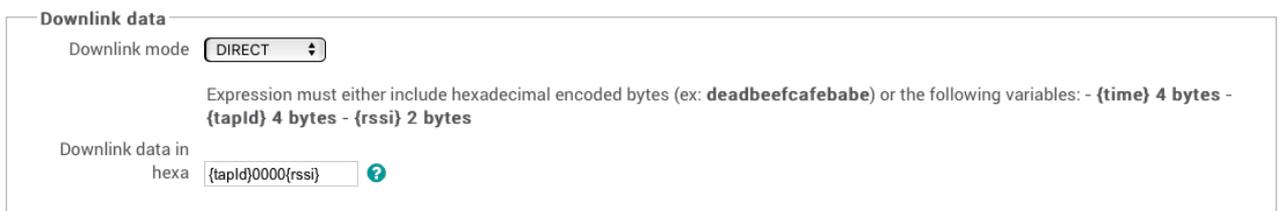
Configuración en el backend de SIGFOX

A lo largo del documento ha sido referido en diferentes ocasiones el término *backend* sin profundizar en su significado dentro del contexto del operador. Entrando en detalles, se entiende por *backend* al panel de control que SIGFOX pone a disposición de aquellos usuarios que han dado de alta algún módulo de comunicaciones en su red. Desde allí, entre otras muchas tareas, se pueden gestionar determinados parámetros que están relacionados con dichos módulos, comprobar el estado de la red o consultar los mensajes recibidos. Es por tanto indispensable, haber dado de alta al menos un módulo de comunicaciones para poder tener acceso a dicho panel.

Una vez se tenga acceso al mismo, hay que configurar las siguientes secciones como mínimo para poder hacer funcionar el sistema completo, tal como se describió en la arquitectura presentada en la Figura 4.

Configuración del Downlink

Esta sección especifica cómo debe ser tratado un mensaje recibido que tenga petición de ACK. Como ya se ha mencionado, un mensaje transmitido con ACK dejará al módulo de comunicaciones esperando un reconocimiento de que el mensaje se ha recibido correctamente. Aquí se especifica si el ACK lo envía directamente el *backend* de SIGFOX o, si por el contrario, éste será reenviado al servidor del usuario para que sea él quien dé el reconocimiento. En la Figura 17 se muestra la configuración correcta.



Downlink data

Downlink mode

Expression must either include hexadecimal encoded bytes (ex: **deadbeefcafebabe**) or the following variables: - **{time}** 4 bytes - **{tapId}** 4 bytes - **{rssi}** 2 bytes

Downlink data in hexa

Figura 17. Configuración de la sección “Downlink data”

Configuración de la decodificación de usuario

Esta sección, aunque no es indispensable, permite realizar una primera decodificación de los datos que ha recibido el *backend* procedentes de un mensaje transmitido por el SCAN-device. Básicamente, especifica qué significa cada byte que hemos enviado en nuestro mensaje y cómo queremos que lo presente en pantalla. La cadena que hay que insertar en el campo “Custom configuration” presentada en la Figura 18 es:

TXID::uint:16:little-endian LEVEL::uint:8 LAT::int:32:little-endian LON::int:32:little-endian



Display type

Type

Custom configuration

Figura 18. Configuración de la sección “Display type”

Configuración del Callback

En esta sección se configura cómo se van a reenviar los mensajes recibidos por las estaciones bases desde el *backend* hacia el SCAN-server. La Figura 19 muestra la configuración requerida.

Callbacks

Type

Channel

Send duplicate

Custom payload
 config TXID::uint:16:little-endian LEVEL::uint:8 LAT::int:32:little-endian LON::int:32:little-endian ?

URL syntax: **http://host/path?id={device}&time={time}&key1={var1}&key2={var2}...**
 Available variables: **device, time, duplicate, snr, station, data, avgSnr, lat, lng, rssi, seqNumber**
 Custom variables: **customData#TXID, customData#LEVEL, customData#LAT, customData#LON**

Url pattern

Use HTTP Method

Send SNI (Server Name Indication) for SSL/TLS connections

Headers

Content type
 time={time}&device={device}&duplicate={duplicate}&station={station}&snr={snr}&rssi={rssi}&data={data}

Body

Figura 19. Configuración de la sección “Callbacks”

Cabe mencionar los siguientes campos, por su importancia en el funcionamiento de la arquitectura:

- **Send duplicate:** debe estar activado para que se reenvíe un mensaje por cada estación base que lo haya recibido, aunque esté duplicado. Esto permite contar el número de estaciones base que han visto una determinada transmisión y analizar por tanto el nivel de redundancia en una localización determinada.
- **Url pattern:** indica la dirección url hacia donde hay que reenviar los mensajes recibidos por el *backend*. Corresponde a la dirección web del SCAN-server y apunta al script que realizará el almacenamiento de los datos.
- **Body:** indica la secuencia de variables que serán enviadas hacia el SCAN-server dentro del cuerpo de cada mensaje (la hora, el identificador del módulo de comunicaciones, si el mensaje es duplicado, la estación que recibió el mensaje, la SNR, el RSSI y finalmente los datos enviados por el prototipo).

Resultado de las pruebas funcionales

Puesto que el objetivo final de las pruebas funcionales es poder verificar que el prototipo desarrollado cumple con los requisitos de diseño iniciales y está listo para ser usado en la campaña de medidas, éstas se centrarán básicamente en realizar transmisiones con diferentes niveles de potencia para comprobar que, por un lado, los mensajes son recibidos correctamente por la red de SIGFOX, y por otro, que estos se retransmiten correctamente hacia el SCAN-server mediante el mecanismo de *callback*.

Cabe mencionar que, mediante las pruebas de transmisión, se verificarán indirectamente otras funcionalidades desarrolladas, como la capacidad de cambiar la potencia transmitida, la capacidad de realizar una transmisión con petición de ACK, así como la incorporación en el mensaje transmitido de la información de posición (y por tanto la verificación de que la comunicación con el GPS es correcta).

En cuanto a las pruebas de comunicación entre el prototipo y un ordenador externo para la descarga de datos y la configuración de parámetros, éstas ya fueron expuestas en el apartado 3.9, mediante la presentación de ejemplos de diferentes comandos enviados al prototipo.

Prueba de transmisión a 14 dBm de potencia

Para esta prueba, se configuró la potencia a transmitir a 14 dBm mediante el pulsador correspondiente y, una vez el GPS se encontraba dando posiciones válidas, se procedió a realizar una transmisión. El hecho de haber configurado en el *backend* el significado de cada byte dentro del mensaje transmitido, permitirá verificar rápidamente si lo que se recibe es coherente con lo que se ha transmitido.

En la siguiente figura se muestra como el mensaje ha sido correctamente recibido y éste ha sido reenviado de forma satisfactoria hacia el SCAN-server mediante el *callback*.



Figura 20. Mensaje a 14 dBm recibido correctamente en el backend.

Como se puede ver, el mensaje recibido es presentado en el *backend* en formato hexadecimal siendo posteriormente decodificado y presentado de forma amigable gracias a la configuración anteriormente mencionada.

Prueba de transmisión a 7 dBm de potencia

La prueba es idéntica a la anterior pero configurando el prototipo para que transmita a una potencia de 7 dBm.



Figura 21. Mensaje a 7 dBm recibido correctamente en el backend.

Prueba de transmisión a 0 dBm de potencia

En este caso se probó la transmisión con la menor de las potencias programadas en el SCAN-device.



Figura 22. Mensaje a 7 dBm recibido correctamente en el backend.

Prueba de transmisión con ACK: ACK recibido correctamente

Esta prueba permite verificar que el prototipo es capaz de transmitir un mensaje con indicativo de ACK y esperar su respuesta. Para ello hay que mantener pulsado el botón "Remote" mientras se presiona el pulsador "Transmit". El proceso de transmitir el mensaje y recibir la respuesta de ACK por parte de la red, tarda aproximadamente 45 segundos. Transcurrido ese tiempo, se pudo comprobar que el LED "TX On" parpadeó con una cadencia de 2 Hz durante 10 segundos, indicando que el ACK se recibió en el prototipo correctamente.



Figura 23. Transmisión con ACK. El backend responde enviando un ACK al prototipo.

Prueba de transmisión con ACK: ACK no recibido

Para esta prueba se configuró el *backend* para forzar un error en el envío del ACK hacia el prototipo. El hecho de que un ACK no se reciba después de haber realizado una transmisión solicitándolo, se refleja en el SCAN-device mediante el LED "TX On". En este caso, cuando transcurre el tiempo de *timeout* sin recibir el ACK, dicho LED, en lugar de parpadear a 2 Hz como en el caso anterior, simplemente se apaga, indicando que no se recibió el ACK.



Figura 24. Transmisión con ACK. Se fuerza un error en el backend para que no envíe el ACK hacia el prototipo.

Pruebas de consumo

Con el objeto de tener una referencia del consumo del prototipo, se procedió a conectar un multímetro, configurado para medida de corriente, en serie a la línea de 5 V de alimentación. De esta forma se midió la corriente consumida para los cuatro escenarios presentados en la Tabla 12.



Figura 25. Test-setup para la medida del consumo

Tabla 12. Consumo del prototipo medido para diferentes escenarios

ESCENARIO	CONSUMO MÁXIMO (mA)
Prototipo encendido, GPS enganchado y sin realizar transmisión	120
Transmisión a una potencia de 14 dBm	160
Transmisión a una potencia de 7 dBm	150
Transmisión a una potencia de 0 dBm	145

Como se puede observar, la mayor contribución al consumo total proviene de la propia placa controladora Arduino, al que hay que añadir el consumo del GPS, siempre encendido desde que se alimenta el SCAN-device. La contribución máxima del módulo de comunicaciones SIGFOX es de aproximadamente 40 mA, en línea a lo especificado en la documentación del fabricante.

4. Desarrollo de la aplicación

Este capítulo abarca el segundo de los desarrollos importantes llevados a cabo durante la realización del trabajo. Una vez implementado el prototipo hardware (SCAN-device), es necesario disponer de una herramienta software (SCAN-tool) que permita recuperar los mensajes transmitidos, tanto del dispositivo como de la red SIGFOX para poder analizarlos. Tal como se definió anteriormente, dicha herramienta debe permitir cumplir los siguientes requisitos principales:

- Presentar una interfaz gráfica que permita una experiencia de usuario satisfactoria.
- Permitir importar los mensajes almacenados en el SCAN-device.
- Permitir importar los mensajes recibidos por la red SIGFOX.
- Procesar los datos y presentar los resultados en un mapa de conectividad con otra información relevante sobre la calidad del servicio mediante código de colores.

En los siguientes apartados se abordará este desarrollo, presentando las partes más relevantes. Al igual que en el capítulo anterior, se ha evitado incluir código fuente directamente en las diferentes secciones, para mantener la claridad del documento, invitando a aquellos lectores que deseen profundizar en los detalles del mismo, a la consulta del anexo correspondiente.

4.1 Entorno de desarrollo

A la hora de decantarse por un lenguaje de programación determinado para acometer el desarrollo, se han tenido en cuenta tres factores principales.

El primero de ellos tiene que ver con las necesidades técnicas. Si bien en un primer momento se podría pensar que cualquier lenguaje de programación de alto nivel moderno podría ser empleado aquí, hay que tener en cuenta dos particularidades. Por un lado el lenguaje elegido debe permitir su uso a tan alto nivel que facilite la implementación de entornos gráficos de usuario de forma sencilla y por otro, debe ser capaz de llegar hasta tan bajo nivel que nos permita construir una variable concreta a partir de los diferentes bytes que la componen. Y es que, realmente, como veremos más adelante, el software debe manipular variables a nivel de bytes para poder extraer la información contenida en los mensajes transmitidos.

El segundo factor a tener en cuenta está relacionado con los plazos a cumplir. No cabe duda que la implementación se debe acometer en unos plazos bastante ajustados, con lo que la curva de aprendizaje del entorno de programación elegido debe ser la más corta posible.

Finalmente, la licencia del entorno de programación también juega un papel importante a la hora de la distribución del software desarrollado. En este sentido, todo el desarrollo contenido en el presente trabajo quedará finalmente bajo licencia GNU/GPL [23], cuestión a tener en cuenta a la hora de elegir el entorno.

Por todo lo anteriormente expuesto, se ha considerado como mejor opción, emplear el entorno de desarrollo QT [24]. QT es un potente *framework* de librerías y herramientas basadas en C++ que permite el desarrollo de aplicaciones que pueden ser portadas a diferentes plataformas con un mínimo esfuerzo, siendo gratuito su uso para aquellas aplicaciones finales no comerciales bajo licencia GPL. El hecho de que la base de QT sea C++, permite aprovechar toda su potencia para desarrollar código a muy bajo nivel, además de permitir, a través de su rica variedad de clases, construir potentes entornos de usuario minimizando el esfuerzo. Aunque el funcionamiento del entorno queda fuera del alcance de este documento, a modo de ejemplo, la siguiente figura muestra el IDE principal desde donde se desarrolló la aplicación.

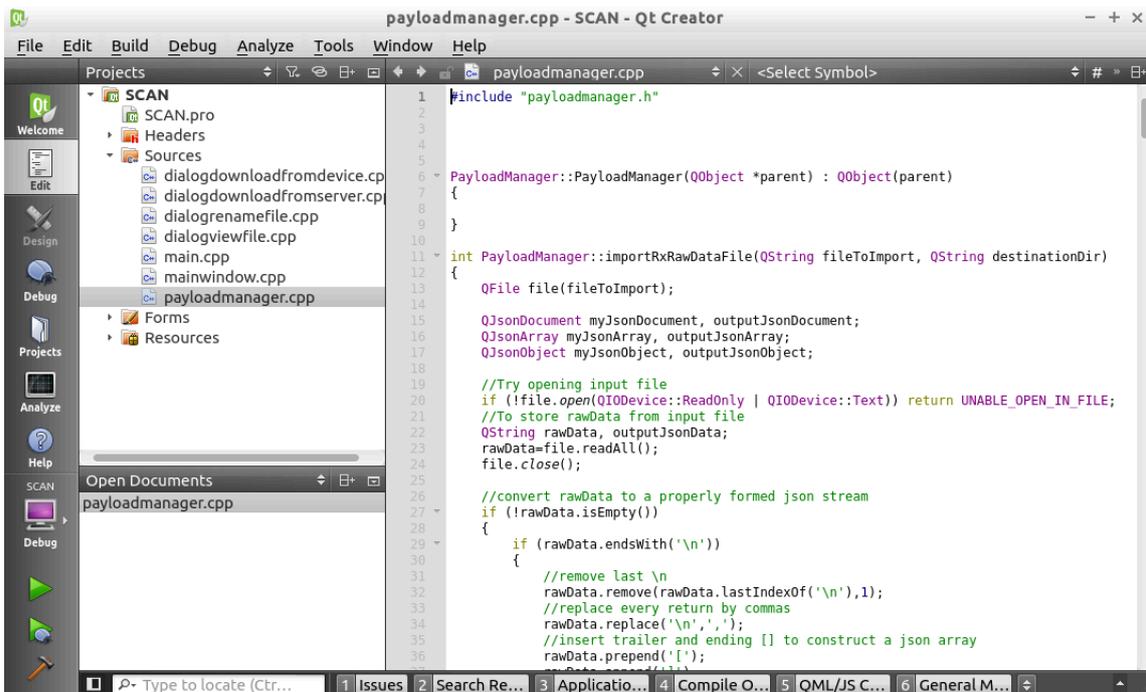


Figura 26. IDE del entorno de desarrollo QT desde donde se desarrolló la aplicación SCAN-tool.

Finalmente, cabe mencionar que, para permitir que el software desarrollado pueda ser ejecutado en un ordenador que no requiera la instalación de un sistema operativo comercial (y por tanto no repercuta en costes adicionales), se ha decidido desarrollar y compilar la versión final sobre una versión GNU/Linux Ubuntu 14.04 [25].

4.2 Diagrama conceptual y formato de ficheros

Antes de entrar en detalle sobre los diferentes módulos que componen el software, conviene detenerse en el diagrama conceptual que nos dará una visión general de las diferentes funcionalidades que implementa. A continuación se muestra dicho diagrama, presentando el detalle de los diferentes formatos de ficheros que se han usado. El uso de estos formatos será justificado en la última parte de esta sección.

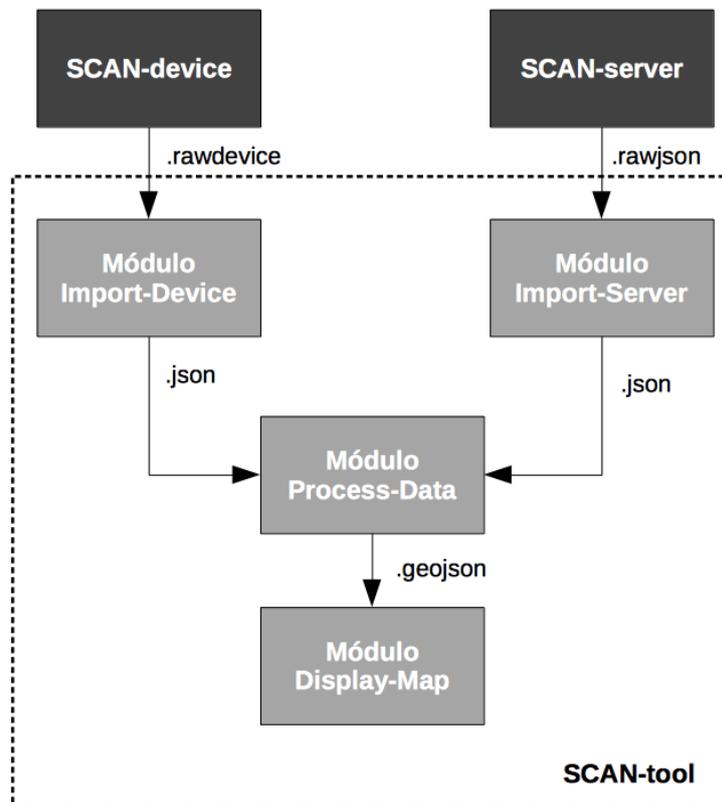


Figura 27. Diagrama conceptual de la aplicación SCAN-tool. Se presentan los diferentes módulos que la componen así como el formato de ficheros de entrada/salida de cada módulo.

Como se puede comprobar en el diagrama, el software estará compuesto por los siguientes cuatro módulos principales.

Módulo “Import Device”

Encargado de importar los datos almacenados en el SCAN-device, tiene la funcionalidad principal de convertir el formato original (*raw-data*) a un formato estándar JSON. Este módulo permite además la gestión y manipulación de los ficheros importados.

Módulo “Import Server”

Similar al anterior, es el encargado de importar los datos almacenados en el SCAN-server, y tiene la funcionalidad principal de convertir el formato original (*raw-data*) a un formato estándar JSON. Este módulo permite también la gestión y manipulación de los ficheros importados.

Módulo “Process Data”

Este es el módulo encargado de procesar los datos y obtener un fichero de salida en formato estándar GeoJSON. Las entradas del módulo son los ficheros importados tanto del SCAN-device como del SCAN-server, y permite realizar un análisis de los datos para obtener puntos de información de servicio SIGFOX geolocalizados incluyendo, para cada uno de ellos, determinada información estadística (número de estaciones base que recibieron el mensaje, señal máxima, señal mínima, etc.).

Módulo “Display Map”

De cara al usuario, es la parte más vistosa de la aplicación, y consiste en la representación en un mapa de toda la información contenida en el fichero GeoJSON obtenido en el módulo de procesado. Permite navegar por el mapa y, seleccionando cualquier punto de servicio, desplegar una ventana *pop-up* con el resto de información estadística asociada a esa transmisión.

Para facilitar la tarea de importa, procesar y presentar los datos, estos cuatro módulos conceptuales corresponden a cuatro pestañas diferenciadas dentro del entorno gráfico como se puede apreciar en la Figura 28.

Por último, en cuanto al formato de ficheros empleados, como se puede comprobar, se ha utilizado el estándar JSON para aquellos ficheros importados en la fase previa al procesado, y el formato GeoJSON para almacenar los datos finales procesados. Por un lado el formato JSON, ampliamente usado en intercambio de información en servicios web, permite realizar un almacenamiento estructurado de la información. Por otro lado, el formato GeoJSON es un formato abierto basado en JSON, que permite representar elementos geométricos además de sus atributos sobre un mapa. La elección de GeoJSON como formato de salida ha sido motivada por el hecho de que se trata de un estándar ampliamente soportado por la mayoría de programas de información geográfica (GIS). Esto implica que los ficheros generados por el SCAN-tool podrán ser visualizados y manipulados en otros programas GIS.

Cabe mencionar que, todos los ficheros importados y generados quedan guardados en la siguiente estructura de carpetas bajo el directorio de instalación de la aplicación.

- raw: directorio destinado a albergar los ficheros que se han descargado del SCAN-device (ficheros con extensión *.rawdevice) y del SCAN-server (ficheros con extensión *.rawjson) pero que todavía no han sido importados.
- tx: directorio destino de los ficheros importados del SCAN-device (ficheros con extensión *.json)
- rx: directorio destino de los ficheros importados del SCAN-server (ficheros con extensión *.json)
- out: directorio destino de los fichero que han sido procesados (ficheros con extensión *.geojson)

En los siguientes apartados se entrará en detalle sobre cada uno de los módulos que componen el software.

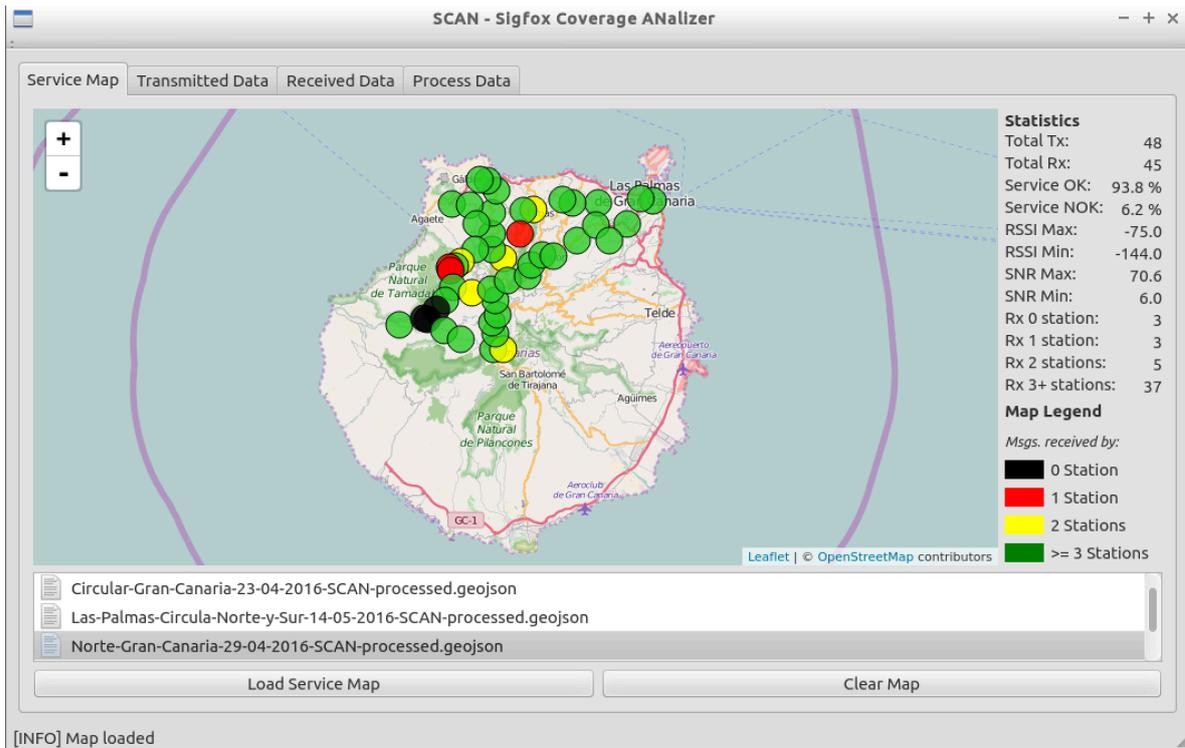


Figura 28. Ventana principal de la herramienta SCAN-tool

4.3 Importar datos del SCAN-device

En la segunda pestaña de la interfaz gráfica se tiene acceso al módulo “Import device”.

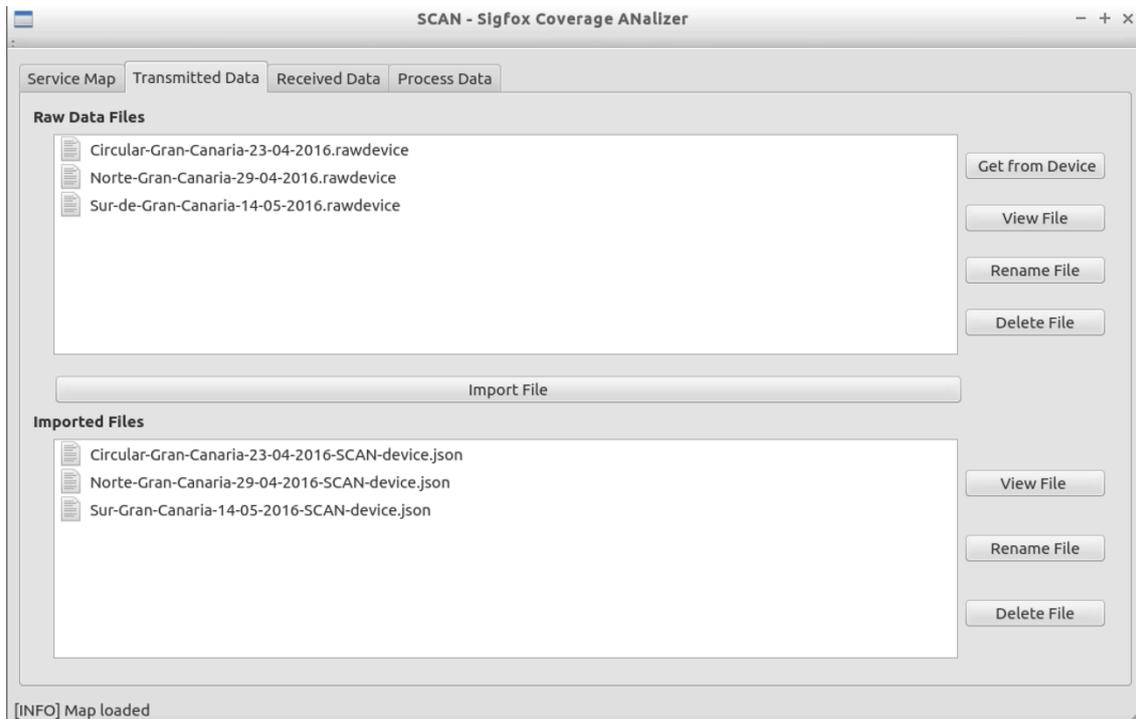


Figura 29. Pestaña del módulo Import-device. Permite importar datos del SCAN-device, así como realizar otras tareas relacionadas con la gestión de ficheros.

Dentro de las diferentes acciones que se implementan, hay dos que son las principales. Por un lado, a través del botón “Get from Device”, se tiene acceso a la ventana que permitirá conectarse, a través el puerto serie, al SCAN-device y descargar los datos de mensajes transmitidos (Figura 30). Cabe mencionar que también se pueden descargar los datos usando cualquier programa externo que permita enviar comandos por el puerto serie y recibir los mensajes. Ese fichero descargado bastará con copiarlo dentro del directorio “raw” para que pueda ser importado por el SCAN-device.



Figura 30. Diálogo que permite conectarse al SCAN-device a través del puerto serie para la descarga de datos.

La segunda acción importante que se puede realizar es la importación de los datos para generar un fichero estándar JSON. Las principales acciones que realiza el programa cuando se pulsa el botón “Import File” son:

- Para cada una de las líneas del fichero de entrada, extrae el valor de sus campos.
- Para algunos de los campos extraídos, se realizan ciertas transformaciones y los resultados se almacenan, junto con el resto de campos, en el fichero de salida como un array de objetos JSON. Estas transformaciones están relacionadas con la conversión de la latitud y la longitud, del formato tipo Long, a un formato entendible tipo Float.

En la siguiente figura se muestra un ejemplo del contenido de un fichero en el formato final una vez ha sido importado.

```
[
  {
    "latitude": "28.119852066040",
    "longitude": "-15.424726486206",
    "power": "14",
    "rxACK": false,
    "txACK": false,
    "txID": "107"
  },
  {
    "latitude": "28.123359680176",
    "longitude": "-15.440066337585",
    "power": "14",
    "rxACK": false,
    "txACK": false,
    "txID": "108"
  },
  {
    "latitude": "28.095064163208",
    "longitude": "-15.456278800964",
    "power": "14",
    "rxACK": false,
    "txACK": false,
    "txID": "109"
  }
]
```

Figura 31. Formato JSON de salida de un fichero que ha sido importado del SCAN-device.

4.4 Importar datos del SCAN-server

Antes de describir este módulo, cabe detenerse en las funciones del SCAN-server. Tal como se representó en la arquitectura del sistema, el *backend* de SIGFOX ha sido configurado para retransmitir todos los mensajes recibidos por todas las estaciones de cada una de las transmisiones realizadas por el SCAN-device hacia un servidor externo de usuario (SCAN-server). Este modo de funcionamiento se conoce como *callback*, y es el mecanismo recomendado por SIGFOX para la recuperación de los mensajes que se transmiten desde cualquier sensor hacia la red.

El SCAN-server está formado por una combinación de servidor *http* con capacidad de ejecutar código *php*, en el que se ha instalado una página que ejecuta un pequeño código php para almacenar los mensajes recibidos desde los servidores de SIGFOX. Este servidor puede estar instalado en modo local en la propia máquina del usuario o en servidores externos y el único requisito indispensable es que su dirección IP sea accesible desde el *backend* de SIGFOX para que el mecanismo de *callback* pueda redirigir los mensajes hacia él. En el repositorio correspondiente (ver Anexo A.1) se puede consultar el código de la página encargada de ir almacenando los datos recibidos como objetos JSON dentro del fichero. Este fichero tendrá el formato [IDSensor].rawjson, donde [IDSensor] será el número identificativo del sensor que ha transmitido el mensaje.

Pasando a la descripción del módulo, éste está representado por la tercera pestaña del programa y su funcionamiento es prácticamente igual al anterior, con la diferencia de que en este caso, la opción “Get from device” queda sustituida por la opción “Get from server” (Figura 32). En este caso, se abrirá una ventana que permitirá la descarga de los mensajes recibidos por la red SIGFOX y que han sido retransmitidos hacia el SCAN-server.

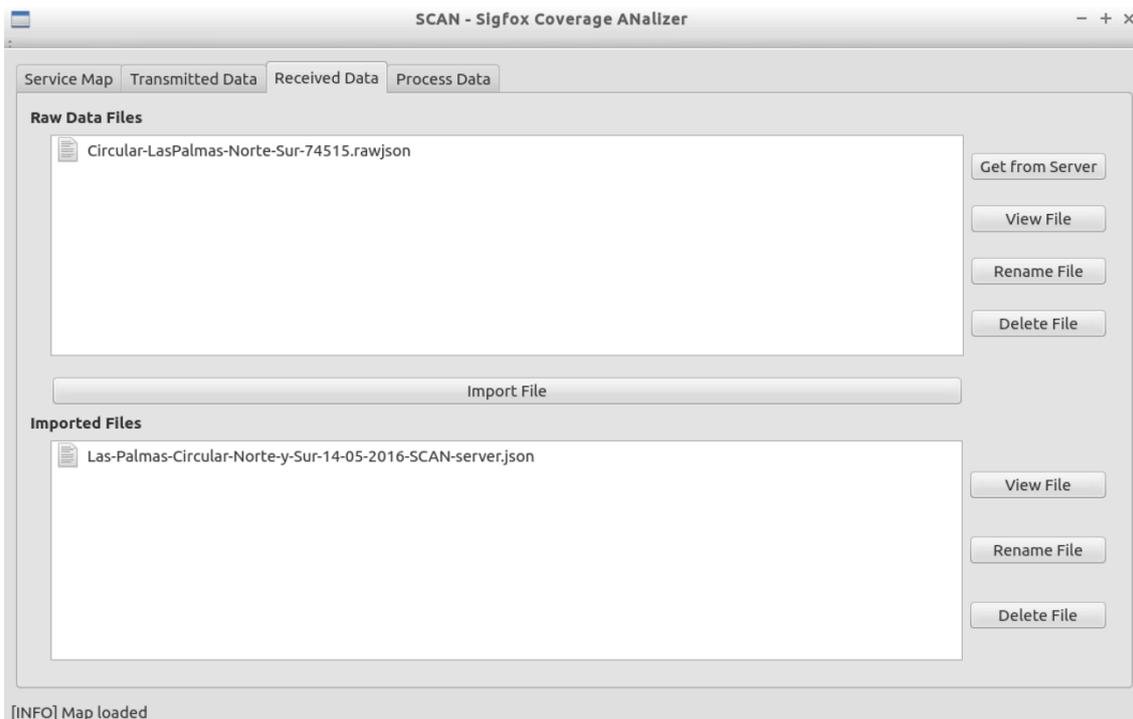


Figura 32. Pestaña del módulo Import-server. Permite importar datos del SCAN-server, así como realizar otras tareas relacionadas con la gestión de ficheros.

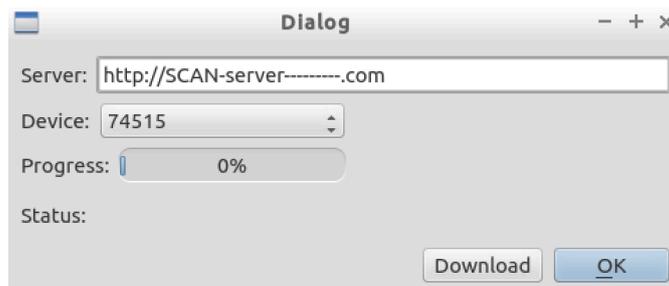


Figura 33. Diálogo que permite descargar los datos del SCAN-server

Al igual que el anterior módulo, el botón “Import File” permite generar un fichero JSON estándar a partir del fichero descargado del servidor. Las principales acciones realizadas por el programa cuando se pulsa el botón “Import File” son:

- A partir de cada una de las líneas del fichero de entrada sin formatear, crea un array de objetos JSON realizando determinadas transformaciones a los datos.
- La primera de estas transformaciones consiste en convertir la hora a la que se recibió el mensaje proporcionada por los servidores SIGFOX de formato UNIX a un formato estándar.
- La siguiente transformación importante consiste en, a partir de los 11 bytes que fueron originalmente transmitidos por el SCAN-device, ir extrayendo byte a byte y construyendo las diferentes variables que contienen (TXID, potencia, latitud y longitud). Es importante hacer notar aquí que en dicha transformación hay que

tener en cuenta si la máquina sobre la que se realiza esta decodificación tiene arquitectura *Big-endian* o *Little-endian*, pues influirá en el orden en que se colocan los bytes para construir la variable en cuestión. Este hecho se tiene previsto en el código llamando a una función determinada que nos dice el tipo de arquitectura de la máquina antes de realizar la decodificación.

La siguiente figura muestra un ejemplo de fichero JSON que ha sido importado del SCAN-server.

```
[
  {
    "device": "74515",
    "duplicate": "false",
    "latitude": "28.113721847534",
    "longitude": "-15.425077438354",
    "power": "14",
    "rssi": "-123.00",
    "snr": "16.85",
    "station": "085A",
    "time": "10-04-2016 14:01:19",
    "txID": "30"
  },
  {
    "device": "74515",
    "duplicate": "true",
    "latitude": "28.113721847534",
    "longitude": "-15.425077438354",
    "power": "14",
    "rssi": "-132.00",
    "snr": "13.25",
    "station": "0B63",
    "time": "10-04-2016 14:01:19",
    "txID": "30"
  }
]
```

Figura 34. Ejemplo de formato JSON de salida de un fichero que ha sido importado del SCAN-server

4.5 Procesado de datos

La cuarta pestaña disponible en el programa da acceso a la capacidad de poder procesar los ficheros importados para obtener un fichero GeoJSON con los resultados. Como se puede ver, aparte de las típicas opciones de gestión de ficheros como renombrar o borrar, permite seleccionar uno o varios ficheros importados de mensajes transmitidos y un fichero de mensajes recibidos por la red para ser procesados.

El hecho de realizar la selección de esta manera (varios ficheros de mensajes transmitidos, pero sólo un fichero de mensajes recibidos) tiene su justificación en el hecho de que una campaña de medidas puede durar uno o varios días. Durante el periodo completo que dure la campaña, el fichero alojado en el SCAN-server será único, e irá recopilando todos los mensajes recibidos por parte de los servidores de SIGFOX. Por otro lado, si la campaña dura varios días, tiene sentido que sean varios los ficheros importados procedentes del SCAN-device, pues cada vez que se alcance

la capacidad de 90 transmisiones, se deberá proceder a la descarga de datos, y por tanto a la generación de un fichero nuevo.

Para proceder por tanto al procesado de los datos, una vez seleccionados los ficheros, se pulsará el botón "Process Files". En ese instante, las acciones principales que realizará el programa serán las siguientes.

- Comprobar que la combinación de ficheros seleccionados es correcta.
- Si es así, para cada una de las transmisiones realizadas (TXID), busca si se ha recibido en la red o no y va guardando los resultados en el fichero de salida GeoJSON.
- Si se ha producido recepción del mensaje por parte de la red SIGFOX, también cuenta el número de estaciones que lo recibieron y va calculando el RSSI máximo, RSSI mínimo, SNR máximo y SNR mínimo.

Cuando finaliza el procesado, en el fichero GeoJSON se tendrá un array de objetos tipo Feature representando, cada uno, una transmisión geolocalizada y conteniendo toda la información anteriormente calculada.

```
{
  "features": [
    {
      "geometry": {
        "coordinates": [
          -15.424726486206,
          28.11985206604
        ],
        "type": "Point"
      },
      "properties": {
        "popupContent": "TXID: 107<br>Service: OK<br>Transmitted Power: 14 dBm<br>Transmission with ACK: No",
        "power": "14",
        "rssiMax": -92,
        "rssiMin": -142,
        "service": "OK",
        "snrMax": 16.73,
        "snrMin": 8.2200000000000006,
        "stations": 16,
        "txID": "107"
      },
      "type": "Feature"
    },
  ],
  "type": "FeatureCollection"
}
```

Figura 35. Contenido parcial de un fichero GeoJSON obtenido al procesar varios ficheros JSON de entrada.

La siguiente figura muestra la pestaña dedicada a procesar los datos.

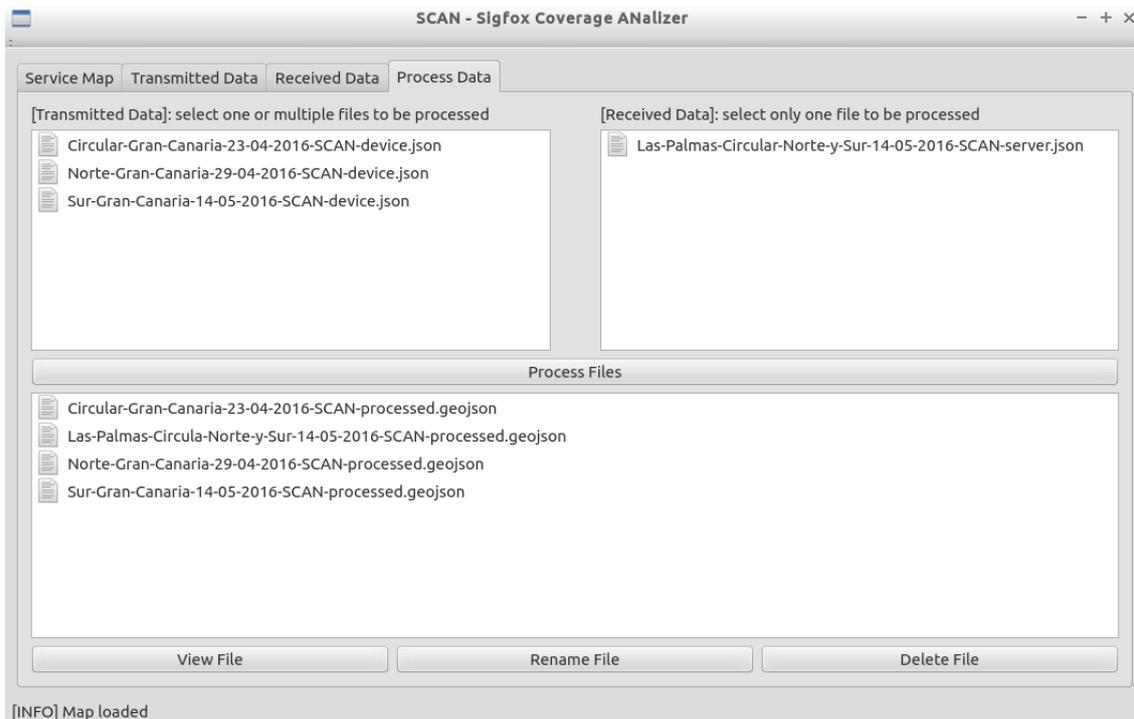


Figura 36. Pestaña del módulo Process-data.

4.6 Presentación de resultados en el mapa

Esta sección del programa corresponde a la primera pestaña y es la que aparece activada por defecto cuando se inicia la aplicación. Está dividida en tres secciones principales como se puede apreciar en la siguiente Figura 37.

La primera de ellas muestra la lista de ficheros procesados GeoJSON que hay disponibles en la aplicación listos para ser visualizados. Cada uno de estos fichero corresponde al resultado de un procesado concreto. Desde aquí podemos seleccionar el que queramos mostrar en el mapa.

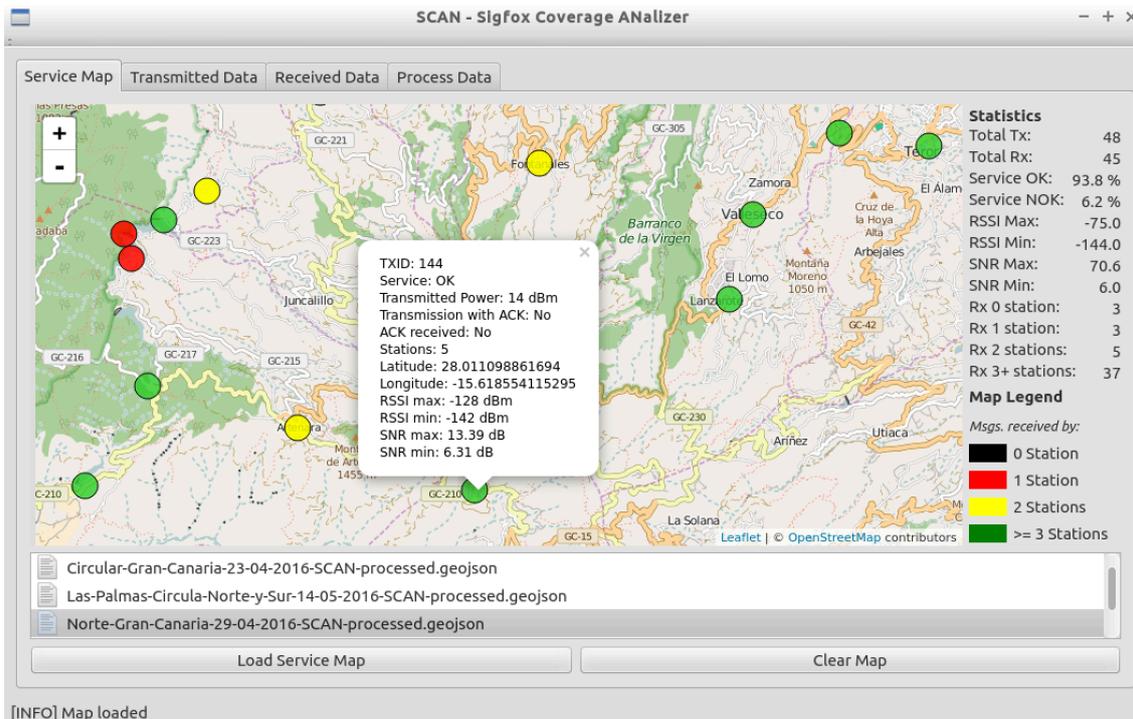


Figura 37. Ventana principal de presentación del Mapa y resultados.

La segunda sección importante corresponde al mapa donde se presentarán los resultados. Cabe detenerse aquí para aclarar la estrategia seguida a la hora de programar este módulo. Después de estudiar varias posibilidades de programación que permitiese representar circunferencias de diferentes colores sobre un mapa, finalmente se decidió emplear aquella que ofreciera la posibilidad de reutilizar el código para poder cumplir de forma rápida con uno de los últimos objetivos definidos en el trabajo. Este objetivo consistía en presentar los resultados de la campaña de medidas en una página web de acceso público donde se pudiesen mostrar las medidas realizadas.

Es por ello que, aprovechando que el *framework* de QT disponía de una clase que permitía visualizar, directamente en una ventana, el resultado de procesar código html (clase QWebView), se optó por dicha solución. Es decir, la presentación del mapa con los resultados de las medidas, se ha implementado generando código html desde la propia aplicación, a partir del fichero GeoJSON, y pasando dicho código a la clase QWebView. De esta forma, el código generado es totalmente reutilizable a la hora de presentar los datos en una web.

Cabe destacar que, para la presentación de los puntos de colores, así como del mensaje con la información asociada que aparece al pulsar sobre ellos, se ha hecho uso de las librerías de código abierto y libre uso, LeafLet [26] y jQuery [27], además de los mapas de acceso libre de OpenStreetMap [28].

Finalmente, la tercera sección es un espacio dedicado a las estadísticas calculadas sobre el mapa. Cada vez que el usuario carga un mapa nuevo, el programa calcula una serie de estadísticas sobre el fichero GeoJSON que se está representando. Los valores que se calculan son:

- Total Tx: número total de mensajes transmitidos.
- Total Rx: número total de mensajes recibidos.
- Service OK: porcentaje de mensajes que han sido recibidos sobre el total de transmitidos.
- Service NOK: porcentaje de mensajes que se han perdido sobre el total de mensajes transmitidos.
- RSSI Max: nivel de señal máximo recibido.
- RSSI Min: nivel de señal mínimo recibido.
- SNR Max: relación señal-a-ruido máxima recibida.
- SNR Min: relación señal-a-ruido mínima recibida
- Rx 0 station: número total de mensajes que no han sido recibidos por ninguna estación base.
- Rx 1 station: número total de mensajes que han sido recibidos por una única estación base.
- Rx 2 stations: número total de mensajes que han sido recibidos por 2 estaciones base.
- Rx 3+ stations: número total de mensajes que han sido recibidos por 3 o más estaciones base.

5. Medidas de Campo

Este capítulo abordará el tercer y último objetivo planteado en el alcance de este trabajo. Consistirá en el diseño y realización de una campaña de medidas de campo en la isla de Gran Canaria, y permitirá cumplir dos propósitos simultáneamente.

Por un lado, servirá para poner en práctica el funcionamiento, tanto del prototipo desarrollado como de la herramienta de análisis en su conjunto, permitiendo validar que se ha cumplido satisfactoriamente el alcance del proyecto.

Por otro lado, va a permitir generar un mapa preliminar de medidas reales del servicio de la red SIGFOX en la isla, y que podrá ser usado por cualquiera que esté interesado en realizar un despliegue de sensores en el área estudiada y que pretenda usar esta tecnología de acceso. En este sentido, los resultados obtenidos serán finalmente alojados en un sitio web de acceso público para su consulta.

5.1 Criterios y planificación de las localizaciones

Gran Canaria es una de las islas que forman el Archipiélago Canario, situada en el océano Atlántico a $28^{\circ} 58'$ de latitud Norte y $15^{\circ} 35'$ de longitud Oeste y a aproximadamente 220 Km del continente Africano. Con forma casi circular, tiene una extensión de unos 1550 km^2 , con una altitud máxima de 1949 m siendo la isla que presenta la topografía más accidentada y compleja del Archipiélago [29]. Este hecho hace que sea especialmente interesante analizar cuál es el nivel de cobertura del servicio alcanzado por SIGFOX ante un escenario de complejas estructuras topográficas.





Figura 38. Muestra de algunas de las estructuras topográficas recorridas durante la campaña de medidas de campo en la isla de Gran Canaria.

La isla, en general, presenta buenas conexiones por carretera, existiendo un anillo de autovía y autopista que cubre el perímetro Norte, Este y Sur, quedando el perímetro Oeste recorrido por una carretera sinuosa bastante complicada que discurre en su mayor parte por acantilados. El resto de la isla queda abarcada por una amplia red de carreteras secundarias distribuida prácticamente en forma de estrella, de tal manera que forman accesos desde la costa hasta la cumbre a distancias casi equidistantes.

Habiendo estudiado la mejor forma de afrontar la mayor superficie de la isla en el tiempo disponible, se ha planificado la campaña en tres jornadas de 10 horas cada una. Para cada una de estas jornadas y, debido a la cantidad de kilómetros que se recorren en cada una -más de 200 Km por jornada-, se ha planificado una hora para descanso acumulado y otra hora para avituallamiento, con lo que el tiempo efectivo de medidas, por jornada, quedaba fijado en 8 horas.

Cada una de estas tres jornadas seguirá una ruta predefinida, quedando distribuidas de la siguiente forma:

Día 1. Ruta perimetral (227 Km)

Puesto que, exceptuando la zona suroeste, el resto del perímetro de costa de Gran Canaria se encuentra recorrido por una anillo de carreteras, esta jornada se centrará en cubrir esa ruta circular.

Día 2. Ruta Norte (214 Km)

Esta ruta comienza y termina en Las Palmas de Gran Canaria, realizando un recorrido por el Norte y Oeste de la isla, empleando para ello las diferentes carreteras secundarias que conectan las medianías con la cumbre.

Día 3. Ruta Sur (323 Km)

Será la última jornada de medidas, y tiene una planificación similar a la anterior (alternando subidas y bajadas desde la costa hasta la cumbre), pero cubriendo la zona Este y Sur de la isla. Como el resto, comenzará y terminará en la capital.

Cabe mencionar que, puesto que en las pruebas funcionales del prototipo, también se realizaron medidas que quedaron validadas dentro de la ciudad de Las Palmas de

Gran Canaria, éstas también se incluyeron en el mapa final. La siguiente figura muestran sobre el mapa las rutas planificadas y ejecutadas.



Figura 39. Detalle de las rutas planificadas durante la campaña de medidas. Día 1 en rojo; Día 2 en blanco y día 3 en amarillo. Total 764 Km recorridos.

Por otro lado, el *test-setup* para la realización de las pruebas consistió en instalar el SCAN-device sobre una motocicleta donde, aparte del prototipo, también se emplearon los siguientes elementos auxiliares:

- GPS auxiliar para el seguimiento y grabación del *track* planificado.
- Toma de corriente USB conectada a la batería de la motocicleta. Esta toma de corriente fue empleada para la alimentación del GPS auxiliar.
- Batería de respaldo. Aunque el SCAN-device podría haberse conectado directamente a la toma de corriente proporcionado por el vehículo, por seguridad y para evitar cualquier avería debida a posibles picos de sobretensión, se decidió alimentarlo a través de una batería externa.

En la siguiente fotografía se puede observar como quedó configurado el panel de medidas completo instalado sobre el vehículo.



Figura 40. Configuración del panel de medida instalado en la motocicleta

Cabe mencionar que, la posición final del SCAN-device fue dispuesta de tal manera que la antena permanecía en posición perfectamente vertical cuando la motocicleta se encontraba en posición de reposo (apoyada sobre su soporte de aparcamiento) y siempre a la misma distancia sobre el suelo entre medidas (1,2 metros) .

Finalmente, a la hora de elegir las localizaciones y planificar las transmisiones, se siguieron las siguientes pautas. Por un lado, se han elegido las principales presas de Gran Canaria, al ser elementos representativos y susceptibles de ser objeto de monitorización -y por tanto de despliegue de sensores en su entorno-. Estas construcciones, se encuentran emplazadas en lugares remotos en los cauces de los principales barrancos de la isla, y es por ello por lo que también representan puntos de interés de cara al reto de poder conseguir conectividad de la red SIGFOX en su entorno.

Por otro lado, tal como se ha referido en diferentes partes del documento, la máxima capacidad de transmisión que nos permite esta tecnología, regulada por normas ETSI, es de un mensaje transmitido cada 10 minutos aproximadamente (esto permite un máximo de 140 mensajes por día). Es por ello que, independientemente de que se realicen las medidas en los puntos de interés antes mencionados, para poder aprovechar al máximo dicha capacidad de transmisión durante las horas que dure cada jornada de medidas, una vez en ruta, se ha definido una parada cada 10 minutos para realizar una transmisión.

Tal como se explicaba en el capítulo dedicado al desarrollo del SCAN-device, éste permite dos modos de transmisión. El primero es el modo nominal, en el cual se transmite el mensaje, y el mismo es recibido por las estaciones base, si está bajo cobertura de alguna de ellas. Este modo requiere aproximadamente unos 12 segundos

para completar la transmisión (6 segundos de encendido del módulo de comunicaciones, más otros 6 segundos empleados en realizar la transmisión).

El segundo modo es el modo de transmisión con reconocimiento (acknowledgment-ACK). Este modo requiere aproximadamente 45 segundos para completarse, pues el módulo de comunicaciones quedará a la espera de que se reciba un ACK por parte de la red indicando que se ha recibido correctamente el mensaje, y permite una capacidad de como máximo 4 mensajes de *downlink* al día. Aunque este modo de transmisión puede ser útil para verificar de forma puntual la conectividad no solo a nivel de *uplink* (único modo disponible en la versión original del protocolo SIGFOX y optimizado para ahorro de batería), sino también en el *downlink* (incluido posteriormente en la revisión del protocolo), en las medidas de campo aquí realizadas no se empleará la transmisión con ACK.

Por último, aunque el prototipo permite transmitir hasta con tres potencias diferentes, se tardarían unos 30 minutos en completar cada medida (habría que esperar 10 minutos entre transmisiones), lo cual es inviable si queremos cubrir el área planificada en el tiempo disponible. Es por ello que se ha decidido emplear la máxima potencia (14 dBm) para todas las transmisiones, lo que permite simular un dispositivo de tipo 0U, y por tanto comparar los resultados obtenidos directamente con el mapa de cobertura de servicio simulado por SIGFOX (disponible públicamente para unas condiciones de transmisión de dispositivos certificados de tipo 0U).

5.2 Análisis y publicación de resultados

En este apartado se presentarán los resultados obtenidos durante la campaña de medidas, desglosándolos, en un primer momento por jornadas de medidas, y presentando finalmente el mapa completo de servicio incluyendo todas las transmisiones. Cabe resaltar que todos los mapas y resultados aquí presentados han sido generados con el herramienta desarrollada SCAN-tool.

Resultados del Día 1

Las siguientes figuras muestran una visión global de la isla por donde discurrió la jornada con cada una de las transmisiones realizadas, así como algunos ejemplos de detalle de zonas donde se han dado los cuatro posibles casos que se pueden presentar en el mapa y que dan una visión sobre la calidad del servicio. Éstos son:

- Puntos verdes: transmisiones localizadas en zonas cubiertas por 3 o más estaciones, es decir, las transmisiones realizadas en estos puntos fueron recibidos por 3 o más estaciones base.
- Puntos amarillos: transmisiones localizadas en zonas cubiertas por 2 estaciones. Las transmisiones realizadas en estos puntos fueron recibidos por 2 estaciones base.
- Puntos rojos: transmisiones localizadas en zonas cubiertas por 1 estación, es decir, las transmisiones realizadas en estos puntos fueron recibidos por solo una estación base.
- Puntos negros: transmisiones en los que no se ha recibido el mensaje por ninguna estación base. Se considera que en ese punto no hay servicio.

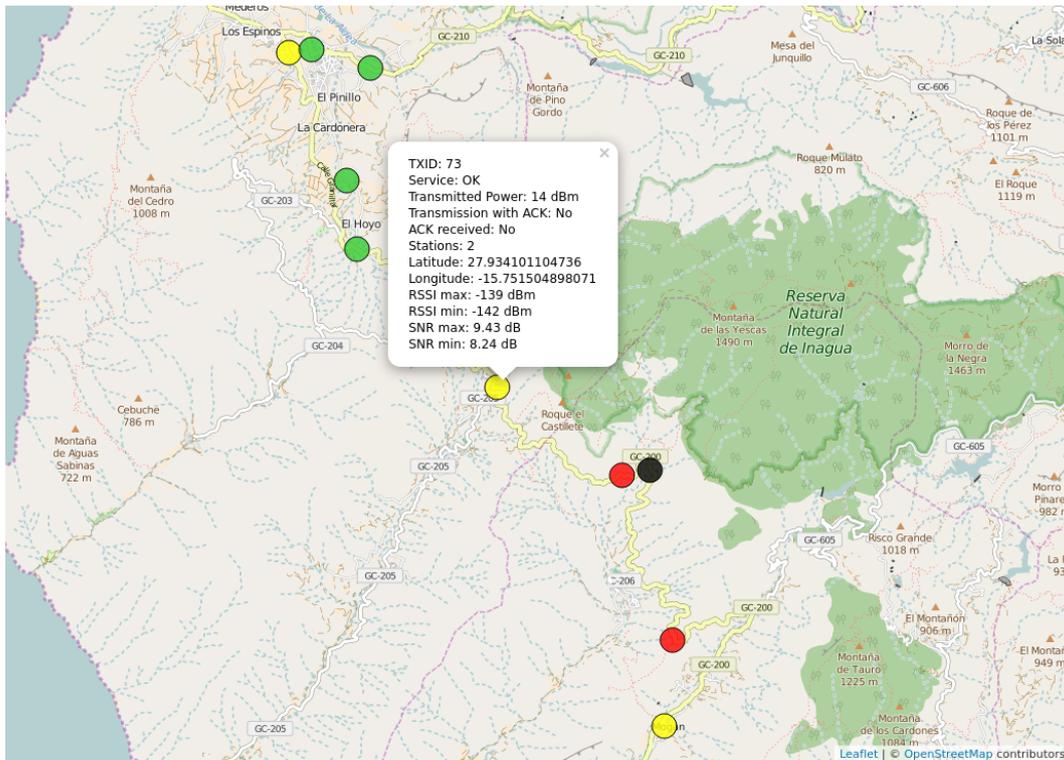


Figura 43. Detalle de una transmisión que ha sido recibida por 2 estaciones base.

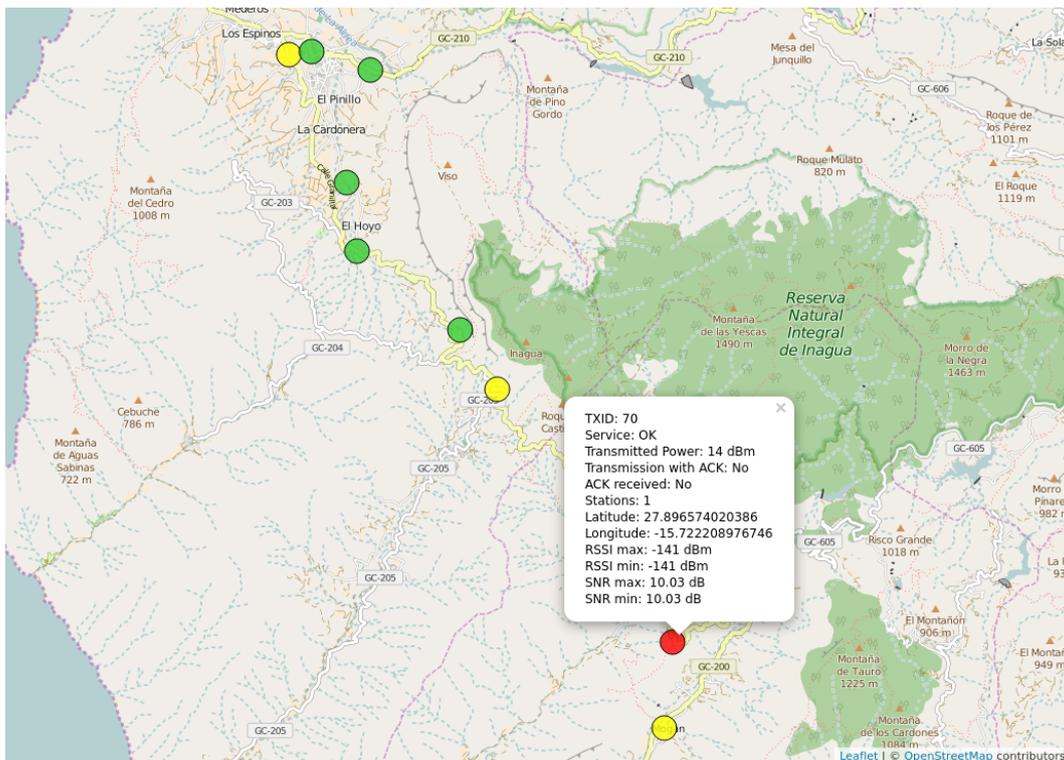


Figura 44. Detalle de una transmisión que ha sido recibida por 1 estación base. En este caso, no existe redundancia en el número de estaciones base que cubren ese punto.

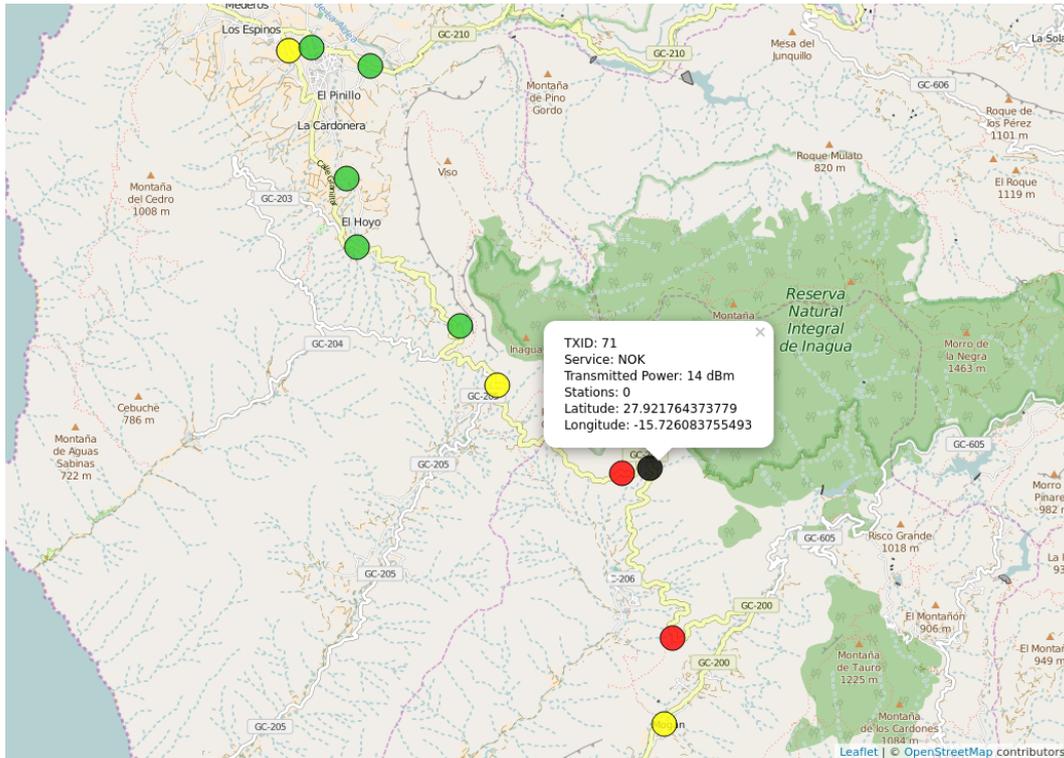


Figura 45. Detalle de una transmisión que no ha sido recibida por ninguna estación base. En ese punto se puede considerar que no hay servicio de la red SIGFOX.

En cuanto a las estadísticas calculadas por la herramienta para esa jornada, éstas quedan resumidas en la Tabla 13.

Tabla 13. Resultado de las estadísticas calculadas por el SCAN-tool para la primera jornada de la campaña de medidas

Número de mensajes transmitidos	75
Número de mensajes recibidos por la red SIGFOX	74
Porcentaje de transmisiones que han sido recibidas por la red	98.7%
Porcentaje de transmisiones que no han sido recibidas por la red	1.3%
Nivel de señal recibido máximo (RSSI max)	-73 dBm
Nivel de señal recibido mínimo (RSSI min)	-147 dBm
Relación Señal-a-Ruido máxima (SNR max)	36.5 dB
Relación Señal-a-Ruido mínima (SNR min)	3.9 dB
Número de transmisiones que no fueron recibidas por ninguna estación base	1
Número de transmisiones que fueron recibidas por sólo 1 estación base	4
Número de transmisiones que fueron recibidas por sólo 2 estaciones base	8
Número de transmisiones que fueron recibidas por 3 o más estaciones base	62

Se puede apreciar que el nivel de servicio es muy bueno para la ruta estudiada, consiguiendo conectividad en el enlace ascendente, al menos a través de una estación base, en un 98.7% de las transmisiones efectuadas.

En cuanto a los valores de RSSI y SNR, éstos son datos medidos y proporcionados por las estaciones base que recibieron las transmisiones y que son incluidos en los mensajes de *callback* hacia el SCAN-server. Cabe destacar, en este sentido, los -147 dBm de sensibilidad máxima de alguna de las estaciones base que recibieron mensajes. Este hecho es coherente con las altas capacidades de margen de enlace que caracterizan a las redes LPWAN y que permiten conseguir conectividad a grandes distancias.

Finalmente, se puede apreciar a simple vista, a través del mapa de servicio de la Figura 41, que la zona donde menor redundancia presenta la red (es decir, donde la zona se encuentra cubierta por menos estaciones base) se da para la zona Sur-Oeste de la isla.

Puesto que ya se han presentado en este apartado algunos ejemplos detallados de los diferentes puntos de servicio que se pueden representar en el mapa, en los resultados correspondientes a las siguientes jornadas, sólo se mostrará el mapa general obtenido con la herramienta, así como las estadísticas calculadas.

Resultados del Día 2

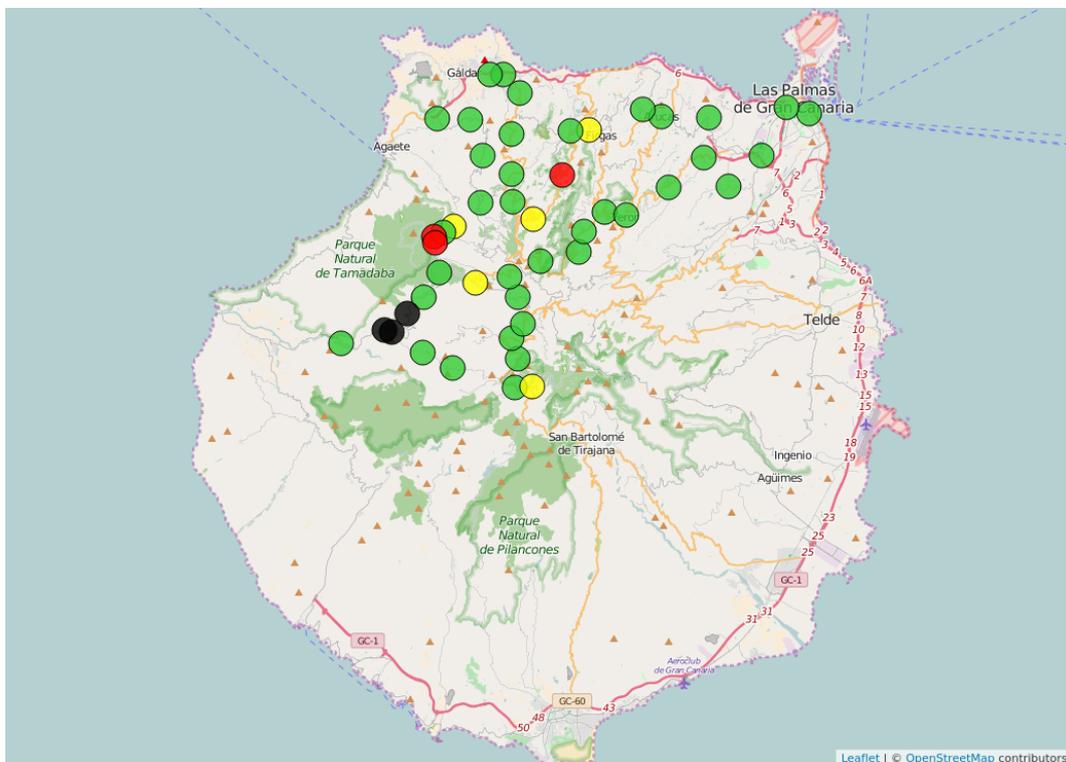


Figura 46. Mapa de nivel de servicio correspondiente a la segunda jornada de medidas.

Tabla 14. Resultado de las estadísticas calculadas por el SCAN-tool para la segunda jornada de la campaña de medidas

Número de mensajes transmitidos	48
Número de mensajes recibidos por la red SIGFOX	45
Porcentaje de transmisiones que han sido recibidas por la red	93.8%
Porcentaje de transmisiones que no han sido recibidas por la red	6.2%
Nivel de señal recibido máximo (RSSI max)	-75 dBm
Nivel de señal recibido mínimo (RSSI min)	-144 dBm
Relación Señal-a-Ruido máxima (SNR max)	70.6 dB
Relación Señal-a-Ruido mínima (SNR min)	6.0 dB
Número de transmisiones que no fueron recibidas por ninguna estación base	3
Número de transmisiones que fueron recibidas por sólo 1 estación base	3
Número de transmisiones que fueron recibidas por sólo 2 estaciones base	5
Número de transmisiones que fueron recibidas por 3 o más estaciones base	37

Resultados del Día 3

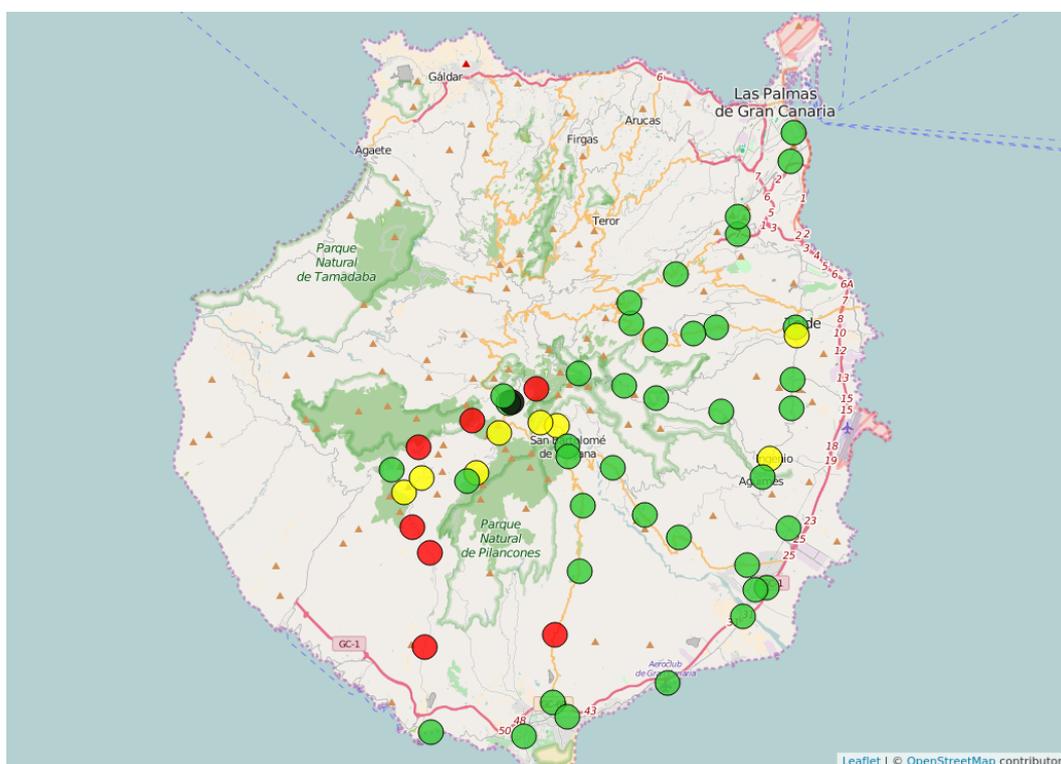


Figura 47. Mapa de nivel de servicio correspondiente a la tercera jornada de medidas.

Tabla 16. Estadísticas globales de la campaña de medidas de campo

Número de mensajes transmitidos	178
Número de mensajes recibidos por la red SIGFOX	173
Porcentaje de transmisiones que han sido recibidas por la red	97.2%
Porcentaje de transmisiones que no han sido recibidas por la red	2.8%
Nivel de señal recibido máximo (RSSI max)	-73.0 dBm
Nivel de señal recibido mínimo (RSSI min)	-147.0 dBm
Relación Señal-a-Ruido máxima (SNR max)	70.6 dB
Relación Señal-a-Ruido mínima (SNR min)	3.9 dB
Número de transmisiones que no fueron recibidas por ninguna estación base	5
Número de transmisiones que fueron recibidas por sólo 1 estación base	14
Número de transmisiones que fueron recibidas por sólo 2 estaciones base	21
Número de transmisiones que fueron recibidas por 3 o más estaciones base	138

Finalmente, y con el objeto de que pueda ser consultado de forma pública por todos aquellos que deseen conocer el estado del servicio de la red SIGFOX sobre el área estudiada, se ha publicado el mapa final de servicio en la página web <http://coverage.planetamaxwell.com>. Desde esa página, se puede interactuar directamente con cada uno de los puntos representados para conocer más detalles sobre los mismos, así como ampliar la zona bajo estudio hasta el máximo nivel de zoom disponible.

5.3 Cobertura real medida vs cobertura simulada

Una vez obtenido el mapa con la información de más de 170 puntos de servicio, cabría preguntarse si lo que se ha medido es coherente con la simulación de cobertura que SIGFOX presenta en su página web. Es por ello que se planteó la posibilidad de superponer el mapa obtenido en la campaña de medidas con el mapa de cobertura simulada calculada por el operador [5].

Para tal propósito y, debido que el mapa de cobertura disponible no reunía las condiciones de resolución mínima necesarias para poder realizar la comparativa, se contactó directamente con el operador, proporcionando éste el mapa representado en la Figura 53. Este mapa, aunque no presenta información de redundancia (áreas diferenciadas por color, con información del número de estaciones que cubren una determinada zona), dispone de la resolución suficiente como para realizar un análisis cualitativo de la comparación entre las medidas reales y las simuladas.

El mapa simulado bicolor de la Figura 49 representa, en color verde, las zonas donde se prevé que haya servicio, mientras que las zonas en color gris representan aquellas donde no hay cobertura, según la simulación.

Cabe destacar que esta simulación corresponde a la estimación de cobertura para localizaciones en exterior y para dispositivos de clase 0U, es decir, aquellos transmitiendo a la máxima potencia. Es por ello que el hecho de haber realizado la

campana de medidas transmitiendo como un dispositivo de tipo 0U, permite hacer directamente la comparación de los resultados obtenidos con el mapa simulado.



Figura 49. Mapa proporcionado por el operador, con la simulación de la cobertura de servicio de la red SIGFOX para la isla de Gran Canaria.

Para la representación del mapa con los puntos de servicio, se procedió a extraer, mediante una aplicación de tratamiento de imagen, una capa conteniendo solamente la información de interés. Posteriormente se incorporó dicha capa sobre el mapa de simulación de cobertura, utilizando puntos de referencia para alinear las imágenes. El resultado de este proceso se puede ver en la Figura 50.

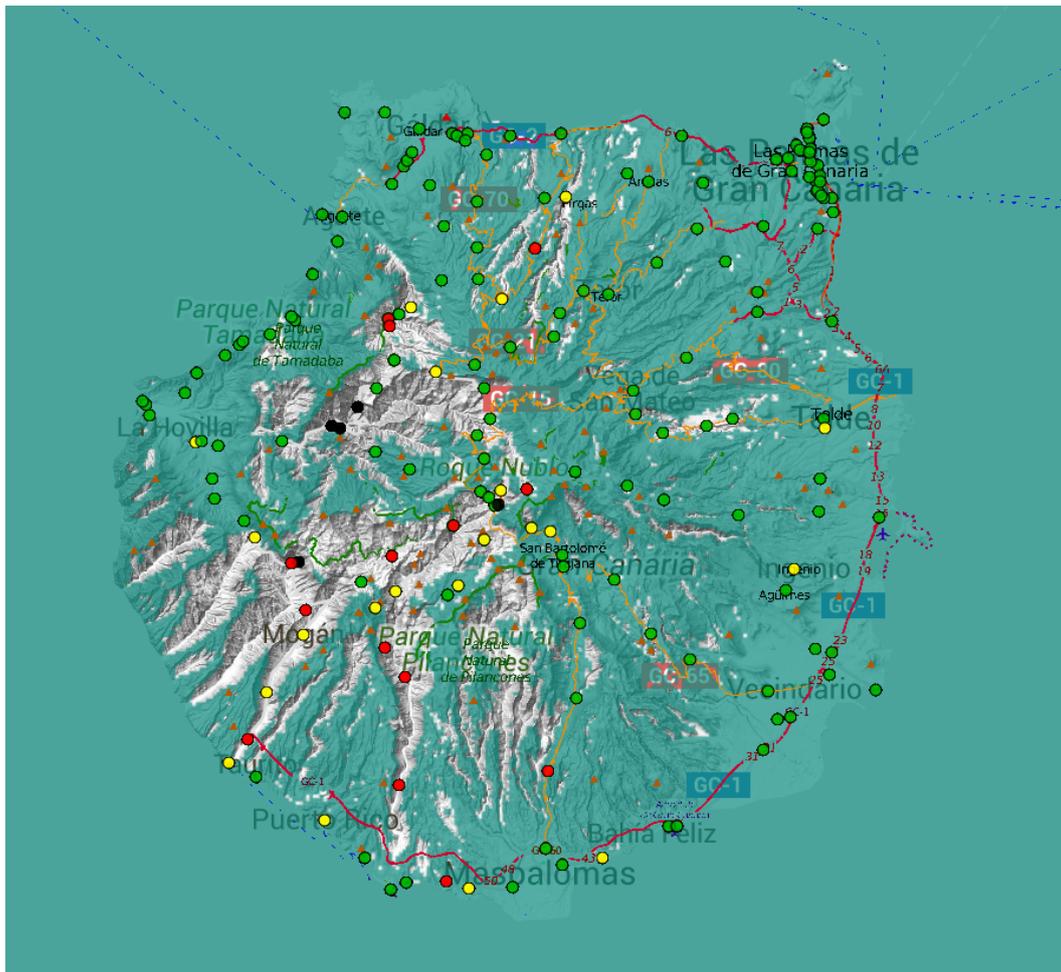


Figura 50. Mapa de simulación de cobertura con información de los puntos de servicio obtenidos en la campaña de medidas.

Dentro de las posibilidades limitadas de análisis que el mapa permite realizar, se puede observar que, en general, las zonas que fueron declaradas por la simulación como zonas con servicio, fueron efectivamente verificadas como tales por las medidas reales, siendo cubiertas, en la mayor parte de los puntos de medidas, por 3 o más estaciones base (puntos en color verde).

En cuanto a las medidas realizadas que no consiguieron conectividad con la red (puntos en color negro), también se verifica que caen en zonas en las que la simulación pronosticaba la falta de servicio.

Por otro lado, también se observa que en muchas de las zonas en las que, según la simulación no debería haber servicio, durante la campaña de medidas sí que se consiguió conectividad, aunque eso sí, la mayoría con un nivel de redundancia bajo (puntos en color rojo o amarillo). Esta diferencia entre la simulación y las medidas reales podría deberse a una subestimación deliberada en los parámetros de la simulación para aquellas zonas que se encontrasen en los umbrales de cobertura, o incluso al hecho de la posible presencia de alguna nueva estación base no actualizada en la propia simulación.

Finalmente, con el objeto de poder tener una cuantificación aproximada que permita comparar la simulación con las medidas reales, se presenta en la siguiente tabla los cuatro casos posibles, extrayendo los datos directamente de la Figura 50.

Tabla 17. *Cuantificación de coincidencias entre la simulación y las medidas reales obtenidas*

Número de casos en los que las pruebas de campo confirmaban cobertura, mientras que la simulación indicaba zona sin cobertura	27
Número de casos en los que las pruebas de campo confirmaban cobertura, y la simulación también indicaba zona con cobertura	146
Número de casos en los que las pruebas de campo confirmaban zona sin cobertura, y la simulación también indicaba zona sin cobertura	5
Número de casos en los que las pruebas de campo confirmaban zona sin cobertura, mientras que la simulación indicaba zona con cobertura	0

6. Conclusiones

A lo largo de los diferentes capítulos de la memoria, se han presentado los detalles de los diferentes desarrollos llevados a cabo durante la realización de este trabajo final de máster, comenzando el mismo definiendo unos objetivos claros a alcanzar, así como la planificación a cumplir para conseguirlos.

Una vez finalizado el desarrollo, se ha obtenido un prototipo totalmente funcional basado en una placa de control de bajo coste que permitió realizar la campaña de medidas prevista, recorriendo más de 700 Km a pleno funcionamiento. Por otro lado, se desarrolló una aplicación capaz de obtener los productos definidos en los requisitos iniciales, permitiendo generar un mapa con la cobertura real de la red SIGFOX, medida en más de 170 localizaciones en la isla de Gran Canaria.

La solución implementada permite pues, tal como se definió en el alcance del trabajo, disponer de una herramienta que sirva de apoyo a aquellos proyectos que necesiten conocer con exactitud cuál es el nivel de cobertura de la red SIGFOX en un emplazamiento determinado. De esta forma, se facilitan las tareas de optimización, tanto de las localizaciones más idóneas, como de las potencias máximas a transmitir por los dispositivos a instalar.

Durante la realización del trabajo, se tuvo especial control sobre el cumplimiento de la planificación prevista inicial, para poder lograr el alcance dentro de los plazos definidos. En este sentido, hay que destacar que la estrategia de compra de componentes, o el entorno de desarrollo elegido han jugado un papel decisivo para poder cumplir los plazos. Es por ello que una elección equivocada, por ejemplo a la hora de decantarse por un entorno de desarrollo con una curva de aprendizaje demasiado larga, hubiese comprometido sin duda el éxito del proyecto.

En cuanto a posibles trabajos futuros, la herramienta aquí desarrollada constituye un prototipo inicial, tanto software como hardware, y ha servido como demostrador funcional de la solución completa propuesta, manteniendo acotadas sus funcionalidades solamente al alcance definido, para poder culminar todos los objetivos dentro de plazo. Es por ello que, las posibilidades de mejora y expansión de sus funcionalidades son muy amplias.

En este sentido, acciones futuras como sustituir la placa de control elegida por otra con tecnología de bajo consumo, permitiría reducir el tamaño del prototipo e incorporar una batería interna que permitiese un funcionamiento autónomo durante un mayor tiempo. Por otro lado, también cabe una mejora importante en la ampliación del almacenamiento interno, lo que permitiría disponer de más jornadas de medidas antes de proceder a la descarga de datos.

Finalmente también cabrían innovaciones en la capacidad de visualización del mapa de servicio en tiempo real a través de dispositivo móvil, o funcionalidades extra como poder exportar los datos obtenidos a otros formatos de ficheros.

7. Glosario

ACK	Acknowledgment
ASCII	American Standard Code for Information Interchange
BLE	Bluetooth Low Energy
CPU	Central Processing Unit
CRFL	Carrier-Return Feed-Line
CSS	Chirp Spread Spectrum
DSSS	Direct-Sequence Spread Spectrum
EEPROM	Electrically Erasable Programmable Read-Only Memory
ETSI	European Telecommunications Standards Institute
GNSS	Global Navigation Satellite System
GPIO	General Purpose Input/Output
GPL	General Public License
GPS	Global Position System
IDE	Integrated Development Environment
IoT	Internet of Things
ISM	Industrial, Scientific and Medical Band
LED	Light Emitting Diode
LPWAN	Low-Power Wide-Area Network
M2M	Machine to Machine
MIT	Massachusetts Institute of Technology
NFC	Near Field Communication
NMEA	National Marine Electronics Association
PCB	Printed Circuit Board
PEC	Prueba de Evaluación Continua
PIFA	Planar Inverted-F Antenna
RPMA	Random Phase Multiple Access
RSSI	Received Signal Strength Indication
RTCM	Radio and Telephony Communications Maritime
SBAS	Satellite Based Augmentation System
SCAN	SIGFOX Coverage Analyzer
SDRAM	Synchronous Dynamic Random Access memory
SNR	Signal to Noise Ratio
SPI	Serial Peripheral Interface
SRAM	Static Random Access Memory
UART	Universal Asynchronous Receiver-Transmitter
UBX	Ublox protocol
UNB	Ultra-Narrow Band
USB	Universal Serial Bus

8. Bibliografía

- [1] K. Ashton, "That 'Internet of Things' Thing," *RFiD J.*, p. 4986, 2009.
- [2] Gartner, "Gartner Says 6.4 Billion Connected 'Things' Will Be in Use in 2016, Up 30 Percent From 2015," 2015. [Online]. Available: <http://www.gartner.com/newsroom/id/3165317>. [Accessed: 08-Jun-2016].
- [3] SIGFOX, "About SIGFOX," 2016. [Online]. Available: <http://makers.sigfox.com/>. [Accessed: 08-Jun-2016].
- [4] L. Alliance, "LoRa Technology," 2016. [Online]. Available: <https://www.lora-alliance.org/What-Is-LoRa/Technology>. [Accessed: 08-Jun-2016].
- [5] SIGFOX, "Global coverage," 2016. [Online]. Available: <http://www.sigfox.com/coverage>. [Accessed: 08-Jun-2016].
- [6] S. Andreev, O. Galinina, A. Pyattaev, M. Gerasimenko, T. Tirronen, J. Torsner, J. Sachs, M. Dohler, and Y. Koucheryavy, "Understanding the IoT connectivity landscape: a contemporary M2M radio technology roadmap," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 32–40, 2015.
- [7] M. Centenaro, L. Vangelista, A. Zanella, and M. Zorzi, "Long-Range Communications in Unlicensed Bands: the Rising Stars in the IoT and Smart City Scenarios," *IEEE Wirel. Commun.*, pp. 1–7, 2015.
- [8] Ingenu, "Ingenu's RPMA Technology," 2016. [Online]. Available: <http://www.ingenu.com/technology/rpma/>. [Accessed: 08-Jun-2016].
- [9] V. Turau, M. Witt, and M. Venzke, "Field trials with wireless sensor networks: Issues and remedies," *Second Int. Conf. Wirel. Mob. Commun. ICWMC 2006*, 2006.
- [10] J. Petäjälä, K. Mikhaylov, A. Roivainen, T. Hänninen, and M. Pettissalo, "On the Coverage of LPWANs: Range Evaluation and Channel Attenuation Model for LoRa Technology," pp. 55–59, 2016.
- [11] Semtech, "Semtech LoraMote-User Guide," 2014. [Online]. Available: http://www.semtech.com/images/datasheet/User_Guide_LoRaMote_STD.pdf. [Accessed: 08-Jun-2016].
- [12] A. Llaría, G. Terrasson, H. Arregui, and A. Hacala, "Geolocation and monitoring platform for extensive farming in mountain pastures," pp. 2420–2425, 2015.
- [13] Adeunis, "SIGFOX by ADEUNIS RF Demonstrator - Adeunis RF," 2016. [Online]. Available: <http://www.adeunis-rf.com/en/products/sigfox-ready-products/sigfox-by-adeunis-rf-demonstrator>. [Accessed: 08-Jun-2016].
- [14] Arrow, "Home of SmartEverything board," 2016. [Online]. Available: <http://www.smarteverything.it/>. [Accessed: 08-Jun-2016].
- [15] Axel-Electronica, "SmartEverything FOX User Guide," 2015. [Online]. Available: <http://www.smarteverything.it/wp-content/uploads/2015/10/SmartEverything->

- User-Guide-Ver-1.01.pdf. [Accessed: 08-Jun-2016].
- [16] Raspberry, "Raspberry Pi Products," 2016. [Online]. Available: <https://www.raspberrypi.org/products/>. [Accessed: 08-Jun-2016].
 - [17] Arduino, "Arduino - Products," 2016. [Online]. Available: <https://www.arduino.cc/en/Main/Products>. [Accessed: 08-Jun-2016].
 - [18] Wiring, "About \ Wiring," 2016. [Online]. Available: <http://wiring.org.co/about.html>. [Accessed: 08-Jun-2016].
 - [19] Open-Impulse, "GY-NEO6MV2 GPS Module," 2016. [Online]. Available: <https://www.openimpulse.com/blog/products-page/product-category/gy-neo6mv2-gps-module/>. [Accessed: 08-Jun-2016].
 - [20] U-blox, "NEO-6 u-blox GPS Modules," 2016. [Online]. Available: <https://www.openimpulse.com/blog/wp-content/uploads/wpsc/downloadables/GY-NEO6MV2-GPS-Module-Datasheet.pdf>. [Accessed: 08-Jun-2016].
 - [21] NMEA, "Marine Electronics | Marine Accessories | National Marine Electronic Association," 2016. [Online]. Available: http://www.nmea.org/content/nmea_standards/nmea_standards.asp. [Accessed: 08-Jun-2016].
 - [22] Libelium, "Sigfox module for Arduino, Waspote and Raspberry Pi - 868 MHz [XBee Socket]," 2016. [Online]. Available: <https://www.cooking-hacks.com/sigfox-module-for-arduino-waspote-raspberry-pi-intel-galileo-868-mhz-7184>. [Accessed: 08-Jun-2016].
 - [23] Free-Software-Foundation, "GNU-GPL," 2007. [Online]. Available: <https://www.gnu.org/licenses/gpl-3.0.txt>. [Accessed: 08-Jun-2016].
 - [24] Qt, "Qt - Home," 2016. [Online]. Available: <https://www.qt.io/>. [Accessed: 08-Jun-2016].
 - [25] Lubuntu, "lubuntu | lightweight, fast, easier," 2016. [Online]. Available: <http://lubuntu.net/>. [Accessed: 08-Jun-2016].
 - [26] Leaflet, "Leaflet - a JavaScript library for interactive maps," 2016. [Online]. Available: <http://leafletjs.com/>. [Accessed: 08-Jun-2016].
 - [27] JQuery, "jQuery," 2016. [Online]. Available: <http://jquery.com/>. [Accessed: 08-Jun-2016].
 - [28] OpenStreetMap, "OpenStreetMap," 2016. [Online]. Available: <http://www.openstreetmap.org>. [Accessed: 08-Jun-2016].
 - [29] GEVIC, "Situación y descripción paisajística - GRAN CANARIA - (GEVIC) Gran Enciclopedia Virtual Islas Canarias," 2016. [Online]. Available: http://www.gevic.net/info/contenidos/mostrar_contenidos.php?idcat=37&idcap=74&idcon=378. [Accessed: 08-Jun-2016].

9. Anexos

A.1 Repositorio del código fuente

Debido a la cantidad de código fuente generado durante la realización de los diferentes desarrollos, se ha optado por incluir el mismo en un repositorio de acceso público dedicado a este proyecto (GitHub). Este anexo presenta los enlaces a dichos repositorios, de tal manera puedan ser consultados o descargados en línea por aquellos lectores que deseen profundizar en el código desarrollado.

Repositorio General

Enlace: <https://github.com/scan-tfm?tab=repositories>

Este enlace presenta todos los repositorios disponibles del proyecto.

Repositorio del firmware del SCAN-device

Enlace: <https://github.com/scan-tfm/SCAN-device>

Este repositorio contiene el firmware que ha sido desarrollado e incluido en el microcontrolador de la placa Arduino. Está desarrollado en *Wiring* y para compilarlo basta con descargarlo y abrirlo bajo el propio IDE de Arduino.

Para la comunicación con el módulo de comunicaciones se hace uso de las librerías proporcionadas por el propio fabricante y que deben ser instaladas en el IDE de Arduino antes de la compilación. El enlace para la descarga de dichas librerías es el siguiente:

http://www.cooking-hacks.com/media/cooking/images/documentation/tutorial_kit_sigfox/arduinoSigfox-library_v1_1.zip

Repositorio de la aplicación SCAN-tool

Enlace: <https://github.com/scan-tfm/SCAN-tool>

Este repositorio contiene todo el código fuente que compone la aplicación SCAN-tool. Contiene tanto los ficheros de cabecera e implementación de las diferentes clases desarrolladas, como los ficheros que describen la interfaz gráfica de usuario (ficheros con extensión .ui) generados desde el IDE de QT.

Para compilar el código es necesario descargar e instalar el entorno de desarrollo de QT para generación de software bajo licencia GPL (<https://www.qt.io/download>), e instalarlo en un sistema operativo Ubuntu o similar de 32 bits (la versión actual se desarrolló y probó sobre Ubuntu 14.04).

Una vez instalado el entorno, basta con descargar todos los ficheros del repositorio y crear un proyecto nuevo a partir del fichero de proyecto **SCAN.pro**.

Una vez compilado, se debe descargar la librería jQuery *jquery-2.1.1.min.js* (<https://jquery.com/download/>) y copiarla al directorio de la aplicación compilada. Esta librería es usada para la generación del mapa de servicio.

Repositorio de ficheros del SCAN-server

Enlace: <https://github.com/scan-tfm/SCAN-server>

Este repositorio contiene el código fuente de la página alojada en el SCAN-server encargada de recibir y almacenar los mensajes de *callback* procedentes del *backend* de SIGFOX con los datos de los mensajes retransmitidos por todas las estaciones base que recibieron los mensajes transmitidos por el SCAN-device.

Repositorio de ficheros de la página web que almacena el Mapa de Servicio

Enlace: <https://github.com/scan-tfm/SCAN-map>

Este repositorio contiene la página web que genera el mapa que se encuentra alojado en el servidor de acceso público y que muestra los resultados de las medidas de campo realizadas en la isla de Gran Canaria (<http://coverage.planetamaxwell.com>)

Repositorio de utilidades

Enlace: <https://github.com/scan-tfm/SCAN-utils>

Aquí se encuentra alojada una pequeña utilidad desarrollada en *Wiring* para cargar los parámetros por defecto iniciales en la EEPROM del SCAN-device. La idea de esta utilidad es que antes del primer uso del SCAN-device, y por tanto previa carga del firmware de control, se compile y se cargue esta utilidad para resetear todos los contadores alojados en la EEPROM y para cargar los valores por defecto asociados a cada una de las posibles opciones de potencias de transmisión (14, 7 y 0 dBm).

A.2 Manual de usuario del prototipo SCAN-device

En este anexo se presenta un breve manual de usuario que pueda servir de referencia inicial para el correcto manejo del prototipo.

Para la localización de los diferentes pulsadores e indicadores, referirse a la siguiente figura.



Encender el prototipo

El SCAN-device se alimenta mediante 5 V a través del conector USB de la placa de control Arduino, accesible desde la abertura superior de la carcasa. Esta alimentación puede proceder de la conexión a un ordenador, una batería externa o cualquier otra fuente estabilizada capaz de proporcionar como mínimo 200 mA en carga. En caso de usar una batería externa, se recomienda emplear una con capacidad mínima de 2500 mAh, lo que permitiría una autonomía de unas 15 horas.

Hay que asegurarse antes de encender el equipo que la antena se encuentra correctamente conectada y que el *microswitch* de configuración del puerto serie se encuentra en posición XBEE (accesible desde la abertura inferior de la carcasa).

Una vez el SCAN-device ha sido conectado, se verá como el LED verde "Tx Hig" se ilumina y el LED "GPS Lock" comienza a parpadear.

Seleccionar la potencia que se desea transmitir

Presionando el pulsador "Select Power", se verá como los LEDs "Tx Low", "Tx Med" y "Tx Hig" se iluminan secuencialmente con cada pulsación. Seleccionar la potencia deseada.

Realizar una transmisión sin ACK

IMPORTANTE: antes de realizar una transmisión, asegurarse de que el módulo de comunicaciones SIGFOX tiene la antena correctamente conectada. En caso contrario, la potencia reflejada puede dañar la etapa de salida del módulo.

El dispositivo no permite realizar ninguna transmisión, a menos que el GPS esté dando posiciones correctas (LED "GPS Lock" continuamente iluminado). Para proceder a la transmisión, actuar sobre el pulsador "Tx". En caso de que el GPS no esté dando posiciones válidas (LED "GPS Lock" parpadeando), el prototipo no realizará la transmisión.

Una vez se ha iniciado la transmisión, el LED "TX On" se iluminará durante aproximadamente 12 segundos.

Si la transmisión se ha completado con éxito, el LED "TX On" se apagará. En caso de que haya habido algún error, se iluminarán simultáneamente los LEDs "Tx Low", "Tx Med" y "Tx Hig".

Realizar una transmisión con ACK

IMPORTANTE: antes de realizar una transmisión, asegurarse de que el módulo de comunicaciones SIGFOX tiene la antena correctamente conectada. En caso contrario, la potencia reflejada puede dañar la etapa de salida del módulo.

El dispositivo no permite realizar ninguna transmisión, a menos que el GPS esté dando posiciones correctas (LED "GPS Lock" continuamente iluminado). Para proceder a la transmisión con ACK, actuar sobre el pulsador "Tx" mientras se mantiene pulsado el botón "Remote".

En caso de que el GPS no esté dando posiciones válidas (LED "GPS Lock" parpadeando), el prototipo no realizará la transmisión.

Una vez se ha iniciado la transmisión, el LED "TX On" se iluminará durante aproximadamente 45 segundos.

Si la transmisión se ha completado con éxito y se ha recibido correctamente el ACK, el LED "TX On" parpadeará durante 10 segundos con una cadencia de 2 Hz. En caso de que haya habido algún error en la recepción del ACK, el led "TX On" simplemente se apagará.

Memoria EEPROM llena

Para indicar que la memoria se encuentra llena y que no se pueden realizar más transmisiones hasta que se descarguen los datos y se borren, el prototipo enciende los LEDs "Tx Hig" y "Tx Low" y apaga el LED "Tx Med".

Borrar la EEPROM

Para proceder al borrado de los datos almacenados en la memoria interna, mantener presionado el pulsador “Erase EEPROM” durante 5 segundos. El prototipo indicará que la memoria ha sido borrada apagando los LEDs los LEDs “Tx Low”, “Tx Med” y “Tx Hig”.

Pasar el equipo a Remoto para descargar datos y recibir comandos

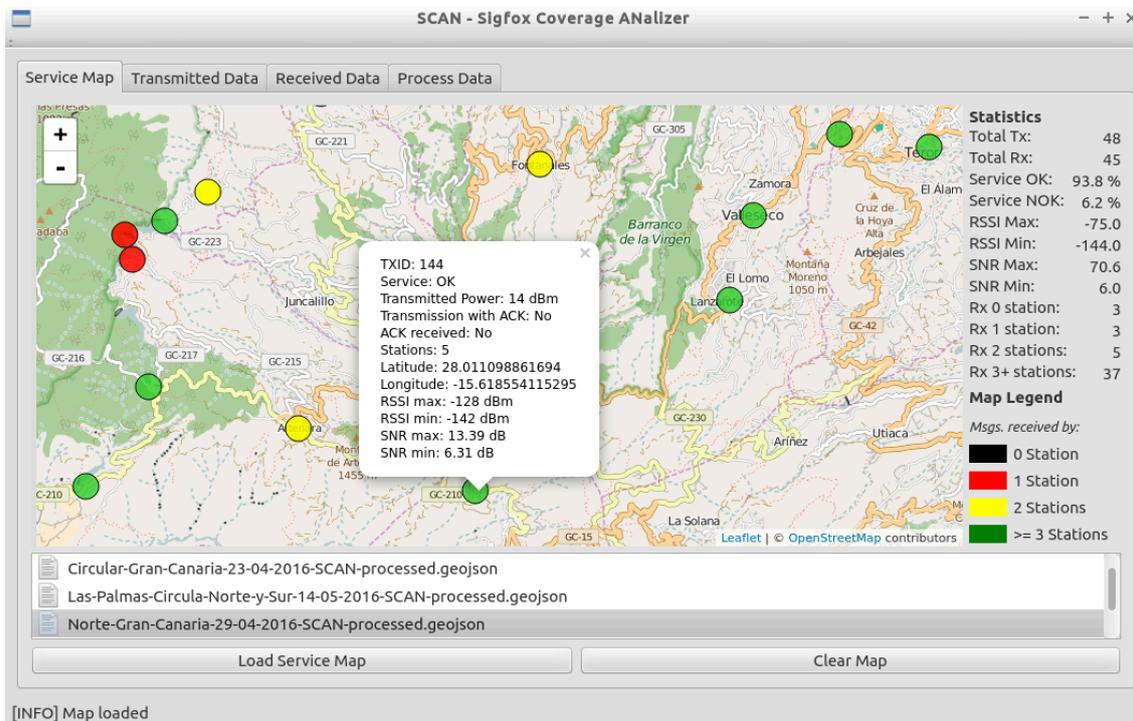
- Desconectar la alimentación del SCAN-device.
- Configurar el *switch* del puerto serie en posición USB.
- Conectar el SCAN-device a un ordenador a través del cable USB.
- Abrir el puerto, usando una aplicación de terminal serie o la propia aplicación SCAN-tool. El puerto serie debe estar configurado como 9600 baudios, 8 bits de datos, no paridad y 1 bit de stop (9600N81).
- Pulsar el botón “Remote”. Veremos como el LED “Tx On” comenzará a parpadear con una cadencia de 1 Hz, indicando que el prototipo está listo para recibir comandos por el puerto serie. En este estado, el prototipo no puede realizar ninguna transmisión.

La lista y formato de los comandos disponibles puede ser consultada en el capítulo 3 de la presente memoria.

A.3 Manual de usuario de la aplicación SCAN-tool

En este anexo se presenta un breve manual de usuario que pueda servir de referencia inicial para el correcto manejo del software SCAN-tool. Se seguirá la secuencia típica desde que se importan los ficheros hasta que se procesan y se obtiene el mapa

Referirse a la siguiente figura para los detalles de las diferentes opciones.



Importar datos del SCAN-device

- Conectar el SCAN-device al ordenador tal como se describe en el Anexo A.2
- En la pestaña "Transmitted Data" pulsar el botón "Get from Device". Se abrirá una nueva ventana.
- En la nueva ventana de diálogo habilitada, seleccionar el puerto serie donde se ha conectado el SCAN-device y pulsar "Open port".
- Pulsar, el botón "Download". Tras un breve tiempo de descarga de datos, un nuevo fichero aparecerá en la sección "Raw Data Files".
- Seleccionar el fichero descargado y pulsar el botón "Import File".
- Un nuevo fichero importado aparecerá en la sección Imported Files.
- Se puede renombrar o borrar cualquier fichero mediante las opciones habilitadas para ello.

Importar datos del SCAN-server

- En la pestaña "Received Data" pulsar el botón "Get from Server". Se abrirá una nueva ventana.
- En la nueva ventana de diálogo habilitada, comprobar que la dirección del SCAN-server desde donde se descargarán los datos es correcta.

- Seleccionar el dispositivo sobre el que queremos descargar los datos.
- Pulsar, el botón “Download”. Tras un breve tiempo de descarga de datos, un nuevo fichero aparecerá en la sección “Raw Data Files”.
- Seleccionar el fichero descargado y pulsar el botón “Import File”.
- Un nuevo fichero importado aparecerá en la sección Imported Files.
- Se puede renombrar o borrar cualquier fichero mediante las opciones habilitadas para ello.

Procesar los Datos

- En la pestaña “Process Data”, seleccionar uno o varios ficheros de la sección “Transmitted Data” y un solo fichero de la sección “Received Data”.
- Pulsar el botón “Process Files”.
- Un nuevo fichero importado aparecerá en la sección de ficheros procesados.
- Se puede renombrar o borrar cualquier fichero mediante las opciones habilitadas para ello.

Visualizar el mapa con los datos procesados

- En la pestaña “Service Map”, seleccionar el fichero que queremos mostrar en el mapa.
- Se puede navegar por el mapa y pulsar sobre los diferentes puntos de servicio para obtener más información relacionada con los mismos.
- Un nuevo mapa se cargará con los datos procesados del fichero seleccionado y nuevas estadísticas asociadas a ese fichero se presentarán en la sección “Statistics”.

Cabe mencionar que el programa SCAN-tool dispone de un fichero de configuración en formato texto editable (config.ini), desde donde se pueden definir los siguientes parámetros:

- lastLat: latitud donde queremos que aparezca centrado el mapa inicial.
- lastLon: longitud donde queremos que aparezca centrado el mapa inicial.
- lastZoom: zoom inicial del mapa.
- server: dirección web o IP desde donde descargar los datos del SCAN-server
- deviceX: lista de números de serie de dispositivos dados de alta en la red SIGFOX: X representa un número entero a partir de 1 (device1, device2, etc.)
- rxRawData: extensión del fichero raw descargado del SCAN-server.
- rxImportedData: extensión del fichero importado del SCAN-server.
- txRawData: extensión del fichero raw descargado del SCAN-device.
- txImportedData: extensión del fichero importado del SCAN-device.
- processedData: extensión del fichero geojson obtenido al procesar.

A.4 Coste de componentes

En este anexo se presentan, a modo de referencia, el coste aproximado de los componentes empleados para implementar el prototipo.

COMPONENTE	COSTE (€)
Placa Arduino Uno	12
Módulo GPS	28
Módulo de Comunicaciones SIGFOX	55
Placa XBee Shield	15
Diodos LEDES	2
Pulsadores	2
Resistencias	0.50
Cables y material de soldadura	2
Placa de prototipos perforada	10
Carcasa envolvente	12
TOTAL	138.5 €

A.5 Esquema de conexiones

