



Herramienta web para la monitorización de flota de vehículos

Francisco Javier Romero Conde

Máster Universitario en Ingeniería de Telecomunicación UOC-URL

Área de Telemática

Nombre Consultor: José López Vicario

Nombre Profesor/a responsable de la asignatura: Xavi Vilajosana Guillen

9/06/2019



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/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Herramienta web para la monitorización de flota de vehículos</i>
Nombre del autor:	<i>Francisco Javier Romero Conde</i>
Nombre del consultor/a:	<i>José López Vicario</i>
Nombre del PRA:	<i>Xavi Vilajosana Guillen</i>
Fecha de entrega (mm/aaaa):	06/2019
Titulación:	<i>Máster Universitario en Ingeniería de Telecomunicación UOC-URL</i>
Área del Trabajo Final:	<i>Telemática</i>
Idioma del trabajo:	<i>Castellano</i>
Palabras clave	<i>GPS, LPWAN, REST, SPA, Google Maps</i>

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

Hoy en día disponemos de una gran cantidad de objetos conectados que envían gran cantidad de información a través de la red.

En este TFM se realiza un estudio de como la información de una flota de vehículos (objetos IOT) llega al ser humano para su correspondiente monitorización y entendimiento, es decir, el camino por el que se desplaza la información hasta la llegada al ser humano.

Para ello se ha realizado un estudio del GPS y las tecnologías LPWAN más relevantes en la actualidad como son Sigfox, LoRa/LoRaWAN que trabajan en la banda ISM no licenciadas y las derivadas de LTE, NB-IoT y LTE-M en las bandas licenciadas. Así mismo, se lleva a cabo una comparativa y análisis donde se indica qué tecnología es mejor en determinados casos.

Una vez enviada por la red esta información es necesario almacenarla. Esta tarea se realiza mediante un servidor de base de datos.

Con todo esto, en este TFM y con la ayuda de distintas tecnologías web y framework derivados de ellas, se ha desarrollado la herramienta web para la monitorización de una flota de vehículos que muestra la geolocalización y posiciona al vehículo en el mapa utilizando la API de Google Maps, además de mostrar otra información de los vehículos de la flota para que finalmente el cliente (dispositivo móvil, tablet, pc, ...) pueda consultar esta información almacenada a través de un servidor web mediante peticiones HTTP y REST.

Abstract (in English, 250 words or less):

Today we have a large number of connected objects that send a large amount of information through the network.

In this TFM a study is made on how the information of a fleet of vehicles (IOT objects) reaches the human being for its corresponding monitoring and understanding, that is, the way in which the information travels until the arrival to the human being.

To this end, a study of GPS and the most relevant LPWAN technologies such as Sigfox, LoRa / LoRaWAN that work in the unlicensed ISM band and those derived from LTE, NB-IoT and LTE-M in the licensed bands has been carried out. Likewise, a comparison and analysis is carried out indicating which technology seems to be the best in each case.

Once it has been sent through the network, this information is necessary to be stored. This task is done through a database server.

With all this, in this TFM and with the help of different web technologies and framework derived from them, the web tool has been developed for the monitoring of a vehicle fleet that shows the geolocation and locates the vehicle on the map using the API Google Maps, apart of showing other information about the vehicles of the fleet so that, the client (mobile device, tablet, pc, ...) can consult this information stored through a web server by means of HTTP and REST requests.

Índice

1.	Introducción.....	1
1.1.	Contexto y justificación del Trabajo.....	1
1.2.	Objetivos del Trabajo	1
1.3.	Enfoque y método seguido	2
1.4.	Planificación del Trabajo.....	3
1.5.	Breve resumen de productos obtenidos	6
1.6.	Breve descripción de los otros capítulos de la memoria.....	6
2.	Estado del arte.....	7
2.1.	El Sistema de Posicionamiento Global o GPS	7
2.2.	El Internet de las cosas (IoT).....	9
2.3.	LPWAN	11
2.3.1.	Sigfox	12
2.3.2.	LoRa/LoRaWAN	16
2.3.3.	LTE: NB-IoT y LTE-M.....	22
2.3.4.	Comparativa.....	28
2.3.5.	Conclusiones	32
2.4.	Almacenamiento y acceso a la información	33
2.4.1.	Servidor	33
2.4.2.	Servidor de base de datos	34
2.4.3.	Servidor web	35
2.4.4.	Arquitectura REST.....	38
2.4.5.	Conclusiones	41
2.5.	Tecnologías para aplicaciones web del lado del cliente	41
2.5.1.	HTML.....	42
2.5.2.	CSS	42
2.5.3.	JavaScript	42
2.5.4.	AJAX	42
2.5.5.	SPA (Single Page Application)	45
2.5.6.	Framework	47
2.5.7.	Conclusiones	50
3.	Requisitos del sistema, diseño e implementación	51
3.1.	Requisitos y configuración	51
3.1.1.	Requisitos de red, software, funcionales y de diseño	51
3.1.2.	Google Maps.....	52
3.2.	Diseño e implementación.....	54
3.2.1.	Arquitectura completa del sistema	54
3.2.2.	Estado de la red LPWAN	55
3.2.3.	Diseño del modelo de datos	55
3.2.4.	Diagrama de estados y navegación	57
3.2.5.	Esqueleto de la aplicación web	58
3.2.6.	Librerías y dependencias	60
3.2.7.	Presentación de la herramienta web.....	61
3.3.	Presupuesto.....	63
3.3.1.	Recursos materiales.....	63
3.3.2.	Software de desarrollo	63
3.3.3.	Desarrollo de la Herramienta web	63
3.3.4.	Coste total de desarrollo	64
3.3.5.	Costes en producción.....	64

3.3.6. Conclusión	65
4. Verificación.....	66
4.1. Prueba 1. Inicio de sesión incorrecto	66
4.2. Prueba 2. Límite de número de vehículos consultados a la API REST	67
4.3. Prueba 3. Movimiento de un vehículo al cabo de 1 minuto	67
4.4. Prueba 4. Cambio de idioma	68
4.5. Prueba 5. Diseño responsive.....	69
5. Conclusiones.....	72
6. Glosario	73
7. Bibliografía	75
8. Anexos.....	79
8.1. Ejemplo Sigfox. Regulación europea 1% en la banda ISM	79
8.2. Ejemplo LoRaWAN. Regulación europea del 1% en la banda ISM	79
8.3. El Spreading Factor (SF) de LoRa	80
8.4. MCL (Pérdida de acoplamiento máxima).....	80
8.5. Registro Google Cloud Platform para usar la API de Google Maps	81
8.6. Instalación de dependencias y otras opciones del proyecto	82

Índice de figuras

Figura 1. Diagrama de Gantt PEC2.....	5
Figura 2. Determinación de la posición del receptor GPS [1]	8
Figura 3. Módulo GPS	9
Figura 4. Modelo de referencia de IoT [2]	10
Figura 5. Velocidad de datos vs. Área de cobertura de tecnologías inalámbricas [4].....	12
Figura 6. Sigfox ancho de banda [6].....	13
Figura 7. Tipos de suscripciones de Sigfox (mensajes de enlace ascendente / descendente) [6]	13
Figura 8. Arquitectura Sigfox [6]	14
Figura 9. Cobertura mundial Sigfox [6] [7]	15
Figura 10. Módulo Sigfox 868 para Europa con antena de 4.5 dBi [11]	16
Figura 11. Protocolo de comunicación y arquitectura del sistema de LoRaWAN [12].....	17
Figura 12. Arquitectura LoRa [12]	19
Figura 13. Cobertura mundial LoRa [15].....	20
Figura 14. Módulo transceptor LoRa SX1272 y Concentrador LoRa [16].....	21
Figura 15. Módulo RN2433 LoRaWAN para Europa clase A con antena [15] ..	21
Figura 16. Modos de operación NB-IoT [23].....	24
Figura 17. Arquitectura LTE [24].....	25
Figura 18. Arquitectura LTE para NB-IoT [25]	26
Figura 19. Cobertura mundial NB-IoT/LTE-M [26].....	26
Figura 20. Módulo NB-IoT/LTE-M con sus antenas y apertura para nano-SIM de 3.4 dBi [29].....	28
Figura 21. Comparativa LPWAN – Gráfico de tecnologías diseñadas para IoT [31].....	32
Figura 22. Base de datos relacional vs base de datos no relacional documental [34].....	35
Figura 23. Diagrama cliente-servidor vía Internet	36
Figura 24. Petición y respuesta HTTP	37
Figura 25. Arquitectura REST	39
Figura 26. Ejemplo de respuesta API REST en formato JSON con HATEOS ..	40
Figura 27. Tecnologías agrupadas bajo el concepto de AJAX [36].....	43
Figura 28. Comparación gráfica del modelo tradicional de aplicación web y del nuevo modelo propuesto por AJAX [36]	44
Figura 29. Comparación entre las comunicaciones síncronas de las aplicaciones web tradicionales y las comunicaciones asíncronas de las aplicaciones AJAX [36].....	45
Figura 30. Ciclo de vida de una página web tradicional vs ciclo de vida de una SPA [38]	46
Figura 31. Arquitectura página web tradicional vs página SPA [38]	47
Figura 32. De izquierda a derecha, logos de los framework React, VueJS y Angular [39].....	47
Figura 33. Zona de movimiento de los vehículos de la flota	51
Figura 34. Código básico Mapa + Marcador + InfoWindow	53
Figura 35. Resultado del código de la Figura 34	54
Figura 36. Arquitectura completa del sistema.....	54
Figura 37. Ejemplo JSON modelo de un autobús	56

Figura 38. Ejemplo JSON modelo de un usuario	57
Figura 39. Diagrama de estados y navegación de la herramienta web	57
Figura 40. Esqueleto de la aplicación web.....	59
Figura 41. Pantalla de Inicio de sesión	61
Figura 42. Pantalla Tabla de vehículos	62
Figura 43. Mapa con la geolocalización y estado de los vehículos	62
Figura 44. Mapa con ventana de información de un vehículo activa	62
Figura 45. Falta un campo requerido	66
Figura 46. El usuario o la contraseña son incorrectas	66
Figura 47. Llamada a la API con un máximo de 25 vehículos	67
Figura 48. Llamada a la API para consultar todos los vehículos	67
Figura 49. Posición de un vehículo "antes"	68
Figura 50. Posición de un vehículo "después de 1 minuto"	68
Figura 51. Herramienta web en español	69
Figura 52. Herramienta web en inglés	69
Figura 53. Simulación responsive en un dispositivo smartphome Android.....	70
Figura 54. Simulación responsive en un dispositivo Tablet iPad.....	71
Figura 55. Simulación responsive en un dispositivo smartphome iOS	71
Figura 56. Calculadora LoRa	79
Figura 57. Cálculo MCL [49]	81
Figura 58. Habilitar Google Maps Platform	81
Figura 59. Google Cloud Platform. Credenciales para Google Maps JavaScript API.....	82

Índice de tablas

Tabla 1. Tareas e hitos PEC2.....	4
Tabla 2. Comparativa LPWAN - Características técnicas Sigfox vs LoRa/LoRaWAN vs LTE-M vs NB-IoT	29
Tabla 3. Comparativa LPWAN – Coste por usuario [30].....	30
Tabla 4. Comparativa LPWAN – Costes de infraestructura [4]	31
Tabla 5. Modelo de los datos para un autobús.....	56
Tabla 6. Modelo de datos de un usuario	56
Tabla 7. Costes Recursos materiales	63
Tabla 8. Costes Software de desarrollo	63
Tabla 9. Coste Desarrollo de la Herramienta Web	64
Tabla 10. Coste total	64
Tabla 11. Costes fijos de en producción	64
Tabla 12. Costes variables en producción	64

Índice de ecuaciones

Ecuación 1. Cálculo tasa de bit LoRa [14]	80
Ecuación 2. Pérdida de acoplamiento máxima [49]	80
Ecuación 3. Sensibilidad del receptor [49]	80
Ecuación 4. Ruido [49]	80

1. Introducción

1.1. Contexto y justificación del Trabajo

En los últimos años ha surgido el concepto del Internet de las Cosas (IoT, “Internet of Things”). En la actualidad existen multitud de tecnologías de red en el sector IoT.

IoT es una idea en la cual el modelo de Internet se convierte en un modelo enfocado a una conectividad total, donde todos los objetos pueden estar conectados a la red.

Este modelo, se ha convertido hoy en día en uno de los temas más importantes de investigación y de negocio en las empresas.

Dicho crecimiento radica en el número de dispositivos conectados a Internet que actualmente es mucho mayor al número de habitantes del planeta. Es decir, existe un inmenso número de objetos que pueden estar conectados a Internet a través de diferentes tipos de redes (LTE, LoRa, Sigfox, ...). De esta forma se consolida el hecho de que la mayor parte de la información de la red es generada por objetos conectados a Internet.

En el caso en el que se centra este trabajo sobre flotas de vehículos. Una flota de vehículos es el conjunto de automóviles de los que dispone una empresa. Desde vehículos industriales usados para gestiones logística, pasando por otros destinados a los comerciales e incluso los coches de uso diario de los trabajadores, o simplemente bicicletas o patinetes eléctricos. Además, dentro de una flota pueden existir vehículos de distintas características y condiciones.

Hoy en día existen empresa que dan multitud de servicios mediante la utilización de vehículos agrupados en flotas. Estos vehículos necesitan una gestión para su control y seguimiento. Para ello es imprescindible el uso de un dispositivo receptor GNSS (por lo general GPS) que habilita la capacidad al vehículo de estar geolocalizado en todo momento.

Mediante el estudio realizado y la implementación de la herramienta diseñada en este trabajo fin de máster se puede gestionar una amplia cantidad de información entorno a los vehículos de una flota o de cualquier tipo y cuál es el camino, es decir, la red por la que viaja la información hasta que se presenta para que el ser humano sea capaz de visualizarla.

1.2. Objetivos del Trabajo

Los objetivos principales de este trabajo fin de máster son:

- Realizar un estudio y comparativa de las distintas tecnologías existentes en la actualidad que pueden llevar la información desde el vehículo hacia el servidor que “alimenta” a la herramienta web.
 - Funcionamiento del GPS

- Internet de las cosas, IoT
- Redes de área amplia y baja potencia, LPWAN
 - Sigfox
 - LoRa/LoRaWAN
 - NB-IoT y LTE-M
 - Comparativa
- Desarrollar una herramienta web para la:
 - Geolocalización de vehículos de la flota
 - Se muestra la geolocalización de los vehículos de la flota mediante la representación en el mapa.
 - Monitorización de vehículos de la flota
 - Se diferencia el estado de los vehículos según esté:
 - Parado
 - En movimiento
 - Se muestra información propia del vehículo. Tipo de combustible, tipo de vehículo, ...

1.3. Enfoque y método seguido

La elaboración del estado del arte relativo a las tecnologías de comunicación dentro del sector IoT se ha basado en el estudio de diferentes trabajos y búsqueda de investigación y en el análisis de la información aportada por los grupos y compañías que regulan las distintas tecnologías de comunicación estudiadas.

El desarrollo de la herramienta web tiene un enfoque de gestión y se ha seguido una metodología en cascada o secuencial ya que los requisitos de la herramienta son fijos. Además, promueve una metodología de trabajo efectiva: Definir antes que diseñar, diseñar antes que codificar. Esta metodología se basa en 5 etapas:

1. **Análisis de requisitos:** En esta etapa se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir.
2. **Diseño:** Tiene como objetivo definir la estructura de la solución identificando grandes módulos y sus relaciones. Con ello se define la arquitectura de la solución elegida.
3. **Implementación:** Es la fase donde se implementa el código fuente, haciendo uso de prototipos, así como pruebas y ensayos para corregir errores.
4. **Verificación:** Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente. Se buscan sistemáticamente y se corrigen todos los errores antes de ser entregado al usuario final.
5. **Mantenimiento:** Esta última etapa se dedica al mantenimiento del software ya que al utilizarlo como usuario final puede ser que no cumpla con todas nuestras expectativas o tener posibles errores que solucionar.

1.4. Planificación del Trabajo

La planificación del trabajo se ha visto modificada a lo largo de las entregas de las PEC.

Durante la PEC1 principalmente solo se planificó la entrega de la propia herramienta web.

A menudo que avanzaba el estudio y sobre todo en el caso de la PEC2 se actualizó el bloque de contenidos del trabajo adecuándolo a las indicaciones del consultor y extendiendo el contenido tanto teórico como práctico como se muestra en la Tabla 1 y en la Figura 1.

En cuanto a la PEC3 se ha mantenido la planificación descrita en la PEC2 y se han realizado las mejoras (y correcciones indicadas por el consultor), destacando principalmente las siguientes:

- Se añade pequeño resumen de las características más destacadas al final de cada tecnología LPWAN (y además se indica que se habla con más detalle en el capítulo 2.7 Comparativa).
- Se añaden ejemplos de módulos GPS, Sigfox, LoRa/LoRaWAN y LTE-M y NB-IoT (Libelium).
- Se actualiza el requisito impuesto por legislación europea del uso máximo del ciclo de trabajo del 1% de las bandas ISM (Sigfox y LoRaWAN).
- Se añaden los anexos de la regulación del 1% a ISM para Sigfox y LoRa (calculadora LoRa).
- Se añade comentario sobre el coste real de la infraestructura de NB-IOT y LTE-M ya que solo sería necesario adaptar las estaciones base.
- Se realiza una conclusión al final de cada subcapítulo.
- Se continua con la planificación correctamente tal y como se especifica en la PEC2.

Nombre	Fecha de inicio	Fecha de fin	Duración
Planificación	28/02/2019	06/03/2019	7
Entrega PEC1	06/03/2019	06/03/2019	0
Estado del arte	07/03/2019	27/03/2019	21
Funcionamiento GPS	07/03/2019	07/03/2019	1
Introducción IoT	08/03/2019	08/03/2019	1
Redes LPWAN	09/03/2019	27/03/2019	19
Introducción LPWAN	09/03/2019	09/03/2019	1
La tecnología Sigfox	10/03/2019	14/03/2019	5
La tecnología LoRa/LoRaWAN	15/03/2019	19/03/2019	5
La tecnología LTE-M	20/03/2019	23/03/2019	4
La tecnología NB-IoT	23/03/2019	26/03/2019	4
Comparativa red LPWAN	27/03/2019	27/03/2019	1
Almacenamiento de la información	28/03/2019	30/03/2019	3
API REST	31/03/2019	01/04/2019	2
Estudio tecnologías web	02/04/2019	03/04/2019	2
Estudio API de Google Maps	04/04/2019	11/04/2019	8
El mapa de Google Maps	04/04/2019	09/04/2019	6
Marcadores de Google Maps	10/04/2019	11/04/2019	2
Requisitos y configuración	12/04/2019	20/04/2019	9
Configuración entornos de desarrollo	12/04/2019	14/04/2019	3
Configurar modelos de datos	15/04/2019	18/04/2019	4
Añadir librerías y dependencias	19/04/2019	19/04/2019	1
Registro y API Key Google Cloud Platform	20/04/2019	20/04/2019	1
Entrega PEC2	24/04/2019	24/04/2019	0
Diseño e implementación	21/04/2019	19/05/2019	29
Simulación API REST	21/04/2019	24/04/2019	4
Simulación llamada a la API para obtener datos	25/04/2019	01/05/2019	7
Implementación de los datos	02/05/2019	08/05/2019	7
Añadir capa de presentación de Google Maps	09/05/2019	10/05/2019	2
Añadir capa de presentación de los datos al mapa	11/05/2019	19/05/2019	9
Verificación	20/05/2019	22/05/2019	3
Entrega PEC3	22/05/2019	22/05/2019	0
Memoria final	23/05/2019	09/06/2019	18
Revisión	23/05/2019	31/05/2019	9
Conclusión	01/06/2019	04/06/2019	4
Bibliografía Glosario y Anexos	05/06/2019	09/06/2019	5
Entrega de la Memoria final	09/06/2019	09/06/2019	0
Presentación	10/06/2019	16/06/2019	7
Entrega presentación	16/06/2019	16/06/2019	0
Tribunal TFM	17/06/2019	17/06/2019	0

Tabla 1. Tareas e hitos PEC2

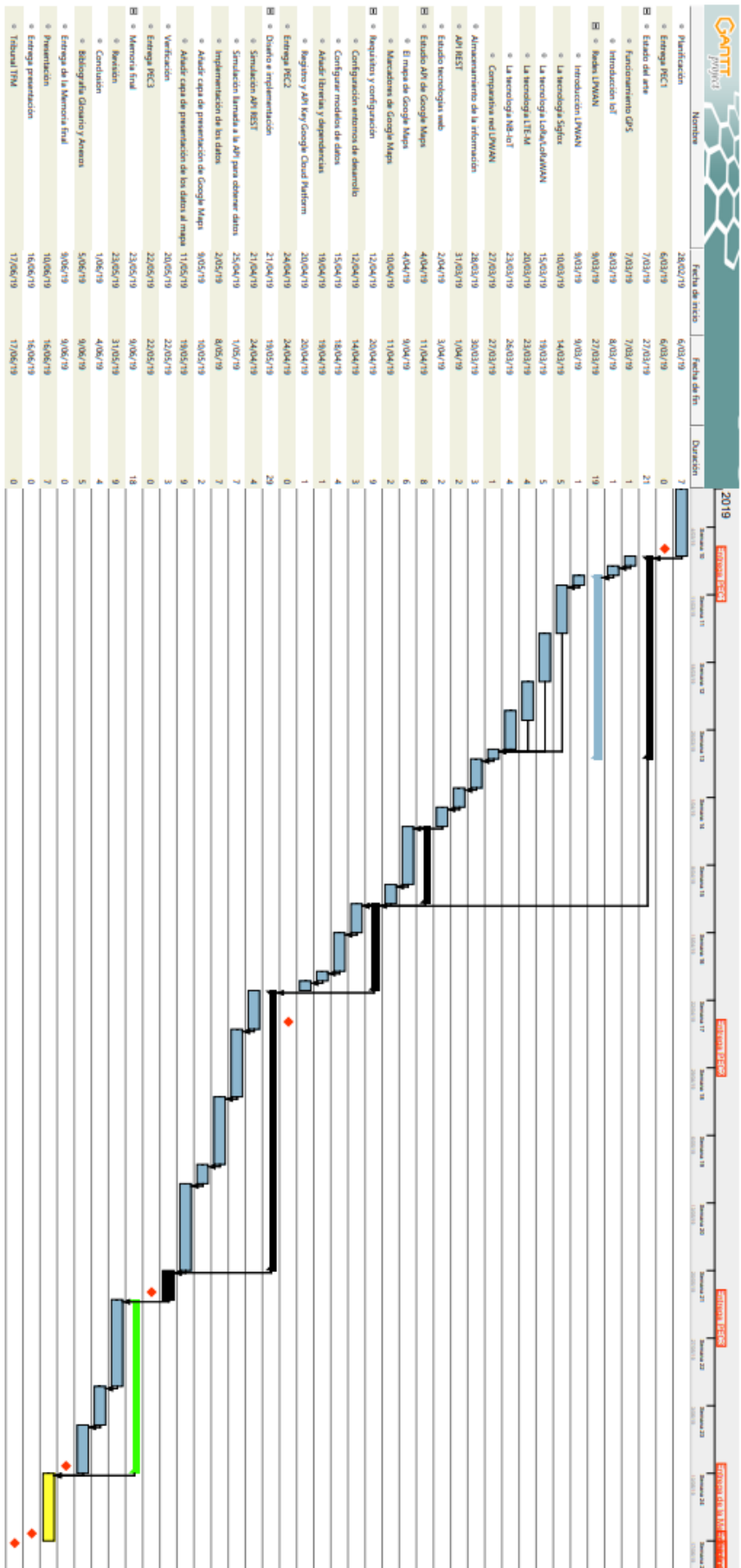


Figura 1. Diagrama de Gantt PEC2

1.5. Breve resumen de productos obtenidos

- Funcionamiento del GPS
- Estudio de LPWAN redes de área amplia de baja potencia
 - Sigfox
 - LoRa/LoRaWAN
 - LTE: NB-IoT y LTE-M
 - Comparativa de redes de área amplia
- Herramienta web que permite al usuario visualizar toda la información acerca de los distintos vehículos que componen una o varias flotas. De esta manera el usuario puede obtener la geolocalización de los vehículos además de otro tipo de información.

1.6. Breve descripción de los otros capítulos de la memoria

Capítulo 2: El estado del arte que contiene el estudio del:

- Funcionamiento del GPS.
- Introducción al IoT
- Estudio de las redes LPWAN más importantes como son Sigfox, LoRa/LoRaWAN y las variantes de LTE: LTE-M y NB-IoT, además de una comparativa.
- Almacenamiento y acceso a la información. Dónde se almacenan los datos y las herramientas o páginas web y como se accede mediante HTTP y la arquitectura API REST.
- Tecnologías para desarrollar aplicaciones web del lado del cliente. Se describen las principales tecnologías web HTML, CSS y JavaScript, así como los framework más populares.

Capítulo 3: Requisitos del sistema, diseño e implementación. Se expone los distintos requisitos del sistema, así como la configuración, implementación y diseño de la herramienta web.

Capítulo 4: Verificación. A través de distintas pruebas realizadas a la herramienta web se comprueba si correcto funcionamiento y que se adecúa a los requisitos funcionales y de diseño.

Capítulo 5: Conclusión. Qué es lo que se ha aprendido realizando este TFM, Como se ha llegado a los objetivos, cambios en la planificación y posibles mejoras y líneas futuras de trabajo.

Capítulo 6: Glosario. Definición de los términos y acrónimos más relevantes utilizados dentro de la Memoria.

Capítulo 7: Bibliografía. Referencias bibliográficas utilizado en la memoria.

Capítulo 8: Anexos. Se han añadido pequeños ejemplos de cálculos y descripciones extra para mejorar el entendimiento de los demás apartados.

2. Estado del arte

En este apartado se explica el funcionamiento del sistema de geolocalización GPS, imprescindible para la geolocalización precisa de vehículos en tiempo real y se presenta una introducción al Internet de las cosas (IoT), y las tecnologías LPWAN más destacadas, sus características, arquitectura y comparativa entre ellas.

Además, se caracterizan los distintos elementos para el almacenamiento y acceso a la información y se estudian las distintas tecnologías existentes para el desarrollo de la herramienta web.

Al final de cada subapartado tras su estudio y análisis se llega a las conclusiones más adecuadas para una correcta implementación en este TFM.

2.1. El Sistema de Posicionamiento Global o GPS

El Sistema de Posicionamiento Global o GPS (Global Position System) es un sistema GNSS (Global Navigation Satellite System), que permite determinar en toda la Tierra la posición de cualquier objeto (una persona, un vehículo) con una precisión de hasta centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión (de 5 a 10 metros). El sistema fue desarrollado, instalado y empleado por el Departamento de Defensa de los Estados Unidos de América. Para determinar las posiciones en el globo, el sistema GPS utiliza la trilateración (la trilateración mide la distancia a los satélites para determinar con precisión la posición exacta en la Tierra).

Funcionamiento:

1. Un receptor GPS recibe señales de onda de radio provenientes de 4 o más satélites (Figura 2).
2. Cada señal contiene información acerca del momento en que fue transmitida desde el satélite y la posición en que se encontraba dicho satélite al momento de ser enviada la señal, como toda onda electromagnética, dichas señales viajan a la velocidad de la luz (casi 300.000 kilómetros por segundo).
3. El receptor calcula su posición comparando las distancias en línea recta hasta cada uno de los satélites (restando al momento de recepción de la señal el momento de envío de la señal desde el satélite, esto le permite saber al dispositivo el tiempo que tardó en llegar la señal desde el satélite. Si se le multiplica a la velocidad de la luz, que es siempre la misma, la cantidad de tiempo que tardó en llegar la señal, el resultado es la distancia en línea recta que hay desde el receptor hasta el satélite).
4. Como cada señal incluye también información acerca de la posición de cada satélite al momento de enviarse las señales, se compara cada posición de los satélites y las distancias a los mismos; formando lo que serían unas esferas imaginarias sobre cuya superficie se encuentra el receptor, siendo la intersección de ellos exactamente la localización

del dispositivo GPS. El receptor realiza las operaciones matemáticas (trilateración) para determinar la posición relativa de cualquier objeto.



Figura 2. Determinación de la posición del receptor GPS [1]

El GPS permite saber dónde te encuentras en este instante y qué camino debes tomar para llegar a cualquier lugar, así como la distancia existente entre un sitio y otro.

Existen otros sistemas GNSS como GLONASS (Rusia), Galileo (Europa), Beidou (China), que pueden ser utilizados para la geolocalización del mismo modo que el GPS.

Para utilizar la constelación de satélites GPS se debe disponer de un dispositivo GPS (chip GPS). Hoy en día muy económicos y además prácticamente ya incorporado en cualquier dispositivo móvil.

En la Figura 3 se muestra un ejemplo de un módulo de desarrollo GPS.

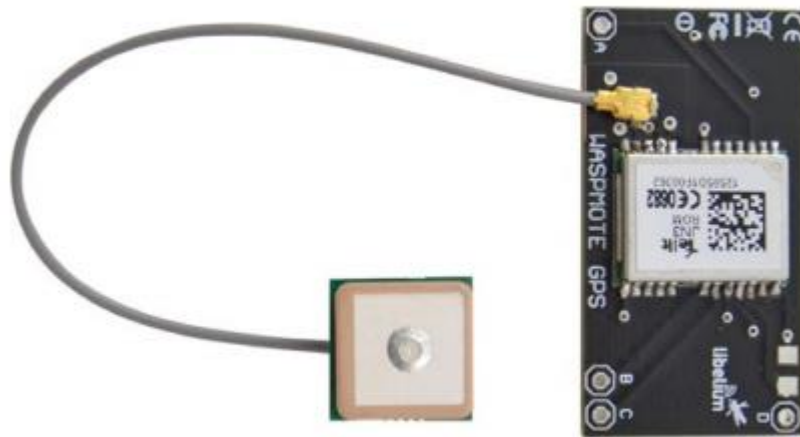


Figura 3. Módulo GPS

2.2. El Internet de las cosas (IoT)

Se puede definir el Internet de las Cosas (IoT, Internet of Things) como la “Infraestructura mundial para la sociedad de la información que propicia la prestación de servicios avanzados mediante la interconexión de objetos (físicos y virtuales) gracias a la interoperatividad de tecnologías de la información y la comunicación presentes y futuras” [2] [3].

Otra definición como la consolidación a través de la red de redes de una "red" que alojase una gran multitud de objetos o dispositivos, es decir, poder tener conectada a esta todas las cosas de este mundo como podrían ser vehículos, electrodomésticos, dispositivos mecánicos, o simplemente objetos tales como calzado, muebles, maletas, dispositivos de medición, biosensores, o cualquier objeto que nos podamos imaginar, y todos estos objetos intercambian información, en definitiva intercambian datos.

El modelo de referencia de IoT consta de cuatro capas y de capacidades de gestión y de seguridad relacionadas con estas cuatro capas. Tal y como se muestra en la Figura 4 las cuatro etapas son las siguientes:

- Capa de aplicación
- Capa de apoyo a servicios y aplicaciones
- Capa de red
- Capa de dispositivo

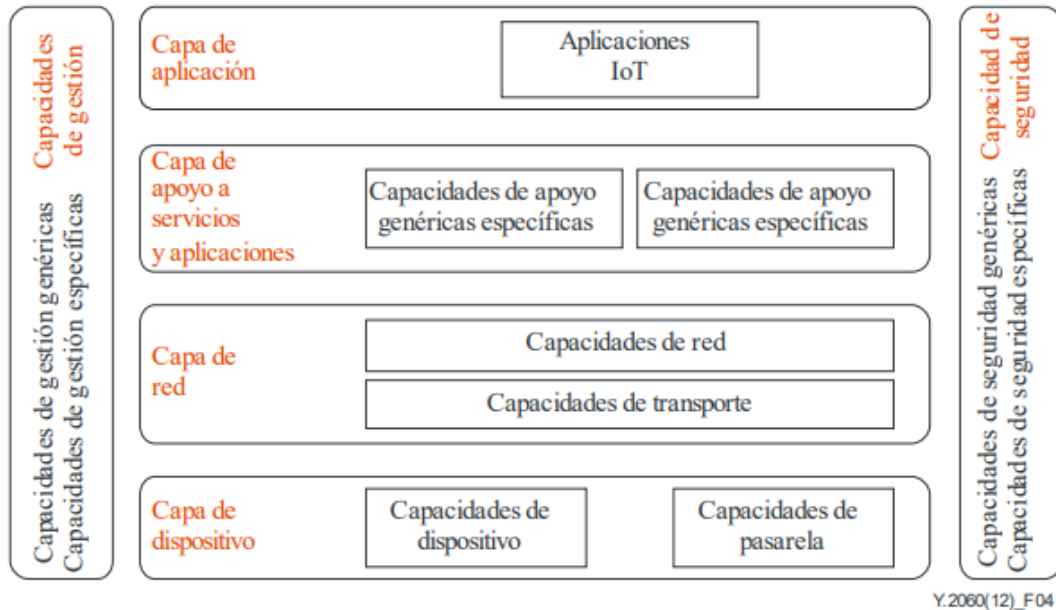


Figura 4. Modelo de referencia de IoT [2]

A continuación, se presenta una breve descripción de cada una de estas capas [3]:

- **Capa de Aplicación:** En esta capa se encuentran los componentes propios de cada implementación que permiten interactuar con el resto de la aplicación, tales como los componentes de gestión, configuración y seguridad, más los componentes de visualización de datos.
- **Capa de apoyo a servicios y aplicaciones:** En esta capa se tienen en cuenta la infraestructura, la localización de dicha infraestructura dentro de la arquitectura y la relevancia que tiene el middleware.
- **Capa de red:** En esta capa se realiza la agregación de datos y se encuentran las tecnologías de comunicación, los algoritmos de agregación, enrutamiento y disseminación, de control de topologías y los protocolos y formatos para el intercambio de mensajes.
- **Capa de dispositivos:** En esta capa se encuentran los dispositivos físicos que soportan la aplicación, como son los nodos sensores, y se gestiona la identificación de estos, las capacidades de hardware que estos tienen y el recurso de energía.

Toda la información viaja a través de una infraestructura de red de comunicaciones.

Las aplicaciones de IoT tienen requisitos específicos tales como largo alcance, baja velocidad de datos, bajo consumo de energía y rentabilidad. Las tecnologías de radio de corto alcance ampliamente utilizadas (por ejemplo, ZigBee, Bluetooth) no están adaptadas para escenarios que requieren transmisión de largo alcance. Las soluciones basadas en comunicaciones celulares (por ejemplo, 2G, 3G y 4G) pueden proporcionar una mayor cobertura, pero consumen una energía excesiva del dispositivo.

2.3. LPWAN

Los requisitos de las aplicaciones de IoT han impulsado el surgimiento de una nueva tecnología de comunicación inalámbrica: las redes de área amplia de baja potencia (LPWAN).

LPWAN (Low Power Wide Area Network) o red de área amplia de baja potencia, conocido también como LPWA o LPN, es un tipo de red de área amplia de telecomunicaciones inalámbricas diseñada para permitir comunicaciones de largo alcance a una tasa de bits baja entre cosas (objetos conectados), como sensores operados con una batería [4]. La baja potencia, la baja velocidad de bits y el uso previsto distinguen este tipo de red de una WAN inalámbrica que está diseñada para conectar a usuarios o empresas, y transportar más datos, usando más energía.

Actualmente se pueden encontrar en el mercado IoT diferentes tecnologías radio para ajustarse a las distintas necesidades de los clientes, y sin duda, uno de los sectores de mayor importancia es el de LPWAN. Las características buscadas en ellas son:

- Larga duración de batería
- Bajo coste, tanto para los módulos de como para los planes de suscripción de acceso a la red
- Cobertura amplia
- Soporte para un gran número de dispositivos

Para tener una idea de la relevancia de LPWAN podemos considerar la predicción de la cantidad de dispositivos LPWAN conectados alrededor del mundo para 2021 sea en torno a los 3.500 millones de dispositivos.

LPWAN está ganando cada vez más popularidad en las comunidades industriales y de investigación debido a su baja potencia, largo alcance, y las características de comunicación de bajo costo. Proporciona una comunicación de largo alcance de 10-40 km en zonas rurales y de 1 a 5 km en zonas urbanas [4].

LPWAN es muy adecuado cuando solo se necesita transmitir pequeñas cantidades de datos en un rango largo, como se muestra en la Figura 5.

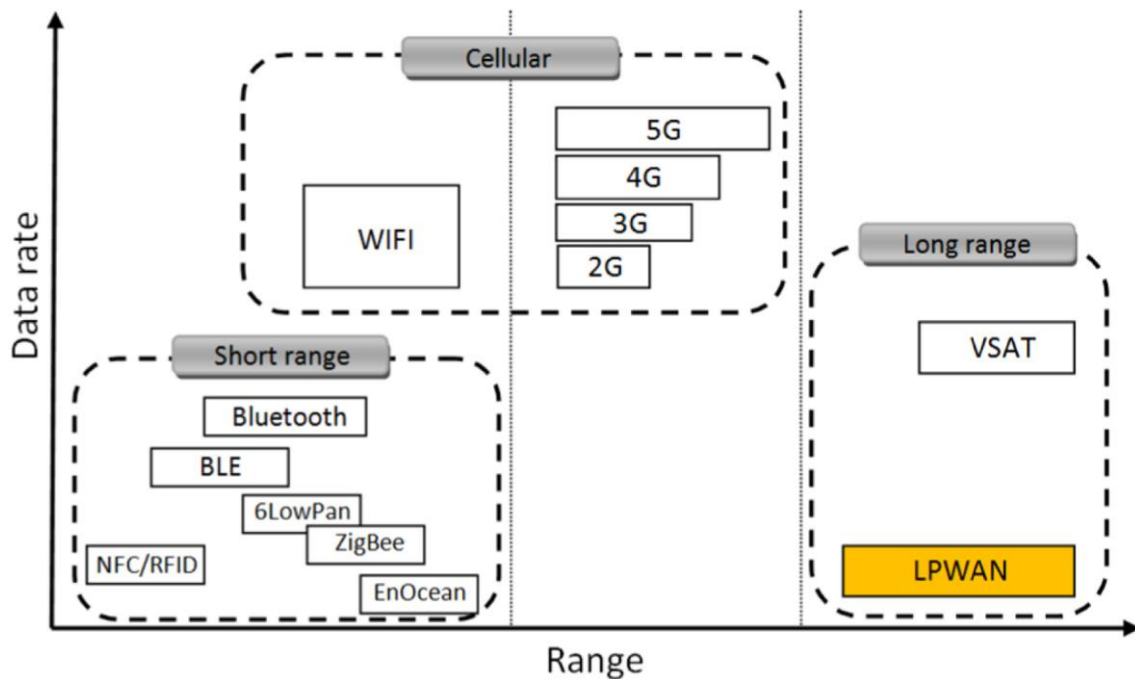


Figura 5. Velocidad de datos vs. Área de cobertura de tecnologías inalámbricas [4]

Existen varias implementaciones de tecnologías LPWAN, las principales son Sigfox, LoRa/LoRaWAN, LTE-M y NB-IoT.

Dentro de las tecnologías que trabajan en las bandas ISM no licenciadas se encuentran Sigfox y LoRa/LoRaWAN. Se trata de soluciones propietarias, no sujetas a ningún órgano general de estandarización. Las bandas en las que trabajan limitan el tiempo de emisión para que las interferencias entre usuarios no sean demasiado grandes. Por otro lado, se destacan las redes derivadas de LTE, NB-IoT y LTE-M que utilizan las bandas del espectro licenciadas.

2.3.1. Sigfox

Sigfox es una tecnología propietaria de la empresa francesa que lleva su propio nombre y está basada en una modulación diferencial DBPSK para subida hacia la plataforma y una modulación GFSK para la descarga de datos. En ambas direcciones trabaja sobre la tecnología de comunicación "Ultra Narrow Band" (UNB) transmitiendo a 100 Hz sobre las bandas de frecuencias de sub-GHz libres.

Sigfox utiliza bandas ISM sin licencia, 868 MHz en Europa y 902 MHz en el resto del mundo (928 MHz en algunas regiones). Por ello se presentan algunas limitaciones en el uso de la red, debiéndose destacar un máximo ciclo de trabajo del 1% que viene impuesto por el Instituto Europeo de Estándares y Comunicaciones (European Telecommunications Standards Institute, ETSI) [5]¹.

Al emplear la banda ultra estrecha, Sigfox utiliza el ancho de banda de frecuencia de manera eficiente y experimenta niveles de ruido muy bajos, lo que lleva a un

¹ Ver Anexo 8.1. Para Sigfox esta limitación no es relevante ya que el máximo de mensajes que permite es 140 al día, cumpliendo con lo exigido por la regulación.

consumo de energía muy bajo, una alta sensibilidad del receptor y un diseño de antena de bajo costo a expensas del rendimiento máximo. Sigfox utiliza 192 KHz de la banda disponible para el intercambio de mensajes. Cada mensaje tiene un ancho de 100 Hz y se transfiere con una velocidad de datos de 100 o 600 bits por segundo (Figura 6), según la región.

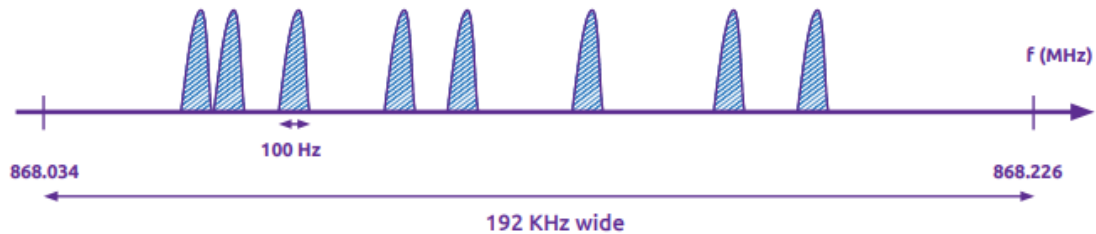


Figura 6. Sigfox ancho de banda [6]

Los paquetes que transmite son de un tamaño reducido, componiéndose por una parte fija de 12 bytes que incluye un preámbulo, un identificador del dispositivo y otros metadatos, y por la parte variable formada por la información, siendo ésta de hasta 12 bytes. En total, cada paquete puede variar desde los 12 hasta los 24 bytes más unos bits extras usados para la autenticación de los parámetros.

A lo largo de un día, Sigfox permite el envío de hasta 140 mensajes (en la Figura 7 se muestran distintos tipos de suscripciones), tardando cada uno de ellos aproximadamente 2 segundos en llegar a la estación base.



Figura 7. Tipos de suscripciones de Sigfox (mensajes de enlace ascendente / descendente) [6]

Una característica para tener en cuenta es que Sigfox es eficaz para las comunicaciones desde los puntos finales a las estaciones base, pero no es particularmente eficaz en transmisiones desde las estaciones base hasta los puntos finales. Las descargas son más lentas que la subida de información.

El alcance de la tecnología varía según el entorno de aplicación, desde los 10 km en entornos urbanos hasta los 10 a 40 km en entornos rurales.

Sigfox es una solución que busca la compatibilidad con muchos fabricantes de módulos que adoptan una posición neutra como Avnet, Telit, Texas Instrument, etc. Esto significa que hay más variedad de dispositivos compatibles con esa tecnología en el mercado.

Actúa directamente en el mercado como un operador y puede hablar directamente con los clientes finales, lo que puede suponer un nuevo entrante en el mercado para los integradores y los especialistas en soluciones y servicios que adoptan esa tecnología y por lo tanto para utilizar Sigfox es necesario utilizar su solución. Algunos operadores han ido incluso más lejos y han decidido entrar directamente en el capital de la empresa como por ejemplo la española Telefónica, la coreana SK Telecom y la japonesa NTT DOCOMO.

2.3.1.1. Arquitectura de red

Sigfox utiliza una topología en estrella sencilla, los objetos transmiten sus mensajes a la red a través de las estaciones habilitadas más cercanas. Cada estación está directamente conectada a la nube de Sigfox que demodulan y reenvían los mensajes a la nube mediante un enlace punto a punto. Por último, estos datos son reportados a los servidores de los usuarios o plataformas IT (Figura 8).

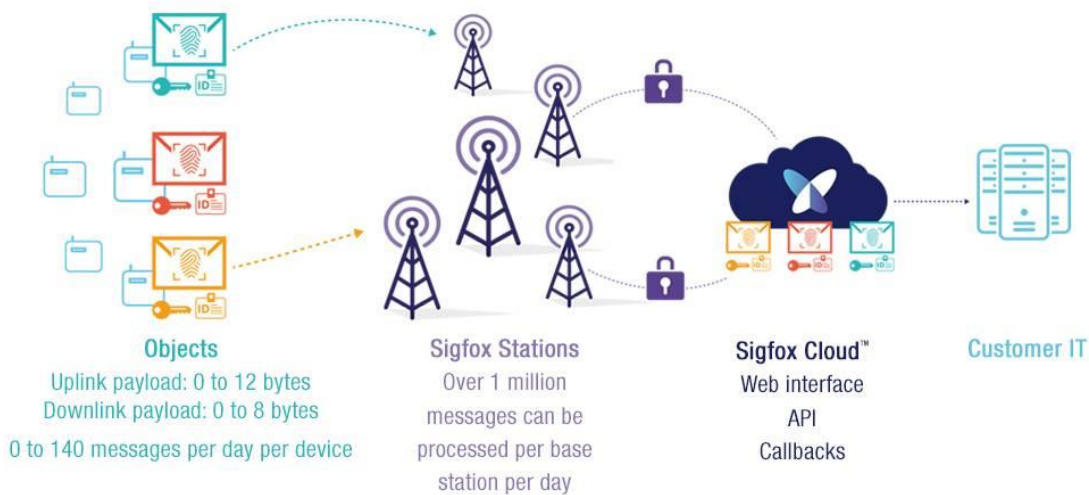


Figura 8. Arquitectura Sigfox [6]

El funcionamiento de esta red es muy similar a las redes de telefonía celular debido a que esta red funciona a partir de la colocación de varias estaciones receptoras y transmisoras; la diferencia entre las estaciones Sigfox y las de telefonía celular es en que los dispositivos y sensores que estén conectados a la red Sigfox no están sujetas a una sola estación base específica, esto es, cualquier estación puede recibir la información y transmitirla hacia la nube.

La red Sigfox conecta los dispositivos de campo a los clústeres de servidores de backend en la nube mediante una red basada en IP. Una vez que el mensaje se envía desde el dispositivo a la estación base y se recibe en el backend de Sigfox, los usuarios pueden administrar los datos recibidos con diferentes Plataformas IoT mediante llamadas HTTP a la API de Sigfox. Es decir, Sigfox dispone de su propio Cloud para el almacenamiento y acceso a la información.

2.3.1.2. Disponibilidad geográfica

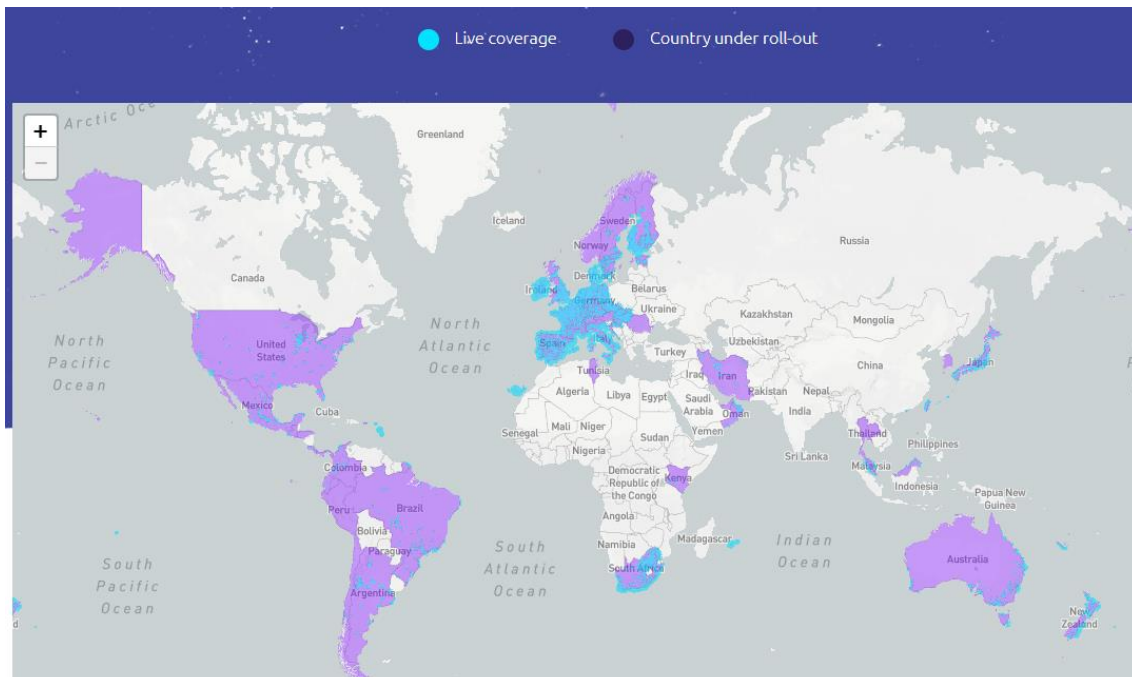


Figura 9. Cobertura mundial Sigfox [6] [7]

En 2018, la compañía ha alcanzado una serie de hitos tales como expandir su red hasta alcanzar 60 países (Figura 9) a través de nuevos operadores, que se han unido al ecosistema IoT más grande del mundo. Sigfox y sus operadores han invertido más de 500 millones de euros hasta la fecha para crear esta red única. El último hito se alcanzó en diciembre, cuando Polonia se unió a la familia de operadores de la red, permitiendo a la firma convertirse en el único operador que ofrece cobertura IoT completa a nivel europeo. La red ya ofrece cobertura a más de 1.000 millones de personas, un gran avance desde el inicio de la compañía en 2011 [8].

En España Sigfox alcanza el 94,5% de la población y el 70,8% del territorio [9]. Un caso de éxito muy conocido es el de Securitas Direct [10].

2.3.1.3. Ejemplo de módulo Sigfox

Un ejemplo de módulo de desarrollo Sigfox es el Sigfox 868.

Las características de un módulo Sigfox 868 para Europa son [11]:

- Frecuencia: ISM 868 MHz
- Potencia de transmisión: 16 dBm.
- Consumo del Chipset:
 - Tx: 51 mA @ +16 dBm.
 - Rx: 16 mA.
- Sensibilidad: -126 dBm.
- Certificado Sigfox: Class 0u (la más alta).

En la Figura 10 se presenta un módulo de Sigfox para trabajar en Europa.

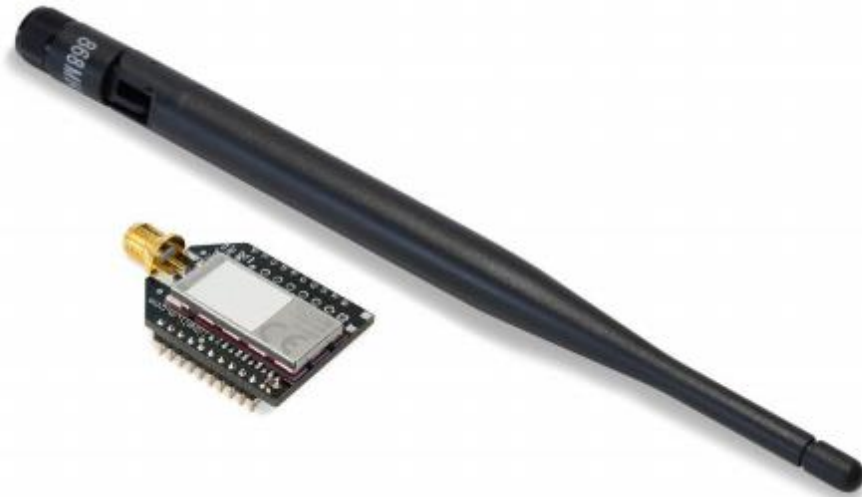


Figura 10. Módulo Sigfox 868 para Europa con antena de 4.5 dBi [11]

2.3.1.4. Resumen de características más relevantes

- Banda ISM a 868 MHz en Europa.
- Tamaño máximo del mensaje de 12 bytes.
- Envío máximo de 140 mensajes al día.
- Alcance máximo de hasta 40 Km.

(En el apartado 2.3.4 Comparativa, se muestran con más detalle todas características).

2.3.2. LoRa/LoRaWAN

LoRa (Long Range), es la capa física o modulación inalámbrica empleada para crear enlaces de largo alcance. La modulación LoRa, patentada por la empresa Semtech, se basa en la técnica de espectro ensanchado (Frequency Spreading Keying - FSK) y una variación del espectro extendido chirp (chirp spread spectrum-CSS), en la cual se mantiene las características de baja potencia de la modulación FSK, pero consiguiendo un rango de comunicación más significativo; también emplea corrección integrada de errores (Forward Error Correction-FEC). LoRa utiliza bandas ISM sin licencia, opera en Europa en el espectro de frecuencias radioeléctricas entre 863 a 870 MHz, entre 902 a 928 MHz en América y centrada en los 433 MHz en Asia [12] [13]. Por ello se presentan algunas limitaciones en el uso de la red, debiéndose destacar un máximo ciclo de trabajo del 1% que viene impuesto por el Instituto Europeo de Estándares y Comunicaciones (European Telecommunications Standards Institute, ETSI) [5]².

El protocolo LoRa utiliza una señal cuya frecuencia aumenta o desciende con el tiempo para codificar la información y extender el ancho de banda final. Existen seis factores de ensanchamiento (SF7 - SF12) y tres anchos de banda (125 kHz,

² Ver Anexo 8.2. En el caso de LoRa/LoRaWAN si es muy importante ya que limita el tamaño de payload, velocidad y número de mensajes que puede enviar.

250 kHz, 500 kHz) [14] en función de los cuales se obtiene una tasa de bits aproximada según la Ecuación 1 del Anexo 8.3:

Cuando se habla de LoRa, esta no se refiere a un caso de implementación de una LPWAN, ya que por sí sola representa solamente la capa física de la red. Para poder crear una red se necesita una capa de control de acceso al medio (MAC), esta capa asociada a los chips LoRa es llamada LoRaWAN.

Esta es la función de LoRaWAN, que es la red en la que LoRa opera y además se encarga de definir el protocolo de comunicación y la arquitectura del sistema para la red, mientras que LoRa, capa física, habilita el enlace de comunicación de largo alcance (Figura 11).

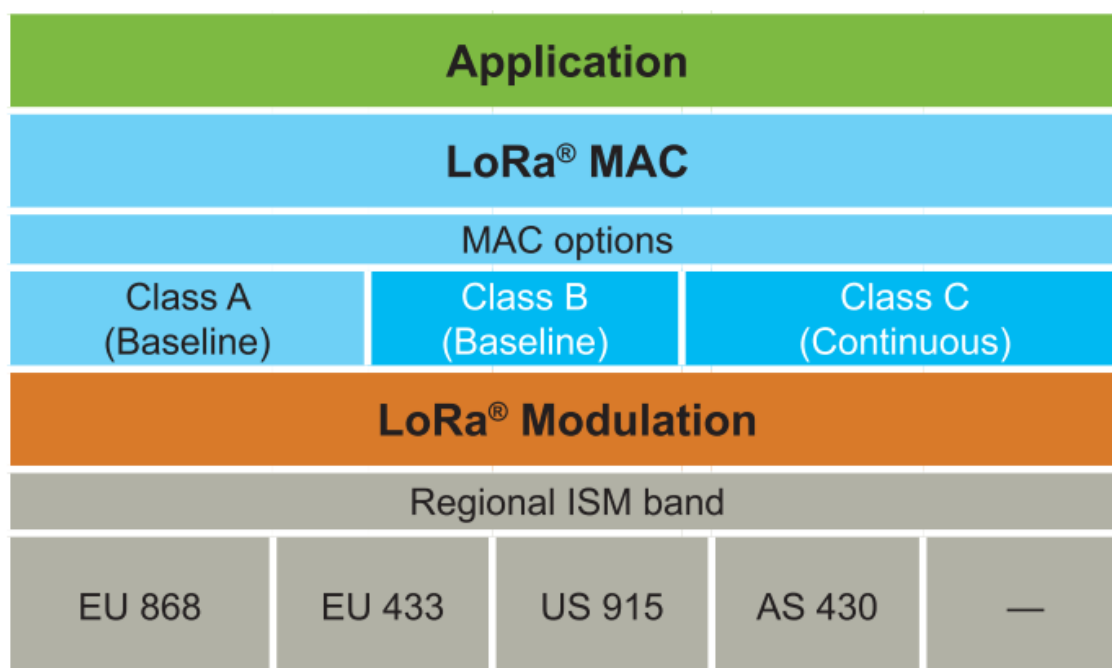


Figura 11. Protocolo de comunicación y arquitectura del sistema de LoRaWAN [12]

LoRaWAN garantiza velocidades de datos de 0,3 kbps hasta 50 kbps, que se consideran aceptables para el envío de datos de sensores en tiempo real en IoT, Machine-to-Machine (M2M), Smart cities y aplicaciones industriales.

Sin embargo, la transmisión de datos de imagen en tiempo real, o cualquier dato que requiera un gran ancho de banda, puede no ser adecuado en redes LoRa. Ésta baja velocidad de datos asegura el bajo consumo de energía de los dispositivos finales, permitiendo el uso de baterías con una vida útil de batería de varios años, ya que LoRa/LoRaWAN está diseñada para sensores y aplicaciones que necesitan enviar pequeñas cantidades de datos a largas distancias, varias veces por hora y desde distintos entornos.

El alcance de la tecnología varía según el entorno y puede llegar desde los 5 km en zonas urbanas hasta 20 km en zonas rurales.

2.3.2.1. Clases de dispositivos LoRaWAN

Existen tres clases de dispositivos LoRaWAN, denominadas Clase A, Clase B y Clase C. Todos los dispositivos LoRaWAN implementan al menos la funcionalidad de clase A. Además, pueden implementar las opciones de Clase B o Clase C [12].

- **Clase A:** Los dispositivos de Clase A permiten una comunicación bidireccional, con la limitación de que sólo pueden recibir datos (canal downlink) si han enviado antes un paquete (canal uplink). Eso es porque cada vez que el dispositivo envía un paquete, se abren dos ventanas de recepción, la primera un segundo después de la transmisión y la segunda un segundo después de la primera ventana, con la oportunidad de recibir un paquete de vuelta. Este paquete de vuelta contiene el ACK del paquete enviado, así como datos de la aplicación si es necesario. Este tipo de dispositivos serán los de menor consumo energético de la especificación, puesto que estarán en modo dormido por defecto e iniciarán toda comunicación. Servirán para aplicaciones en las que los dispositivos no deban recibir datos habitualmente. Un dispositivo compatible con LoRaWAN debe implementar siempre esta clase básica.
- **Clase B:** Los dispositivos de clase B añaden la capacidad de recibir datos (downlink) sin necesidad de enviar un paquete (uplink), de esta forma la aplicación puede enviar datos a los dispositivos de forma programada. Esto se consigue mediante el envío periódico de beacons por parte del gateway. Estos beacons permiten a los dispositivos estar sincronizados con el gateway, y de esta forma pueden negociar tiempos de recepción de paquetes desde el gateway al dispositivo (downlink). Así, las ventanas de recepción se abren en momentos determinados. Esta clase de dispositivos tendrán un consumo mayor de energía que los de clase A debido a la recepción periódica de los beacons desde el gateway.
- **Clase C:** Los dispositivos de clase C están permanentemente escuchando (es decir, en modo de recepción), y por tanto pueden recibir datos (downlink) en cualquier momento (excepto cuando estén enviando datos (uplink)). En otras palabras, las ventanas de recepción se mantienen siempre abiertas excepto cuando transmiten. Esta clase proporciona los mejores tiempos de respuesta y capacidad de envío desde el servidor a los dispositivos, a costa de un consumo energético mucho mayor respecto a las clases A y B.

Si se desea utilizar LoRa, hay 2 opciones:

- **Desplegar una red propia:** Cualquiera puede comprar chips y crear su propia red de dispositivos, gateways y backends. Además, tendrá que implementar y mantener tanto las pasarelas como los dispositivos, pero tendrá un control total sobre su red y podrá adaptarlo y modificarlo a medida que crezca su producto y cambie sus necesidades.
- **Utilizar la red de un operador de red:** Varios operadores están empezando a ofrecer redes LoRaWAN en ciertas áreas. En breve podrán

comprarse planes de conectividad de ellos, aunque hay que asegurar cobertura y continuidad del servicio.

2.3.2.2. Arquitectura de red

LoRa utiliza una topología de estrella, en la que cada nodo final se comunica con varias puertas de enlace que se comunican con el servidor de red.

Tal y como se muestra en la Figura 12 LoRa tiene cuatro elementos de red:

- Los nodos finales (end nodes) recopilan datos del sensor, lo transmiten upstream y downstream, y reciben la comunicación desde el servidor de aplicaciones. Los dispositivos de endpoint usan comunicación inalámbrica single-hop con una o varias compuertas.
- El Concentrador/Compuerta (gateway) actúa como un puente transparente y retransmite los datos bidireccionales entre los nodos finales y los servidores upstream.
- El servidor de red se conecta (network server) a varias puertas de enlace a través de una conexión TCP/IP segura, ya sea por cable o inalámbrica; elimina los mensajes duplicados; decide qué compuerta debe responder a un mensaje de nodo final; y gestiona el nodo final, las velocidades de transmisión de datos con una velocidad de datos adaptable (ADR), destinadas a maximizar la capacidad de la red y extender la vida útil de la batería del nodo final.
- El servidor de la aplicación (application server) recopila y analiza los datos de los nodos finales y determina las acciones del nodo final.

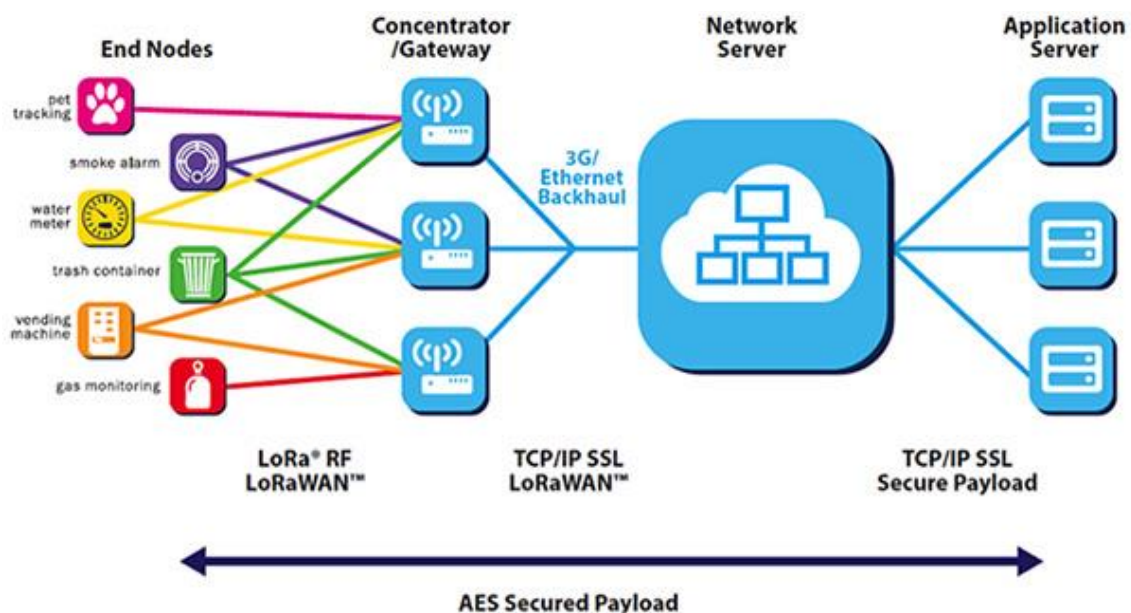


Figura 12. Arquitectura LoRa [12]

2.3.2.3. Disponibilidad geográfica

En la Figura 13 se muestra la disponibilidad geográfica de LoRa/LoRaWAN. En España existen por ejemplo empresas como Redexia o Libelium que ofrecen dispositivos y uso de esta tecnología [15] [16].

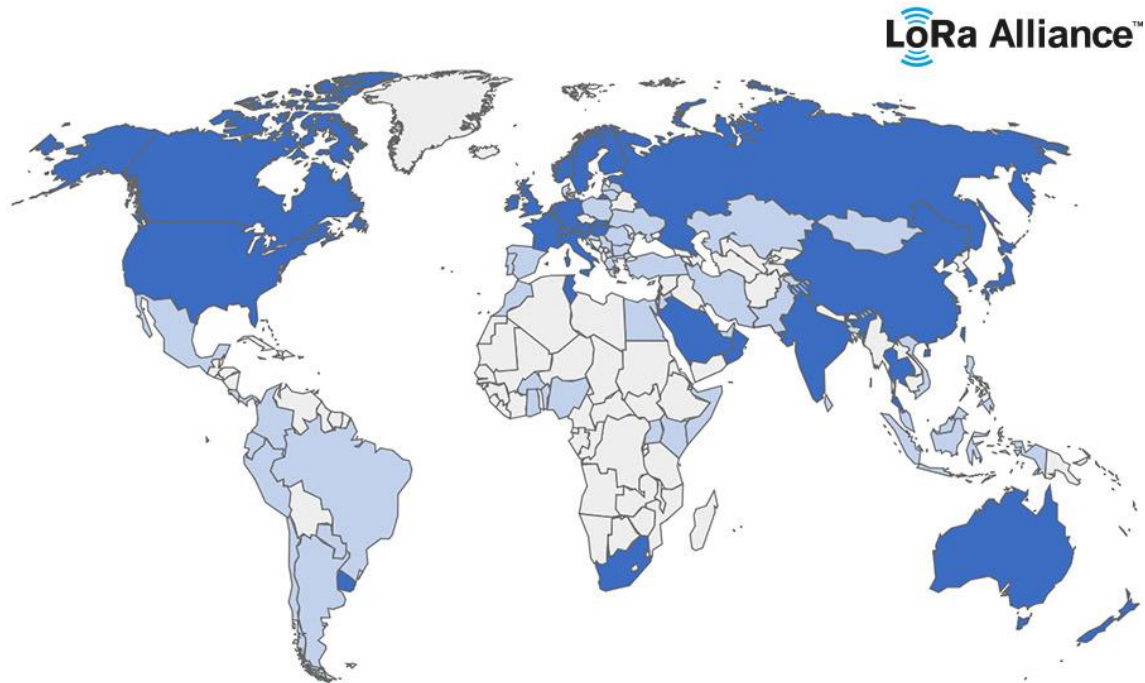


Figura 13. Cobertura mundial LoRa [15]

En el directorio de miembros se puede obtener el listado completo de proveedores de LoRa Alliance en todo el mundo [17].

2.3.2.4. Ejemplo de módulos de desarrollo LoRa/LoRaWAN

Un ejemplo de módulo transceptor LoRa es el SX1272 de Libelium (desarrollado por Semtech) [16].

- Módulo: SX1272
- Frecuencia: 863-870 MHz
- Potencia: (Europa) 14 dBm
- Sensibilidad: -134 dBm
- Canales: 8, 13
- Distancia máxima: 22 km

En la Figura 14 se presenta un módulo de LoRa transceptor y un módulo concentrador para trabajar en Europa.



Figura 14. Módulo transceptor LoRa SX1272 y Concentrador LoRa [16]

Un ejemplo de módulo LoRaWAN es el RN2433 de Libelium.

Las características de un módulo LoRaWAN para Europa son [16]:

- Protocolo: LoRaWAN 1.0, Class A
- Frecuencia: EU 863-870 MHz y EU 433 MHz ISM.
- Potencia de transmisión: > +14 dBm
- Sensibilidad: < -136 dBm
- Radio de cobertura: >15 km en rural y >5 en zonas urbanas.
- Consumo del Chipset: 38.9 mA
- Bit rate: desde 250 hasta 5470 bps

En la Figura 15 se presenta un módulo LoRaWAN para trabajar en Europa.



Figura 15. Módulo RN2433 LoRaWAN para Europa clase A con antena [15]

2.3.2.5. Resumen de características más relevantes

- Banda ISM a 863 MHz en Europa.
- Tamaño máximo del mensaje de 243 bytes.

- Envío máximo de mensajes dependiente del 1% de la regulación europea.
- Alcance máximo de hasta 15 Km.

(En el apartado 2.3.4 Comparativa, se muestran con más detalle todas características).

2.3.2.6. The Things Network

LoRa/LoRaWAN carece de su propio Cloud por lo que es necesario utilizar un tercero. A continuación, se presenta The Things Network³ que es una alternativa gratuita.

The Things Network TTN (The Things Network) es una red IoT basada en LoRaWAN global, abierta y financiada mediante crowdfunding, la cual ofrece cobertura proporcionada por los gateways de los usuarios de la red. Un usuario se conecta a la red y puede registrar un Gateway; desde gateways comerciales hasta gateways hechos mediante una raspberry y un módulo de comunicaciones LoRaWAN, por ejemplo. Actualmente, TTN está desarrollando su propio gateway LoRaWAN, el cual se encuentra en fase de pruebas [18].

TTN proporciona un backend de servidores de aplicación que gestionan los gateways, y una plataforma que permite a los usuarios crear sus propias aplicaciones, asociar sus dispositivos y recibir los datos. Además, ofrece muchas formas de recibir la información de la red, mediante servicios HTTP, servicios REST, o incluso un bróker MQTT al que se puede subscribir un cliente para recibir los datos. TTN proporciona una forma sencilla de desplegar un servicio IoT sin requerir una gran infraestructura, y de forma gratuita. Solo se requiere tener cobertura de un gateway perteneciente a la red, propio o de un tercero y algún dispositivo que envíe información a la aplicación.

2.3.3.LTE: NB-IoT y LTE-M

LTE es un estándar de la 3GPP para transmisión de datos de alta velocidad, que surge tras la tercera generación de comunicaciones móviles. La revisión de este estándar, LTE-A. sí que se considera que constituye la cuarta generación propiamente dicha. Se trata de una tecnología orientada hacia los datos, con una arquitectura de conmutación de paquetes, en la que se buscó alcanzar cotas elevadas de eficiencia espectral y flexibilidad en la asignación de frecuencias. En LTE la voz también utiliza conmutación de paquetes (VOIP) [19].

La comunicación se lleva a cabo utilizando OFDM en bajada y SC-FDMA en subida. Esta última no es más que una pequeña variante de la primera, pues la diferencia entre ambas está en la aplicación de una operación lineal, una DFT preliminar para reducir la relación PAPR. Con OFDM se divide el ancho de banda en subportadoras pequeñas, en las que la respuesta del canal es aproximadamente plana. Para LTE, la separación entre subportadoras puede ser

³ Existen otras alternativas como por ejemplo Lorient (de pago).

de 15 kHz y 7.5 kHz en su modalidad reducida. El ancho de banda empleado puede adaptarse a las necesidades de la comunicación, y varía de 1.4 a 20 MHz.

NB-IOT y LTE-M son dos tecnologías Low Power Wide Area Network (LPWAN) desarrolladas para aplicaciones IoT derivadas de LTE. Ambas son protocolos para comunicaciones celulares con un ancho de banda bajo que conectan a internet dispositivos que necesitan transmitir pequeñas cantidades de datos, a bajo coste (tanto en lo relativo al hardware como a la suscripción) y con una alta duración de la batería.

Se espera que conecten cientos de millones de cosas a Internet en los próximos años y existen unos claros beneficios que hacen esenciales estas dos tecnologías para el futuro de las comunicaciones IoT:

- Bajo consumo de energía (hasta 10 años de duración de la batería)
- Seguridad en la nube
- Fácil despliegue de la red
- Bajo coste de hardware

Las dos tecnologías (NB-IOT y LTE-M) miran hacia los mismos horizontes, pero cada caso de uso específico hará decantar la balanza hacia una u otra.

Las dos tecnologías se han creado por los estándares 3GPP y tienen características y aplicaciones similares, existen algunas diferencias a considerar [20] [21].

El alcance máximo va desde 2 km en zonas urbanas hasta los 10 km en zonas rurales.

2.3.3.1. NB-IOT (Narrowband-IoT)

NB-IoT está desarrollada para permitir una comunicación eficiente y una larga duración de la batería para dispositivos distribuidos de manera masiva, y utiliza la red de telefonía móvil existente para conectar las “cosas”. Está especialmente centrada en el territorio europeo y, después de su primer lanzamiento comercial, se está extendiendo a nivel mundial desde 2018 [21].

NB-IoT utiliza OFDM en bajada y SC-FDMA en subida.

La separación entre subportadoras es de 15 kHz, y de 3,75 kHz opcionalmente para el sentido ascendente. SC-FDMA se utilizará transmitiendo un solo tono o conjuntos de 3, 6, o 12 subportadoras.

La tipología de comunicación es Half-Duplex.

Las modulaciones empleadas son QPSK y BPSK, esta última solo en subida. En cuanto al ancho de banda utilizado, se utilizan 180 kHz por canal. La especificación fija una potencia de emisión de 23 dBm, con los que se consiguen valores elevados de cobertura, de hasta 164 dB de atenuación compensable.

Las velocidades de transmisión oscilan entre los 144 kbps en subida y los 200 kbps en bajada.

Al ser una tecnología enfocada a un bajo consumo energético, se destaca el uso de dos técnicas para el ahorro. En el primero de los modos de funcionamiento, PSM, el dispositivo avisa a la red que pasa a estado de reposo. Esto hace que deje de escuchar de modo indefinido a la estación base. Cuando el terminal decide terminar con este estado, envía el aviso a la red y vuelve a escuchar. El segundo en cuestión es la eDRX. Su funcionamiento es análogo al primero, pero a diferencia de este no funciona de forma asíncrona. El dispositivo avisa a la red de cuánto tiempo estará en reposo entre avisos de la red (Paging Channel).

Una de las características más atractivas, sobre todo de cara a las operadoras, de NB-IoT es la capacidad de desplegarse de diferentes formas. Se puede hacer de tres modos (Figura 16): In-band, Guard Band y Standalone [22] [23]. En el primero de ellos, se utilizan bloques de recursos de la red LTE ya establecida para dar servicio a NB-IoT, permitiendo así reutilizar la infraestructura con una simple actualización de software.

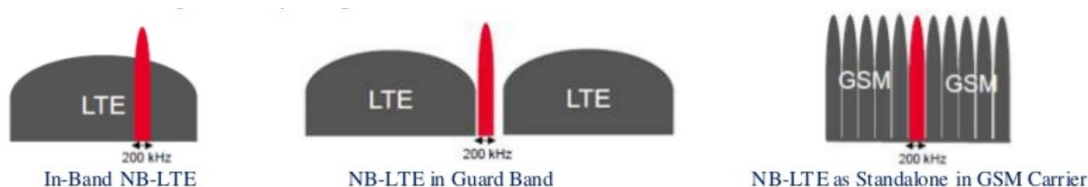


Figura 16. Modos de operación NB-IoT [23]

2.3.3.2. LTE-M (Long Term Evolution For Machines)

LTE-M, también conocida por CAT-M1, utiliza las antenas LTE instaladas y está optimizada para un ancho de banda mayor y para conexiones de móvil que incluyan voz. Por el momento está desplegado en Norteamérica y el lanzamiento global empezó en 2018 [21].

LTE-M utiliza como tecnología de acceso OFDM en bajada y SC-FDMA en subida, con una división entre subportadoras de 15 kHz. La comunicación es Half-Duplex. También se habilitan modos de operación FDD Y TDD. En cuanto a la modulación empleada, en función del estado del enlace se utiliza QPSK o 16-QAM.

El ancho de banda dedicado para cada canal LTE-M coincide con la configuración mínima de LTE para sus canales, siendo esta de 1.40 MHz. La reutilización de la estructura destinada a LTE tiene un coste reducido, pues LTE-M es un servicio compatible con LTE, y, por tanto, solo sería necesaria una actualización de software para poder operar.

Al igual que ocurre con NB-IOT, también se resalta la repetición de paquetes como principal mecanismo para alcanzar buenos valores en cobertura. A esto se añade la posibilidad de aplicar diversidad en frecuencia en los envíos (frequency hopping).

Otras características compartidas con NB-IOT son los modos de ahorro energéticos PSM y eDRX. Este último con una pequeña diferencia, pues los ciclos en los que el dispositivo no escucha para LTE-M llegan hasta en torno los

45 minutos, frente a las 3 horas de NB-IOT. Así pues, con lo anteriormente citado se consiguen hasta 155,7 dB de atenuación máxima compensable para una potencia de emisión de 23 o 20 dBm, fijada en la especificación.

La velocidad es de 1 Mbps tanto en subida como en bajada.

2.3.3.3. Arquitectura de red

La arquitectura de la red utilizada por LTE-M es la misma que LTE y NB-IoT se base en LTE con modificaciones.

La red LTE, también llamada EPS (Evolved Packet System), es una red IP extremo a extremo (E2E) que se divide en dos partes tal y como se muestra en la Figura 17:

- La parte LTE que trata de la tecnología relacionada con una red de acceso de radio E-UTRAN (Evolved Universal Terrestrial Radio Access Network).
- EPC que trata de la tecnología relacionada con la red troncal. Que sea una red IP E2E, extremo a extremo, significa que todo el tráfico fluye mediante IP desde el terminal de usuario (UE), es decir el dispositivo con acceso a IP que utiliza el usuario hasta la red de paquetes o PDN (Packet Data Network) donde está conectada la entidad o entidades que proporcionan el servicio. Una red PDN (Packet Data Networks) es un dominio IP interno o externo al operador, con el cual un UE quiere comunicarse. Estas PDN's proporcionan servicios a los terminales de usuario (UE) tales como, acceso a Internet o Voz/Video mediante el IMS (IP Multimedia Subsystem).

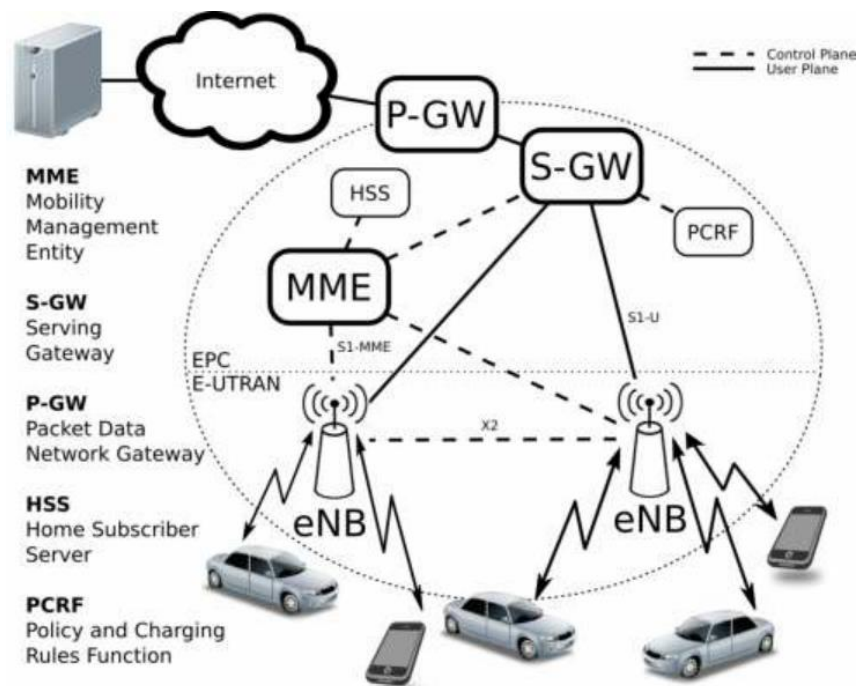


Figura 17. Arquitectura LTE [24]

LTE-M es una tecnología puramente LTE mientras que NB-IoT se basa en un nuevo interfaz radio que puede coexistir con los sistemas LTE, UMTS y GSM actuales y simplifica la arquitectura existente (Figura 18) del núcleo de red (EPC) [25].

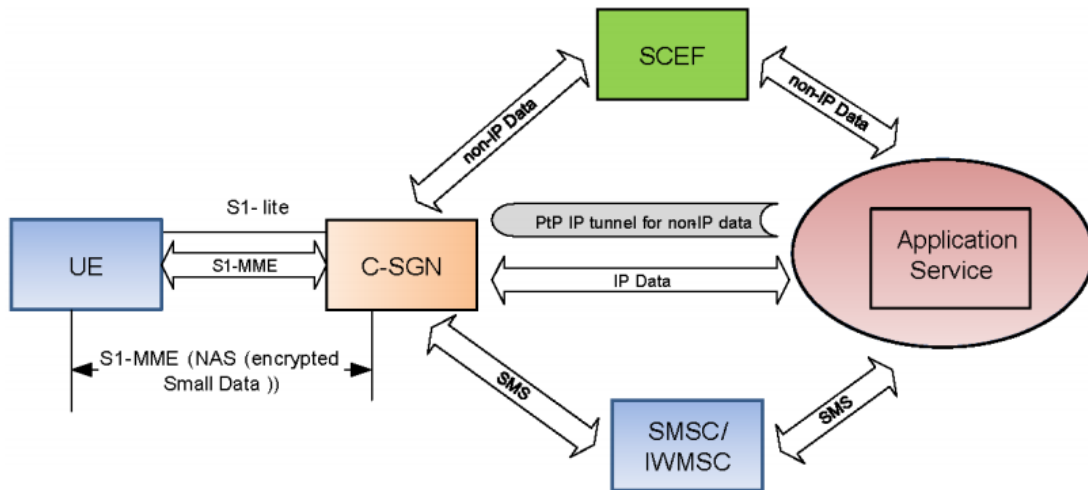


Figura 18. Arquitectura LTE para NB-IoT [25]

2.3.3.4. Disponibilidad geográfica

Para seguir la expansión de estas tecnologías LPWAN podemos acceder a la web de la asociación GSM (GSMA), un organismo que representa los intereses de los operadores de telefonía a nivel mundial. También apoya la estandarización, el despliegue y la promoción de nuevas tecnologías móviles.

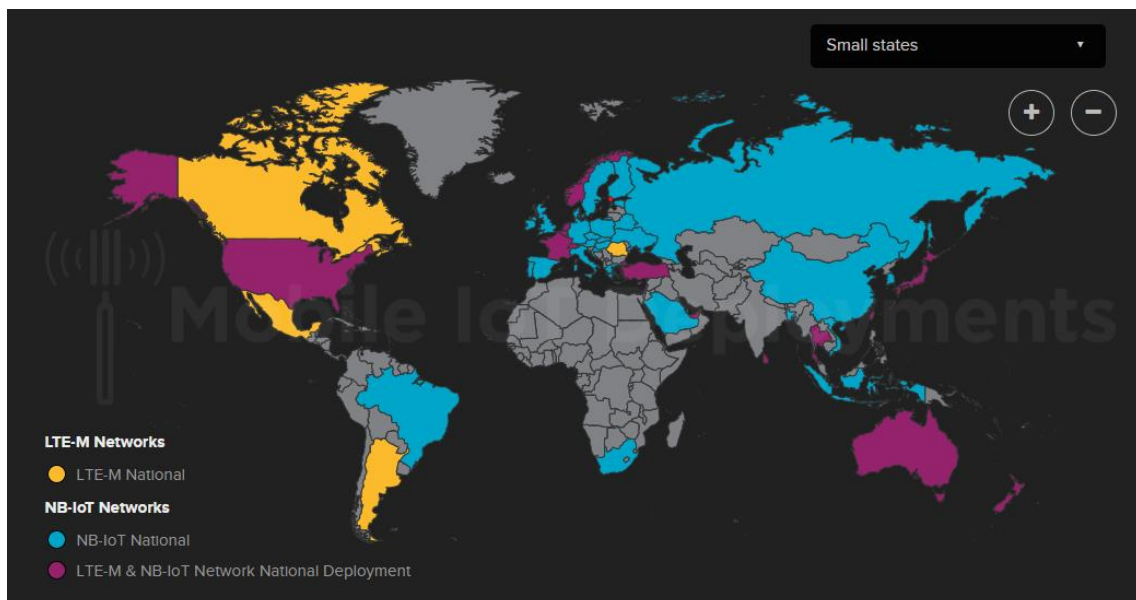


Figura 19. Cobertura mundial NB-IoT/LTE-M [26]

Hasta marzo de 2019, la GSMA ha identificado (Figura 19):

- 149 operadores en 69 países invierten en una o ambas tecnologías de red NB-IoT y LTE-M
- 102 de esos operadores en 52 países habían desplegado / lanzado al menos una de las tecnologías NB-IoT o LTE-M de los cuales 20 operadores en 19 países habían desplegado / lanzado tanto NB-IoT como LTE-M
- 22 países albergan redes NB-IoT y LTE-M desplegadas / lanzadas
- 28 países son el hogar de redes NB-IoT desplegadas / lanzadas solamente
- Dos países albergan únicamente redes LTE-M desplegadas / lanzadas
- 140 operadores en 69 países invierten en redes NB-IoT
- 88 de esos operadores en 50 países habían desplegado / lanzado sus redes
- 60 operadores en 35 países invirtiendo en redes LTE-M
- 34 de esos operadores en 24 países habían desplegado / lanzado sus redes.

En España Vodafone da cobertura NB-IoT en ciudades con más de 25000 habitantes [27]. Movistar también ha comenzado el despliegue de NB-IoT y LTE-M en diversas regiones rurales [28].

2.3.3.5. Ejemplo de módulo de desarrollo NB-IoT/LTE-M

A continuación, se detalla el módulo BG96 (Quectel) NB-IoT/LTE-M [29].

Bandas de frecuencias:

- NB-IoT/LTE-M:
 - LTE FDD: B1/B2/B3/B4/B5/B8/B12/B13/B18/B19/B20/B25**/B26*/B28
 - LTE TDD: B39 (solo para LTE-M)
- Velocidad de transmisión:
 - NB-IoT: Max. 32 kbps (DL), Max. 70 kbps (UL)
 - LTE-M: Max. 375 kbps (DL), Max. 375 kbps (UL)
- GNSS:
 - Supports GPS, GLONASS, BeiDou/Compass, Galileo and QZSS.
- Sensibilidad:
 - -113 dBm @Cat NB1, CE Level 0
 - -107 dBm @Cat M1, 1.4 MHz Bandwidth, CE Mode A
- SIM: Nano-SIM (4FF standard) (no incluida)

El módulo NB-IoT/LTE-M viene con una antena celular para mejorar la recepción de la señal. Además, incluye GPS y también incluye la antena.

Ambas antenas pueden usarse en cualquiera de los 2 conectores del módulo. Las bandas operativas de la antena dipolo van desde 698 a 960 MHz y desde 1710 a 2690 MHz. La ganancia máxima de la antena se observa a 2.6 GHz y es de 3.4 dBi.

Para obtener el máximo rendimiento, se recomienda colocar las antenas de la siguiente forma:

- La antena celular principal debe estar en posición vertical, apuntando al cielo, para irradiar mejor a las estaciones base.
- La antena GPS debe estar en posición horizontal, ya que la señal del satélite GPS provendrá de arriba.

En la Figura 20 se presenta un módulo NB-IoT/LTE-M con las antenas.



Figura 20. Módulo NB-IoT/LTE-M con sus antenas y apertura para nano-SIM de 3.4 dBi [29]

2.3.3.6. Resumen de características más relevantes

- Bandas LTE.
- Tamaño máximo del mensaje de 1600 bytes.
- Envío máximo de mensajes ilimitado.
- Alcance máximo de hasta 10 Km.

(En el apartado 2.3.4 Comparativa, se muestran con más detalle todas características).

2.3.4. Comparativa

En este apartado se muestran varias tablas comparativas de las tecnologías LPWAN descritas anteriormente y se comentan las más relevantes en cuanto a términos de velocidad, límite de tamaño del mensaje, alcance y consumo energético.

2.3.4.1. Comparativa técnica

En la Tabla 2 se resumen las características técnicas más relevantes de cada tecnología LPWAN descrita anteriormente para su fácil comparación.

	Sigfox	LoRa/LoRaWAN	LTE-M	NB-IoT
Frecuencia de operación	Banda ISM no licenciada Europa: 868 MHz Resto del mundo: 902MHz	Banda ISM no licenciada Europa: 868 MHz América: 915MHz Asia: 433 MHz	Banda Licenciada Frecuencias LTE	Banda Licenciada Frecuencias LTE/2G (GSM standalone, LTE in-band, LTE guard-band)
Modulación	UNB/GSK/BPSK	CSS: Chirp Spread Spectrum	OFDMA	OFDMA
Ancho de banda	100 Hz	500 KHz – 125 KHz	1.4 MHz	200 KHz
Bidireccional	Muy limitado / Half-dúplex	Si / Half-dúplex	Si / Half-dúplex	Si / Half-dúplex
Velocidad	Bajada: <100 bps Subida: <100 bps	Bajada: <50 Kbps Subida: <50 Kbps	Bajada: <1Mbps Subida: <1Mbps	Bajada: <144 Kbps Subida: <200 Kbps
Límite de mensajes	1% según regulación ISM Subida: 140 mensajes/día Bajada: 4 mensajes/día	1% según regulación ISM Puede llegar a ser ilimitados dependiendo del tamaño del payload	Ilimitados	Ilimitados
Tamaño máximo útil del mensaje (payload)	Subida: 12 bytes Bajada: 8 bytes	243 bytes	1600 bytes	1600 bytes
Alcance (MCL ⁴)	Urbano: <10 Km Rural: <40 Km (160 dB)	Urbano: <5 Km Rural: <15 km (157 dB)	Urbano: <2 Km Rural: <10 Km (157 dB)	Urbano: <2 Km Rural: <10 Km (164 dB)
Estandarización	Sigfox + ETSI	LoRa Alliance	3GPP	3GPP
Potencia máxima transmitida	Europa: 14 dBm América: 23 dBm	Europa: 14 dBm América: 23 dBm	20-23 dBm	20-23 dBm

Tabla 2. Comparativa LPWAN - Características técnicas Sigfox vs LoRa/LoRaWAN vs LTE-M vs NB-IoT

⁴ Ver Anexo 8.4

2.3.4.2. Velocidad de transmisión de datos

Sigfox se sitúa como la opción de menor velocidad frente LTE-M que destaca por ser la opción de mayor velocidad que además es la única que permite la movilidad del objeto.

NB-IoT, Sigfox y LoRa no son opciones adecuadas para las aplicaciones de uso compartido de automóviles. A diferencia de otras tecnologías de LPWAN, LTE-M es el único que admite movilidad (transferencia, etc.), lo que permite el seguimiento de automóviles en movimiento. Cuando maneja una flota, necesita conectividad continua mientras el vehículo está en movimiento. Con LTE-M, las transferencias son ininterrumpidas.

2.3.4.3. Limite en el número de mensajes

Este es otro aspecto por el que destaca Sigfox negativamente ya que está limitado al envío de 140 mensajes como máximo al día. En cuanto al resto de tecnologías a priori no tienen limitación en cuanto al límite de mensajes que puede enviar al cabo del día.

2.3.4.4. Alcance

En cuanto a alcance Sigfox destaca muy por encima de las demás tecnologías con un alcance de hasta 40 Km.

2.3.4.5. Consumo de batería

Tanto Sigfox como LoRaWAN tiene un menor consumo energético. Por el contrario, NB-IoT y LTE-M tiene mayor consumo de batería a cambio mejora su calidad del servicio y latencias.

2.3.4.6. Comparativa de coste por usuario

En la Tabla 3 se muestra una comparativa respecto al coste desde una relación de coste por usuario.

	Sigfox	LoRa/LoRaWAN	LTE-M	NB-IoT
Módulo (Dispositivo final)	2-5 €	5 €	20 €	10 €
OPEX	2- 20 €/año	2 €/año	20-30 €/año	20-30 €/año
CAPEX	0 €	100-1000 €/gateway (estación base)	0 €	0 €

Tabla 3. Comparativa LPWAN – Coste por usuario [30]

2.3.4.7. Comparativa de coste de infraestructura

En la Tabla 4 se muestra una comparativa respecto al coste desde una relación de coste de infraestructura.

	Sigfox	LoRa/LoRaWAN	LTE-M	NB-IoT
Módulo (Dispositivo final)	2-5 €	5 €	20 €	10 €
Coste del Espectro	0 €	0 €	500*10 ⁶ €/MHz	500*10 ⁶ €/MHz
Coste de red y despliegue	4000 €/estación base	100 €/gateway 1000 €/estación base	15000 €/estación base	15000 €/estación base

Tabla 4. Comparativa LPWAN – Costes de infraestructura [4]

Hay que destacar que algunas de estas tecnologías están sujetas a nuevos modelos de negocio. En un plano económico, una ventaja adicional de NB-IoT y LTE-M respecto a Sigfox o LoRaWAN, puede venir del mayor consenso entre operadores, la integración en las redes celulares y de la resultante economía de escala.

Debido a que NB-IoT y LTE-M son tecnologías inalámbricas de grado celular que utiliza modulación OFDM, los chips son más complejos, pero los presupuestos de enlace son mejores. Eso significa que los usuarios obtienen el alto nivel de rendimiento asociado con las conexiones celulares, pero a costa de una mayor complejidad y un mayor consumo de energía.

Además, las tecnologías NB-IoT y LTE-M al ser tecnologías procedentes de LTE ya tienen el espectro disponible por lo que realmente no sería necesario invertir en ello, de igual modo las estaciones base ya estarían desplegadas y tan solo sería necesario adaptarlas. Por lo tanto, la diferencia real depende del coste de los módulos finales y adaptación de las estaciones base.

En la Figura 21 se muestra un esquema comparativo de las distintas características de las tecnologías estudiadas.

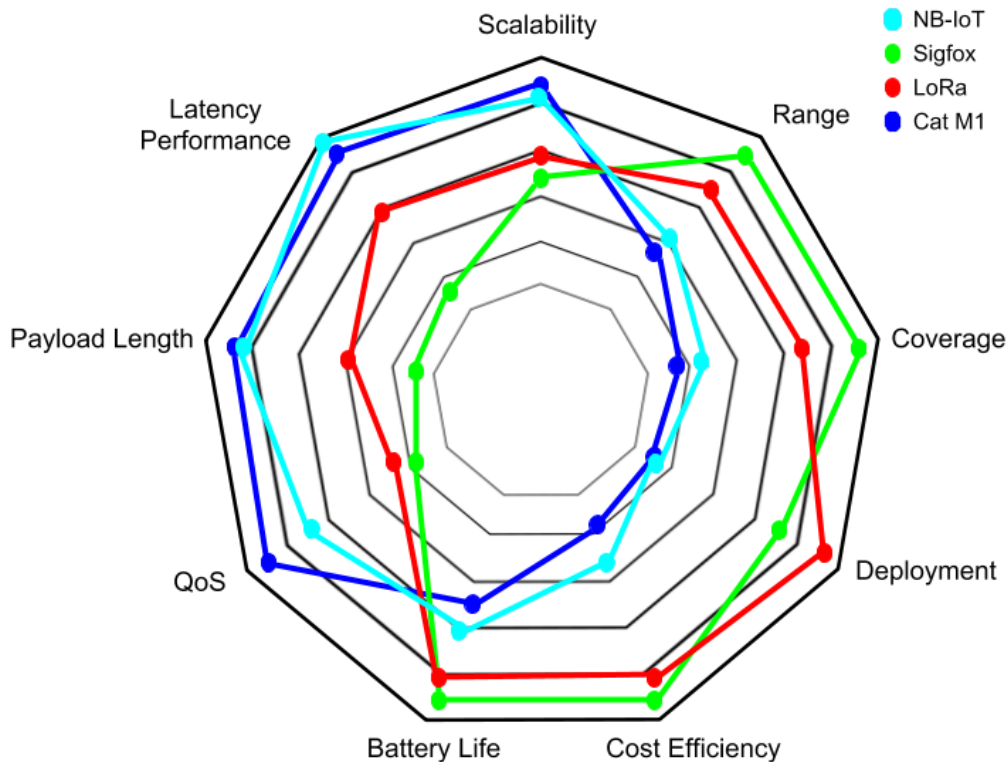


Figura 21. Comparativa LPWAN – Gráfico de tecnologías diseñadas para IoT [31]

2.3.5. Conclusiones

1. Tras el estudio de estas tecnologías se puede concluir que no todas las tecnologías son aptas para todo tipo de aplicaciones IoT.
2. Si se necesitan comunicaciones frecuentes y de baja latencia Sigfox es la peor de las opciones ya que no nos permitirá más de 140 mensajes al día. LoRa/LoRaWAN quedaría en un “puesto” intermedio utilizando dispositivos de clase C, pero sin embargo nos veríamos también obligados a cumplir la legislación europea y la regla del 1%. Por el contrario, NB-IoT y sobre todo LTE-M tienen la posibilidad de tener comunicaciones frecuentes y baja latencia, por lo tanto, esta sería la opción recomendada.
3. Si se necesita principalmente gran ahorro de batería y no importa la latencia la mejor opción es Sigfox, también se puede emplear LoRa/LoRaWAN. Un buen ejemplo sería en zonas agrícolas donde además la cobertura LTE es baja o nula. Además, el rango de cobertura de Sigfox sería especialmente útil en esta circunstancia.
4. Sigfox es la mejor opción para alertas, por ejemplo, de temperatura o alarmas de seguridad donde apenas se producen cambios y en pocas ocasiones se envía información. Lo primordial es la duración de batería y bajos costes.
5. En el caso de seguimiento de vehículos, una buena opción para zonas de baja cobertura LTE es LoRa/LoRaWAN siempre que no se quiera realizar un envío constante de información, es decir, no se podría enviar información en tiempo real, o cada corto periodo de tiempo, ya que no lo permite por un lado la legislación europea (regla del 1%) y

por otro lado hay que tener en cuenta las latencias. En caso de trabajar en zonas urbanas o semiurbanas donde la cobertura LTE es por lo general buena o muy buena la mejor opción sería NB-IoT y especialmente LTE-M para un seguimiento en tiempo real, en esta circunstancia, obviamente el consumo de red, es decir, el consumo de datos sería mayor.

6. En este TFM y aunque a la herramienta web la red de comunicaciones le es abstracta, se pretende enviar información cada 1 minuto, o lo que es lo mismo 60 veces por hora que nos daría un total de $60 \cdot 24 = 1440$ mensajes al día.
7. Por tanto, como se explica en el Anexo 8.1 Sigfox quedaría totalmente descartada (sólo se puede enviar 140 mensajes al día) y en el Anexo 8.2 (depende del SF, el ancho de banda y otros factores) LoRa/LoRaWAN lo cumpliría ya que según el ejemplo podríamos trabajar con hasta 2751 mensajes al día.
8. Sin embargo, si en algún momento se quiere proveer a la herramienta web de la información de manera más rápida, (por ejemplo, en tiempo real) ninguna de estas dos tecnologías sería válida. Además, se va a trabajar en un entorno urbano o semiurbano donde la cobertura LTE está asegurada. Por tanto, la mejor forma de asegurar el buen funcionamiento actual y futuro de la herramienta es utilizar tecnología LTE-M.

2.4. Almacenamiento y acceso a la información

Una de las constantes en los sistemas que dan soporte a la Internet de las cosas son las bases de datos porque ofrecen, frente a la alternativa del almacenamiento en bruto de la información o usando procedimientos de un nivel bajo, un método muy eficaz equilibrado en esfuerzo de instalación, consumo de recursos (procesador y memoria a corto y largo plazo) mantenimiento y rendimiento.

En términos sencillos, existen multitud de soluciones IoT que disponen de servicios que se ejecutan en la nube. Existen distintas alternativas, las más conocidas son Amazon Web Service, Microsoft Azure o Google Cloud.

Por tanto, con la ayuda de alguna de estas soluciones, una vez recopilada la información a través de los dispositivos y enviada por la red de transporte, se va a preservar la información utilizando una base de datos para su almacenamiento y de un servidor de base de datos.

Para poder acceder a estos datos y mostrarlos en una página web es necesario disponer de un servidor web.

2.4.1. Servidor

Un servidor es un ordenador de red de gama alta que gestiona dispositivos conectados (“clientes”) y su acceso a múltiples aplicaciones como recurso central.

Dependiendo del tamaño de la red de una organización, el número de usuarios, los requisitos de accesibilidad, la capacidad de almacenamiento, etc., los servidores pueden configurarse para gestionar una o más funciones de la red. Existen multitud de tipos de servidor: servidor web (de archivos), servidor de base de datos, servidor de email, servidor FTP...

Un solo servidor también es capaz de gestionar varias funciones a la vez, siempre y cuando las especificaciones de hardware satisfagan las demandas de la red.

2.4.2. Servidor de base de datos

Un servidor de base de datos ofrece servicios de almacenamiento y gestión de bases de datos a sus clientes. Permite almacenar grandes cantidades de información.

2.4.2.1. Base de datos

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

Hay programas denominados sistemas gestores de bases de datos, abreviado SGBD (del inglés Database Management System o DBMS), que permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Actualmente, y debido al desarrollo tecnológico de campos como la informática y la electrónica, la mayoría de las bases de datos están en formato digital, siendo este un componente electrónico, por tanto, se ha desarrollado y se ofrece un amplio rango de soluciones al problema del almacenamiento de datos [32].

El almacenamiento de los datos en los sistemas informáticos tradicionales se ha llevado a cabo y todavía se realiza mediante bases de datos relacionales. En este tipo de bases de datos se establecen tablas con campos para la información esperada y se relacionan estas tablas entre sí, siguiendo la relación existente entre los datos. La mayoría de este tipo de bases de datos están basadas en SQL (Structured Query Language).

Por otro lado, con el auge del IoT y el denominado Big Data el uso de las bases de datos no relacionales (denominado NoSQL [Not only SQL]) han ido incrementándose, además existen varios tipos de estas bases de datos.

2.4.2.2. Bases de datos relacionales

El principio de las bases de datos relacionales se basa en la organización de la información en trozos pequeños, que se relacionan entre ellos mediante la relación de identificadores.

En el ámbito informático se habla mucho de ACID, cuyas siglas vienen de las palabras en inglés: atomicidad, consistencia, aislamiento y durabilidad. Son propiedades que las bases de datos relacionales aportan a los sistemas y les permiten ser más robustos y menos vulnerables ante fallos.

La base de datos relacional más usada y conocida es MySQL junto con Oracle, seguida por SQL Server y PostgreSQL, entre otras.

2.4.2.2.1. Bases de datos no relacionales

Como su propio nombre indica, las bases de datos no relacionales son las que, a diferencia de las relacionales, no tienen un identificador que sirva de relación entre un conjunto de datos y otros. La información se organiza normalmente mediante documentos⁵ y es muy útil cuando no tenemos un esquema exacto de lo que se va a almacenar. El formato de documentos más utilizado es el formato JSON (JavaScript Object Notation). Dentro de los principales beneficios de este tipo de base de datos está la fácil escalabilidad y rendimiento [33].

La indiscutible reina del reciente éxito de las bases de datos no relacionales es MongoDB seguida por Redis, Elasticsearch y Cassandra.

En la Figura 22 se muestra una comparativa/transformación de datos entre base de datos relacional y base de datos no relacional documental.



Figura 22. Base de datos relacional vs base de datos no relacional documental [34]

2.4.3. Servidor web

Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o Aplicación del lado del cliente. El código recibido por el cliente es renderizado por un navegador web. Para la transmisión de todos estos datos suele utilizarse algún protocolo. Generalmente se usa el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador [35].

⁵ Existen varios tipos de bases de datos no relacionales: clave-valor (Cassandra), documental (MongoDB), en grafo (Neo4j) y por último orientadas a objetos (Jade).

2.4.3.1. HTTP

El Protocolo de transferencia de hipertexto (Hypertext Transfer Protocol o HTTP) es el protocolo de comunicación que permite las transferencias de información en la World Wide Web.

HTTP fue desarrollado por el World Wide Web Consortium y la Internet Engineering Task Force, colaboración que culminó en 1999 con la publicación de una serie de RFC, el más importante de ellos es el RFC 2616 que especifica la versión 1.1. HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse.

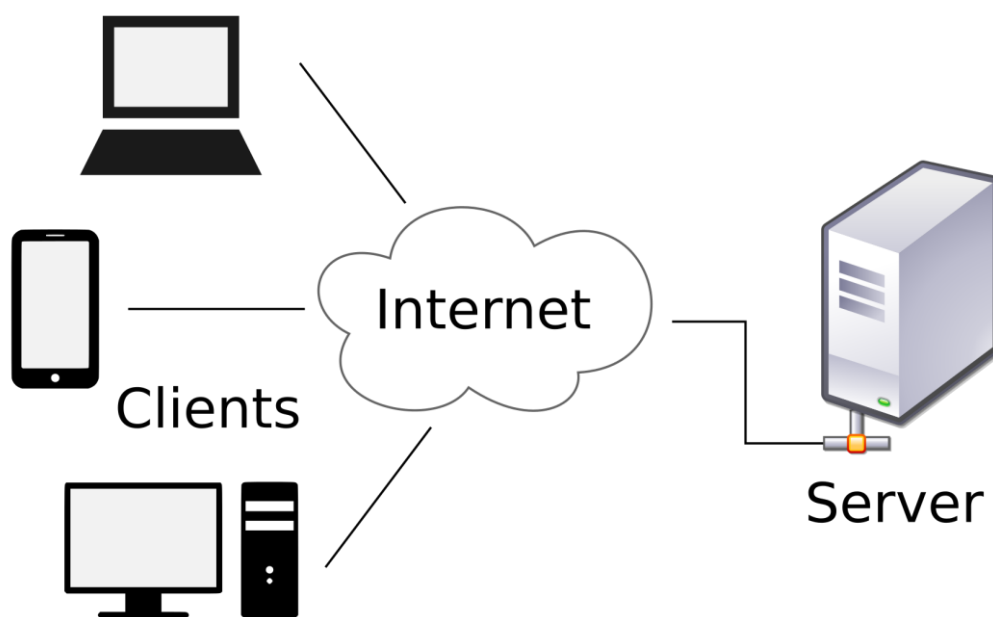


Figura 23. Diagrama cliente-servidor vía Internet

HTTP es un protocolo cliente-servidor (Figura 23), lo que significa que el cliente envía una petición al servidor y espera un mensaje de respuesta del servidor. Es un protocolo sin estado, lo que significa que el servidor no guarda información del cliente, cada petición es independiente de las demás.

2.4.3.2. Mensaje HTTP

Un mensaje HTTP (no importa si es de petición o respuesta) se compone de 3 partes tal y como se muestra en la Figura 24:

- La primera línea (que es diferente para la petición y la respuesta).
- Los encabezados.
- El cuerpo (opcional)



Figura 24. Petición y respuesta HTTP

La primera línea de un mensaje de petición se compone de:

- El verbo
- El recurso
- La versión de HTTP

La primera línea de un mensaje de respuesta se compone de:

- La versión de HTTP
- El código de respuesta

2.4.3.3. El verbo HTTP

La primera línea de un mensaje de petición empieza con un verbo (también se le conoce como método). Los verbos definen la acción que se quiere realizar sobre el recurso. Los verbos más comunes son:

- GET: se utiliza para solicitar un recurso.
- POST: se utiliza para publicar un recurso.
- PUT: se utiliza para reemplazar un recurso.
- DELETE: se utiliza para eliminar un recurso.

2.4.3.4. El código de respuesta

La primera línea de un mensaje de respuesta tiene un código de 3 dígitos que le indica al cliente cómo interpretar la respuesta.

Los códigos de respuesta se dividen en cinco categorías dependiendo del dígito con el que inician:

- 1XX: Información
- 2XX: Éxito
- 3XX: Redirección
- 4XX: Error en el cliente
- 5XX: Error en el servidor

Errores típicos son el error 404 que retorna el servidor cuando el recurso no fue encontrado o el error 500 cuando ocurre un error que el servidor es incapaz de completar.

Existen multitud de servidores web los más populares son:

- **Servidor HTTP Apache:** Es el servidor web más popular y uno de los más utilizados. Es un software de código abierto que se puede instalar en casi todos los sistemas operativos, Linux, Windows e iOS entre ellos.
- **Servidor Ngnix:** Es un servidor web muy ligero que utiliza el proxy inverso, es decir, que protege la identidad de los servidores, lo que mejora la seguridad de toda la información albergada en ellos.
- **Servidor Microsoft IIS:** Es un servidor web desarrollado por Microsoft que ejecuta Windows a través de la tecnología IIS (Internet Information Services). Su gran ventaja es que es compatible con todas las tecnologías de la compañía.
- **Sun Java System Web Server:** Es un servidor web creado para soportar una gran carga de trabajo con tecnologías muy específicas como Java, Perl, Python o Ruby on Rails. Está especialmente pensada para programadores.
- **W3C Jigsaw:** Nació en el World Wide Web Consortium (W3C), un consorcio internacional para establecer los estándares de Internet. Es de código abierto y ha sido programado a partir de Java.

2.4.4.Arquitectura REST

La transferencia de estado representacional (REpresentational State Transfer) o REST es un estilo de arquitectura software para sistemas hipermedia distribuidos como la World Wide Web. El término se originó en el año 2000, en una tesis doctoral sobre la web escrita por Roy Fielding, uno de los principales autores de la especificación del protocolo HTTP y ha pasado a ser ampliamente utilizado por la comunidad de desarrollo.

En la actualidad no existe proyecto o aplicación que no disponga de una API REST para la creación de servicios profesionales a partir de ese software. Twitter, YouTube, los sistemas de identificación con Facebook... hay cientos de empresas que generan negocio gracias a REST y las API's REST. Sin ellas, todo el crecimiento en horizontal sería prácticamente imposible. Esto es así porque REST es el estándar más lógico, eficiente y habitual en la creación de API's para servicios de Internet [34].

REST es una interfaz entre sistemas que usa HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON (Figura 25). Es una alternativa en auge a otros protocolos estándar de intercambio de datos como SOAP (Simple Object Access Protocol), que disponen de una gran capacidad, pero también mucha complejidad. A veces es preferible una solución más sencilla de manipulación de datos como REST.

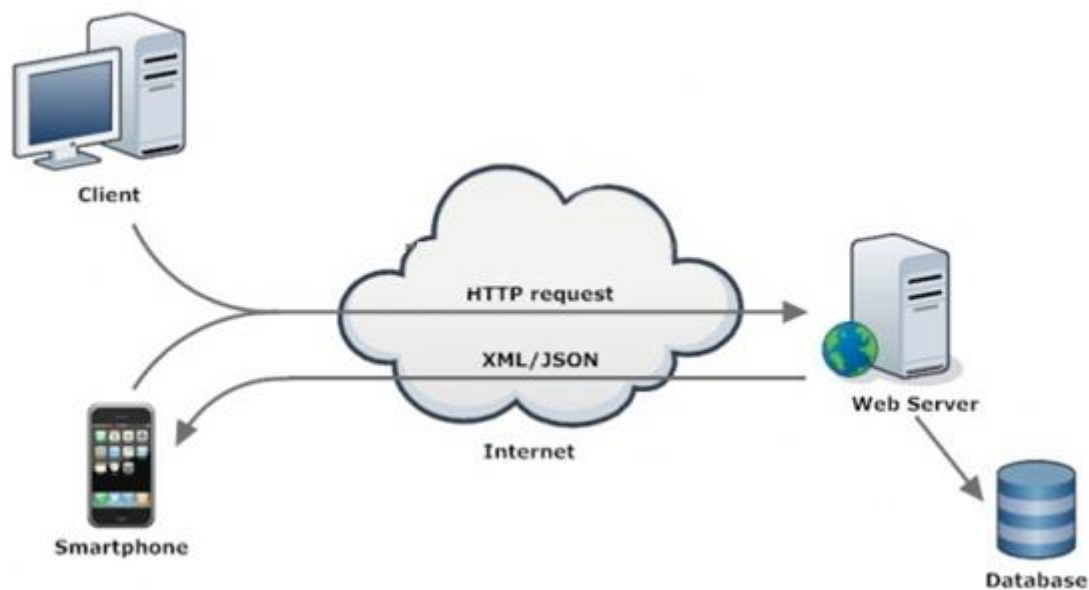


Figura 25. Arquitectura REST

2.4.4.1. Características de REST

- **Protocolo cliente/servidor sin estado:** cada petición HTTP contiene toda la información necesaria para ejecutarla, lo que permite que ni cliente ni servidor necesiten recordar ningún estado previo para satisfacerla. Aunque esto es así, algunas aplicaciones HTTP incorporan memoria caché. Se configura lo que se conoce como protocolo cliente-caché-servidor sin estado: existe la posibilidad de definir algunas respuestas a peticiones HTTP concretas como cacheables, con el objetivo de que el cliente pueda ejecutar en un futuro la misma respuesta para peticiones idénticas. De todas formas, que exista la posibilidad no significa que sea lo más recomendable.
- **Operaciones de datos:** las operaciones más importantes relacionadas con los datos en cualquier sistema REST y la especificación HTTP son cuatro. POST (crear), GET (leer y consultar), PUT (editar) y DELETE (eliminar).
- **Interfaz uniforme:** para la transferencia de datos en un sistema REST, este aplica acciones concretas (POST, GET, PUT y DELETE) sobre los recursos, siempre y cuando estén identificados con una URI. Esto facilita la existencia de una interfaz uniforme que sistematiza el proceso con la información.
- **Sistema de capas:** arquitectura jerárquica entre los componentes. Cada una de estas capas lleva a cabo una funcionalidad dentro del sistema REST.
- **Uso de hipermedios:** hipermedia es un término acuñado por Ted Nelson en 1965 y que es una extensión del concepto de hipertexto. Ese concepto llevado al desarrollo de páginas web es lo que permite que el usuario puede navegar por el conjunto de objetos a través de enlaces HTML. En el caso de una API REST, el concepto de hipermedia explica la capacidad de una interfaz de desarrollo de aplicaciones de

proporcionar al cliente y al usuario los enlaces adecuados para ejecutar acciones concretas sobre los datos.

- **HATEOS:** Para cualquier API REST es obligatorio disponer del principio HATEOAS (Hypermedia As The Engine Of Application State - Hipermedia Como Motor del Estado de la Aplicación) para ser una verdadera API REST. Este principio es el que define que cada vez que se hace una petición al servidor y éste devuelve una respuesta, parte de la información que contendrá serán los hipervínculos de navegación asociada a otros recursos del cliente (Figura 26).

```
{
  - records: [
    - {
      id: 255,
      - coordinates: {
        lng: -4.4581523,
        lat: 36.719933
      },
      lastUpdate: "2019-04-30 23:25:05"
    },
    - {
      id: 644,
      - coordinates: {
        lng: -4.4581523,
        lat: 36.719933
      },
      lastUpdate: "2019-04-30 23:25:05"
    },
    - {
      id: 1766,
      - coordinates: {
        lng: -4.4581523,
        lat: 36.719933
      },
      lastUpdate: "2019-04-30 23:25:05"
    }
  ],
  - links: {
    start: "/api?resource_id=9bc05288-1c11-4eec-8792-d74b679c8fcf",
    next: "/api?offset=0&resource_id=9bc05288-1c11-4eec-8792-d74b679c8fcf"
  },
  total: 25
}
```

Figura 26. Ejemplo de respuesta API REST en formato JSON con HATEOS

2.4.4.2. Ventajas que ofrece REST para el desarrollo

- **Separación entre el cliente y el servidor:** el protocolo REST separa totalmente la interfaz de usuario del servidor y el almacenamiento de datos. Eso tiene algunas ventajas cuando se hacen desarrollos. Por ejemplo, mejora la portabilidad de la interfaz a otro tipo de plataformas, aumenta la escalabilidad de los proyectos y permite que los distintos componentes de los desarrollos se puedan evolucionar de forma independiente.
- **Visibilidad, fiabilidad y escalabilidad:** la separación entre cliente y servidor tiene una ventaja evidente y es que cualquier equipo de desarrollo puede escalar el producto sin excesivos problemas. Se puede migrar a otros servidores o realizar todo tipo de cambios en la base de datos, siempre y cuando los datos de cada una de las peticiones se envíen de forma correcta. Esta separación facilita tener en servidores distintos el frontend y el backend y eso convierte a las aplicaciones en productos más flexibles a la hora de trabajar.
- **La API REST siempre es independiente del tipo de plataformas o lenguajes:** la API REST siempre se adapta al tipo de sintaxis o

plataformas con las que se estén trabajando, lo que ofrece una gran libertad a la hora de cambiar o probar nuevos entornos dentro del desarrollo. Con una API REST se pueden tener servidores PHP, Java, Python o Node.js. Lo único que es indispensable es que las respuestas a las peticiones se hagan siempre en el lenguaje de intercambio de información usado, normalmente XML o JSON.

2.4.5. Conclusiones

1. Para este TFM es necesario la utilización de un servidor de base de datos y un servidor web.
2. Tanto el Frontend como el Backend y la base de datos pueden ir en el mismo servidor o pueden en máquinas diferentes. Hoy en día lo normal es contratar un servicio en la nube (Cloud),
3. La mejor solución es un utilizar una base de datos no relacional documental, ya que es la que mejor se adapta a cualquier tipo de información (es decir, no tiene por qué existir una estructura de datos igual para todos los objetos) que pueda proveer cualquier objeto conectado a la infraestructura IoT.
4. Es indispensable el uso del protocolo HTTP para la transferencia de información web.
5. Es posible utilizar cualquiera de los distintos servidores web, sin embargo, por compatibilidad y simplicidad el mejor a utilizar es el servidor HTTP Apache ya que es el más utilizado y simple de configurar, además de multiplataforma.
6. La mejor alternativa para el consumo de la información es utilizar una arquitectura REST.
7. El intercambio más habitual de información es a través del formato JSON.

2.5. Tecnologías para aplicaciones web del lado del cliente

Las máquinas no necesitan de una interfaz para la representación de los datos, ellas son capaces de entenderse a través del código máquina y código binario. Pero las personas no somos aplicaciones del IoT. Necesitamos de interfaces que nos muestren la información.

Por eso es necesario disponer de herramientas que presenten la información y resultados de una forma sencilla. A través de interfaces y pantallas. Dispositivos que traduzcan la información recopilada por los dispositivos del IoT y analizada por los servidores que están en la nube o por un dispositivo cliente de forma local.

Aquí entran en juego diferentes tecnologías frontend que involucra toda la parte de programación de diseño (toda la parte visual) encargada de la interacción con el usuario.

Un buen punto de partida son las tecnologías web como HTML, CSS y JavaScript. Los estándares actuales del World Wide Web (WWW).

2.5.1.HTML

Es un lenguaje de marcado, diseñado para estructurar textos y definir su presentación en forma de hipertexto, que es el formato estándar de las páginas web.

Gracias a Internet y a los navegadores web, como, por ejemplo, Mozilla Firefox, Google Chrome o Microsoft Edge, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos. Contrariamente a otros lenguajes de programación, el HTML utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determina la forma con la que deben aparecer el texto, así como las imágenes y los demás elementos, en la pantalla del ordenador.

2.5.2.CSS

Es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla. Se trata de una especificación desarrollada por el W3C (World Wide Web Consortium) para permitir la separación de los contenidos de los documentos escritos en HTML, XML, XHTML, SVG, o XUL de la presentación del documento con las hojas de estilo, incluyendo elementos tales como los colores, fondos, márgenes, bordes, tipos de letra, etc, modificando la apariencia de una página web de una forma más sencilla, permitiendo a los desarrolladores controlar el estilo y formato de sus documentos.

2.5.3.JavaScript

Al igual que HTML, JavaScript es un lenguaje de programación que se puede utilizar para construir sitios Web y para hacerlos más interactivos. El lenguaje JavaScript puede interactuar con el código HTML, permitiendo a los programadores web utilizar contenido dinámico. Por ejemplo, hace fácil responder a los acontecimientos iniciados por usuarios, por ejemplo, introducir datos en un formulario. El lenguaje JavaScript es open source, por lo cualquier persona puede utilizarlo sin comprar una licencia.

2.5.4.AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

AJAX es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página, aunque existe la posibilidad de

configurar las peticiones como síncronas de tal forma que la interactividad de la página se detiene hasta la espera de la respuesta por parte del servidor. En JavaScript normalmente se efectúan las funciones de llamada de AJAX mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

AJAX es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

Desarrollar aplicaciones AJAX requiere un conocimiento avanzado de todas y cada una de las tecnologías de la Figura 27.

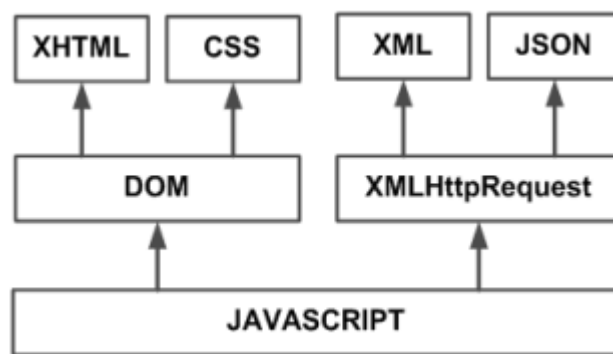


Figura 27. Tecnologías agrupadas bajo el concepto de AJAX [36]

En las aplicaciones web tradicionales, las acciones del usuario en la página (pinchar en un botón, seleccionar un valor de una lista, etc.) desencadenan llamadas al servidor. Una vez procesada la petición del usuario, el servidor devuelve una nueva página HTML al navegador del usuario.

En la Figura 28, la imagen de la izquierda muestra el modelo tradicional de las aplicaciones web. La imagen de la derecha muestra el nuevo modelo propuesto por AJAX.

La técnica tradicional para crear aplicaciones web funciona correctamente, pero no crea una buena sensación al usuario. Al realizar peticiones continuas al servidor, el usuario debe esperar a que se recargue la página con los cambios solicitados. Si la aplicación debe realizar peticiones continuas, su uso se convierte en algo molesto.

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano.

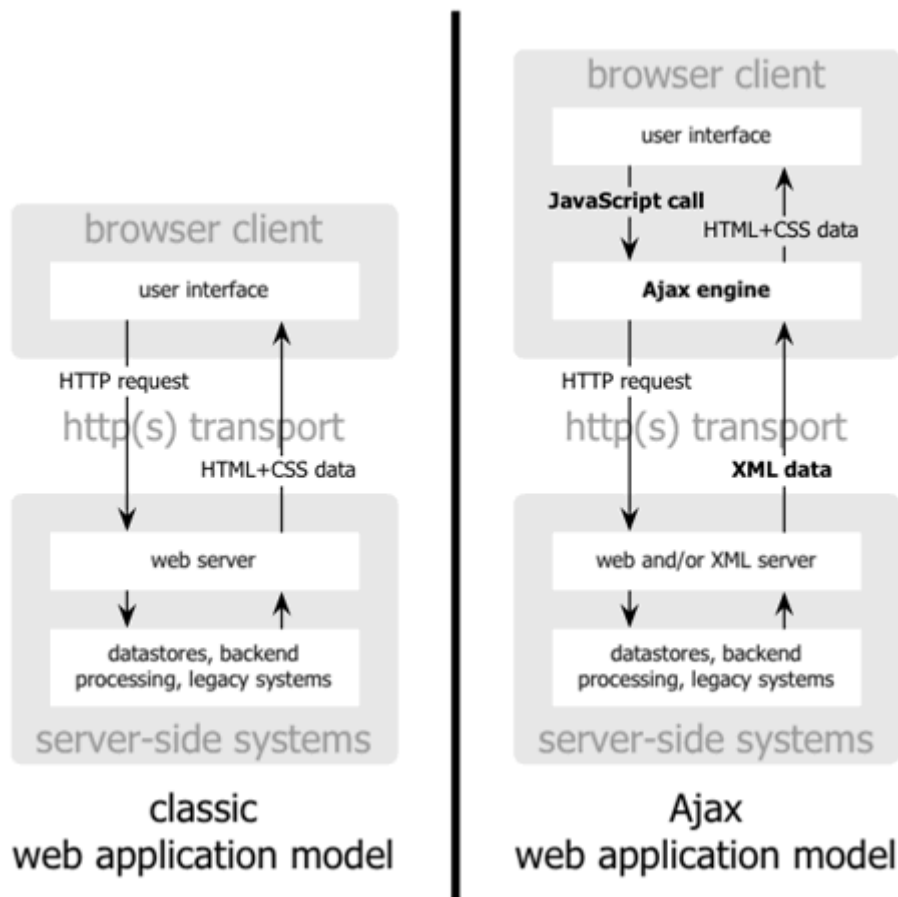


Figura 28. Comparación gráfica del modelo tradicional de aplicación web y del nuevo modelo propuesto por AJAX [36]

Las aplicaciones construidas con AJAX eliminan la recarga constante de páginas mediante la creación de un elemento intermedio entre el usuario y el servidor. La nueva capa intermedia de AJAX mejora la respuesta de la aplicación, ya que el usuario nunca se encuentra con una ventana del navegador vacía esperando la respuesta del servidor.

La Figura 29 muestra la diferencia más importante entre una aplicación web tradicional y una aplicación web creada con AJAX. La imagen superior muestra la interacción síncrona propia de las aplicaciones web tradicionales. La imagen inferior muestra la comunicación asíncrona de las aplicaciones creadas con AJAX.

Las peticiones HTTP al servidor se sustituyen por peticiones JavaScript que se realizan al elemento encargado de AJAX. Las peticiones más simples no requieren intervención del servidor, por lo que la respuesta es inmediata. Si la interacción requiere una respuesta del servidor, la petición se realiza de forma asíncrona mediante AJAX. En este caso, la interacción del usuario tampoco se ve interrumpida por recargas de página o largas esperas por la respuesta del servidor.

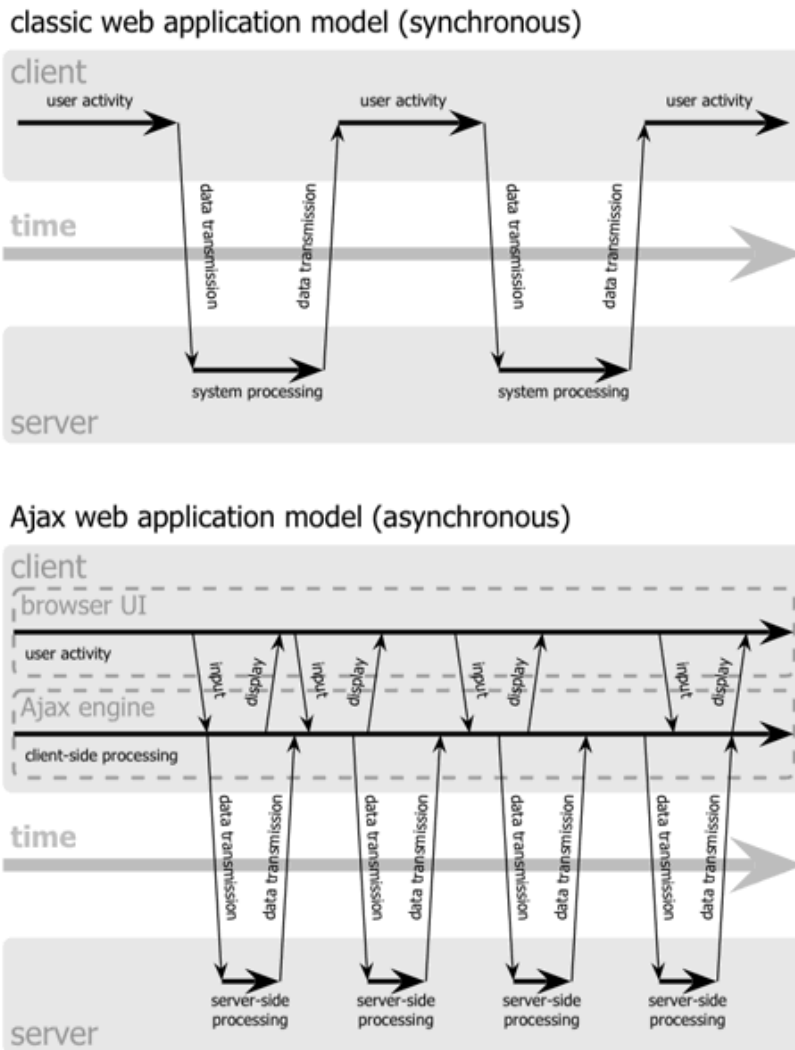


Figura 29. Comparación entre las comunicaciones síncronas de las aplicaciones web tradicionales y las comunicaciones asíncronas de las aplicaciones AJAX [36]

2.5.5.SPA (Single Page Application)

Una Single Page Application (SPA), o aplicación de página única es una aplicación web o es un sitio web que cabe en una sola página con el propósito de dar una experiencia más fluida a los usuarios como una aplicación de escritorio. En un SPA todos los códigos de HTML, JavaScript, y CSS se carga de una vez [37] o los recursos necesarios se cargan dinámicamente como lo requiera la página y se van agregando, normalmente como respuesta de las acciones del usuario. La página no tiene que cargar otra vez en ningún punto del proceso tampoco se transfiere a otra página, aunque las tecnologías modernas pueden permitir la navegabilidad en páginas lógicas dentro de la aplicación. La interacción con las aplicaciones de página única puede involucrar comunicaciones dinámicas con el servidor web que está detrás.

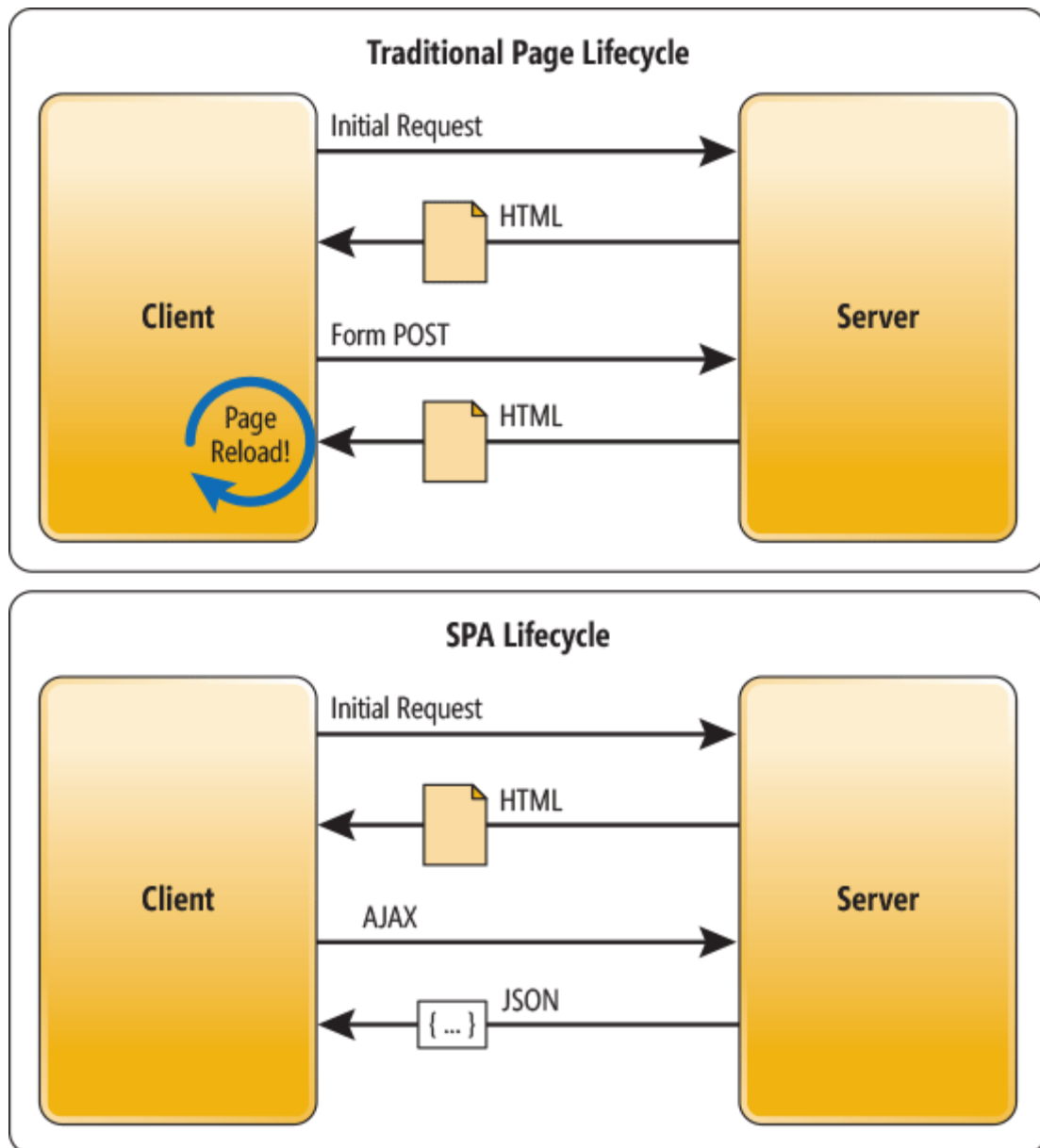


Figura 30. Ciclo de vida de una página web tradicional vs ciclo de vida de una SPA [38]

En cuanto a la arquitectura de la aplicación, la mayor parte de la funcionalidad se lleva al cliente. Lo podríamos ver como un fat-client⁶ que se carga desde un servidor web. El código en servidor se usa básicamente para proveer de una API REST a nuestro código cliente usando AJAX [38].

⁶ Se denomina fat-client (cliente pesado) al programa "cliente" de una arquitectura cliente-servidor cuando la mayor carga de cómputo está desplazada hacia la computadora que ejecuta dicho programa.

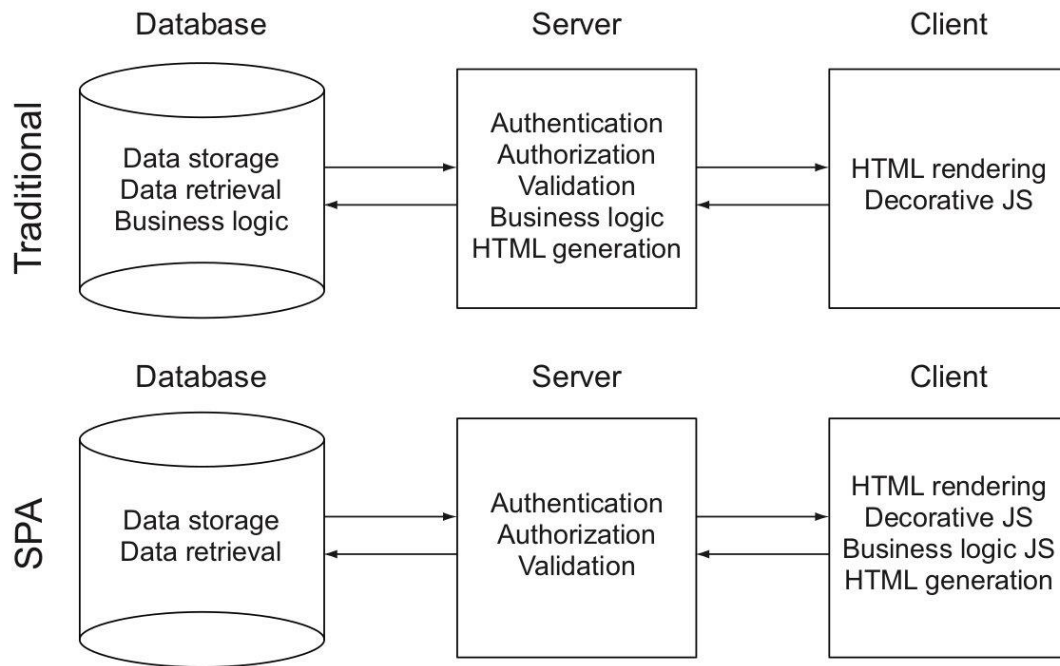


Figura 31. Arquitectura página web tradicional vs página SPA [38]

2.5.6. Framework

A partir de los lenguajes y técnicas descritas en los apartados anteriores han surgido frameworks para acelerar y mejorar el proceso de construcción de la web.

Un framework es una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación web. Es considerada una aplicación genérica incompleta y configurable a la que el desarrollador le puede añadir las piezas restantes para construir una aplicación web completa.

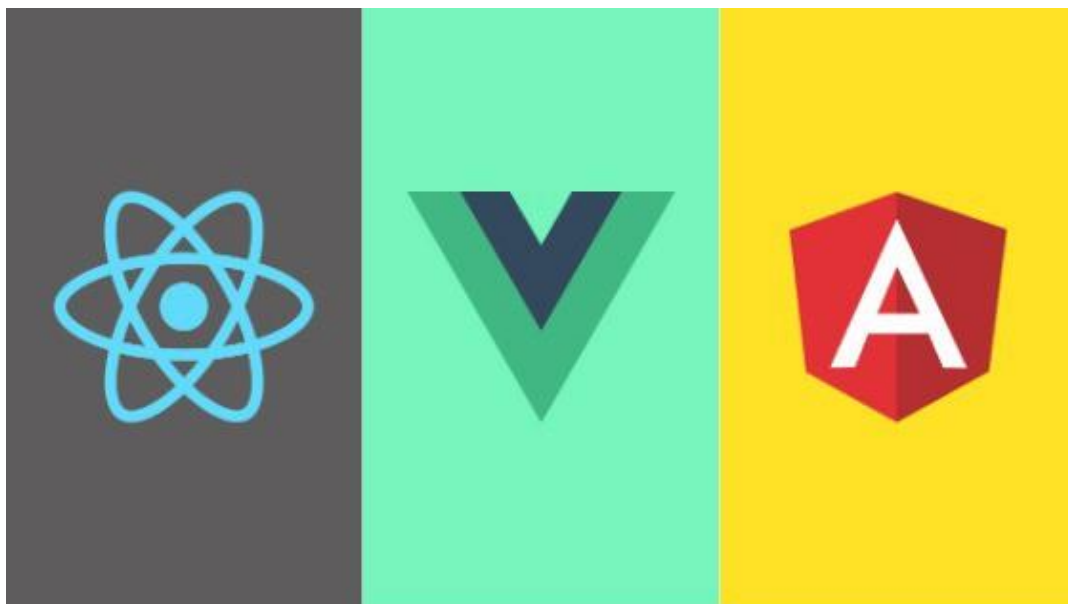


Figura 32. De izquierda a derecha, logos de los framework React, VueJS y Angular [39]

Existen multitud de frameworks frontend, en la actualidad los más relevantes son Angular, VueJS y React [39] (en la Figura 32 se muestran los logos identificativos de estos frameworks).

2.5.6.1. Angular

Es un framework de Google fundado en 2009 es increíble para la creación de aplicaciones web altamente interactivas.

- Beneficios:
 - Compilación rápida (en menos de 3 segundos), lanzamiento nuevo de módulos para trabajar con HTTP (HttpClient).
 - Documentación detallada que permite obtener toda la información necesaria para el desarrollador. Sin embargo, esto requiere más tiempo para el aprendizaje.
 - Enlace de datos bidireccional que permite un comportamiento singular para la aplicación que minimiza los riesgos de posibles errores.
 - Utiliza el patrón MVVM (Model-View-ViewModel) que permite al desarrollador trabajar por separado en la misma sección de la aplicación utilizando el mismo conjunto de datos.
 - Inyección de dependencia de las características relacionadas con los componentes con módulos y modularidad en general.
- Desventajas:
 - La sintaxis compleja que proviene de la primera versión de Angular. Sin embargo, a partir de la versión de Angular 2 usa TypeScript 2.4, que es el menos difícil de aprender en comparación, aunque sigue teniendo una gran curva de aprendizaje.
 - Problemas de migración que pueden aparecer al pasar de la versión anterior a la última.

2.5.6.2. VueJS

Framework de JavaScript, lanzado en 2013, que se adapta perfectamente a la creación de interfaces de usuario altamente adaptables y sofisticadas aplicaciones de una sola página.

- Beneficios:
 - Gran escala. Vue.js puede ayudar a desarrollar plantillas reutilizables bastante grandes que se pueden crear sin tiempo adicional asignado para eso según su estructura simple.
 - Diminuto tamaño. Vue.js puede pesar alrededor de 20 KB manteniendo su velocidad y flexibilidad que permite alcanzar un rendimiento mucho mejor en comparación con otros framework.
 - Adaptabilidad. Proporciona un período de cambio rápido de otros frameworks a Vue.js debido a la similitud con Angular y React en términos de diseño y arquitectura utilizando el patrón MVVM.
 - Impresionante integración. Vue.js se puede utilizar tanto para crear aplicaciones de una sola página como para interfaces web de

aplicaciones más difíciles. Lo principal es que las partes interactivas más pequeñas se pueden integrar fácilmente en la infraestructura existente sin ningún efecto negativo en todo el sistema.

- HTML habilitado. Esto significa que Vue.js tiene muchas características similares con Angular y esto puede ayudar a optimizar el manejo de bloques HTML con el uso de diferentes componentes.
- Documentación detallada Vue.js tiene una documentación muy circunstancial que puede ajustar la curva de aprendizaje para los desarrolladores y ahorrar mucho tiempo para desarrollar una aplicación utilizando solo los conocimientos básicos de HTML y JavaScript.

- Desventajas:

- Riesgo de exceso de flexibilidad. A veces, Vue.js puede tener problemas al integrarse en grandes proyectos y todavía no hay experiencia con posibles soluciones, pero definitivamente vendrán pronto.

2.5.6.3. ReactJS

La biblioteca de JavaScript ideal para la construcción de grandes aplicaciones web donde los datos se pueden cambiar de forma dinámica.

- Beneficios:

- Fácil de aprender. Reaccionar es mucho más fácil de aprender debido a su simplicidad en términos de sintaxis. Los ingenieros solo necesitan recordar sus habilidades de escritura HTML y eso es todo. No es necesario aprender TypeScript como en Angular.
- Alto nivel de flexibilidad y máximo de capacidad de respuesta.
- DOM virtual (modelo de objeto de documento) que permite organizar documentos en formatos HTML, XHTML o XML en un árbol desde el cual es mejor aceptado por los navegadores web al analizar diferentes elementos de la aplicación web.
- Combinado con ES6 / 7, ReactJS puede trabajar con la alta carga de una manera fácil.
- Enlace de datos hacia abajo, lo que significa que, con este tipo de flujo de datos, los elementos secundarios no pueden afectar los datos principales.
- Biblioteca de JavaScript de código abierto 100% que obtiene muchas actualizaciones y mejoras diarias según las contribuciones de desarrolladores de todo el mundo.
- Absolutamente liviano porque los datos que se ejecutan en el lado del usuario se pueden representar fácilmente en el lado del servidor simultáneamente.
- La migración entre versiones es generalmente muy fácil, con Facebook proporcionando "codemods" para automatizar gran parte del proceso.

- Desventajas:
 - Falta de documentación oficial: el desarrollo súper rápido de ReactJS no deja lugar para la documentación adecuada, que ahora es un poco caótica, ya que muchos desarrolladores contribuyen de forma individual sin ningún enfoque sistemático.
 - React es no ionizado, lo que significa que los desarrolladores a veces tienen demasiadas opciones.
 - Mucho tiempo para dominar, lo que significa que ReactJS requiere un conocimiento profundo de cómo integrar la interfaz de usuario en el framework de MVC.

2.5.7. Conclusiones

1. A partir de las distintas tecnologías Web (HTML, CSS, JavaScript, AJAX), han surgido framework para facilitar el diseño e implementación de la web.
2. Es imprescindible tener un buen conocimiento de estas tecnologías para aprovechar el desarrollo de la estructura, apariencia y programación de la web.
3. Una SPA facilitará la carga inicial de la herramienta web a 1 sola petición y dará una experiencia más fluida al usuario.
4. Comparando las ventajas e inconvenientes descritas anteriormente de los framework la mejor opción es realizar el desarrollo de la herramienta web utilizando el framework VueJS ya que es fácil de implementar, y tiene una curva de aprendizaje rápida. Además, dispone de buena documentación, librerías (dependencias) y complementos o “plugins” para acelerar el proceso de creación de la herramienta web y mejoras en la visualización e interfaz de usuario.

3.Requisitos del sistema, diseño e implementación

3.1.Requisitos y configuración

En este apartado se especifican los requisitos y configuraciones para el diseño e implementación de la herramienta web para una flota de vehículos.

3.1.1.Requisitos de red, software, funcionales y de diseño

Para el desarrollo de esta herramienta web es necesario:

- Disponer de un PC (el sistema operativo es indiferente, pero se ha utilizado Windows 10).
- Node.js y npm [40].
- Vue CLI [41].
- Yarn [42].

En el Anexo 8.6 se especifica con más detalle los comandos para la puesta en funcionamiento del proyecto tanto en modo desarrollo como en producción.

En este TFM se ha realizado un diseño e implementación basado en una flota de vehículos de autobuses, la herramienta web tiene los siguientes requisitos y características:

- Se geolocalizará el autobús con un GPS.
- Se utilizará una red LPWAN para la transmisión de la información.
 - La información y la geolocalización se actualizará cada 1 minuto o incluso en menor tiempo, es decir, se cumplirá una latencia inferior a 1 minuto, y la calidad del servicio tiene que estar garantizada.
 - Debe tener cobertura en toda la región donde los vehículos se muevan (en este caso en la localidad de Málaga), que viene representado aproximadamente por la Figura 33.

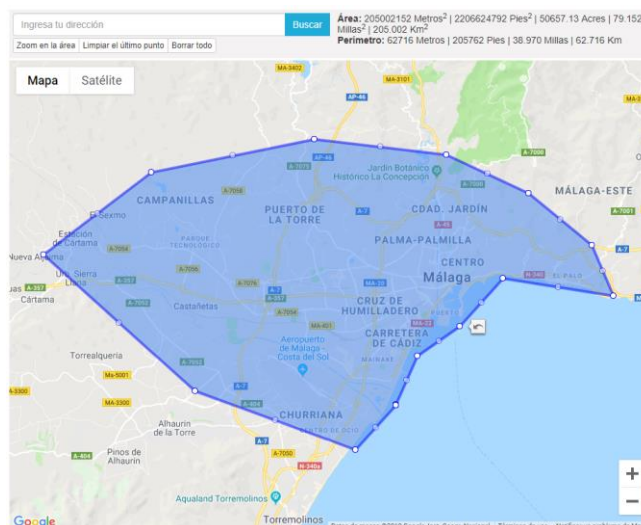


Figura 33. Zona de movimiento de los vehículos de la flota

- Un autobús tiene las siguientes características más relevantes:
 - Identificación del vehículo (información estática)
 - Combustible (información estática)
 - Diesel
 - Gas
 - Híbrido
 - Tipo de vehículo (información estática)
 - Autobús (normal)
 - Autobús articulado
 - Autobús de doble altura
 - Microbús
 - Código de línea (información estática)
 - Código de la primera parada (información estática)
 - Identificación del conductor (información dinámica)
 - Sentido (información dinámica)
 - Latitud (información dinámica)
 - Longitud (información dinámica)
 - Última actualización (información dinámica)
 - Estado (información calculada)
 - Parado (el vehículo lleva más de 1 minuto en la misma ubicación o la última actualización es mayor a 1 minuto)
 - En movimiento
 - “Otra información” (información caculeada)
- Mostrar la geolocalización de los vehículos utilizando Google Maps.
- Mostrar una tabla con toda la información de la flota de vehículos.
- Un usuario tiene que iniciar sesión para acceder a la herramienta web.
- La herramienta será responsiva, es decir, se podrá visualizar en cualquier dispositivo móvil, PC o Tablet.
- Los textos se deben mostrar en los idiomas inglés y español.

3.1.2. Google Maps

El servidor de aplicaciones de mapas de Google nos permite utilizar su funcionalidad mediante API's para poder cargar el mapa. Para ello es imprescindible registrarse en Google Cloud Platform y obtener la API Key de Google Maps⁷.

Google Maps provee a los desarrolladores distintas alternativas de API's según la plataforma desde la que se quiere acceder. En este TFM es necesario la Google Maps API para JavaScript.

3.1.2.1. API de Google Maps para JavaScript

Para este TFM además de la API Key es necesario utilizar principalmente 3 opciones de la API para JavaScript [43]:

- **Mapa:** Es decir, la declaración del mapa principal donde aparecen los vehículos geolocalizados.

⁷ Ver Anexo 8.5

- **Marcadores:** Son los elementos para representar, en este caso cada vehículo será un marcador. Además, tiene la opción de modificar el icono o la imagen a mostrar.
- **Ventanas de información:** Es una ventana superpuesta en el mapa que por lo general se muestra cuando se clicca en el marcador para mostrar más información acerca de él.

En la Figura 34 se muestra el código básico HTML + CSS + JavaScript para implementar en el mapa un marcador con una ventana de información. En la Figura 35 se muestra el resultado del código.

```

1  <!DOCTYPE html>
2  <html>
3  <head>
4  </head>
5  <body>
6  <div id="map"></div>
7  <script async defer
8  src="https://maps.googleapis.com/maps/api/js?key=AIzaSyD1z0hvqa*****.*****qBL2Ro&callback=initMap">
9  </script>
10 <script>
11   function initMap() {
12     var myLatLng = { lat: 36.7123103, lng: -4.448098 }
13
14     var map = new google.maps.Map(document.getElementById('map'), {
15       zoom: 13,
16       center: myLatLng
17     })
18
19     var infowindow = new google.maps.InfoWindow({
20       content: 'A info window example'
21     })
22
23     var marker = new google.maps.Marker({
24       position: myLatLng,
25       map: map,
26       title: 'A marker'
27     })
28
29     marker.addListener('click', function () {
30       infowindow.open(map, marker);
31     })
32   }
33 </script>
34 <style>
35   html,
36   body {
37     height: 100%;
38     margin: 0;
39     padding: 0;
40   }
41   #map {
42     height: 100%;
43   }
44 </style>
45 </body>
46 </html>
47

```

Figura 34. Código básico Mapa + Marcador + InfoWindow

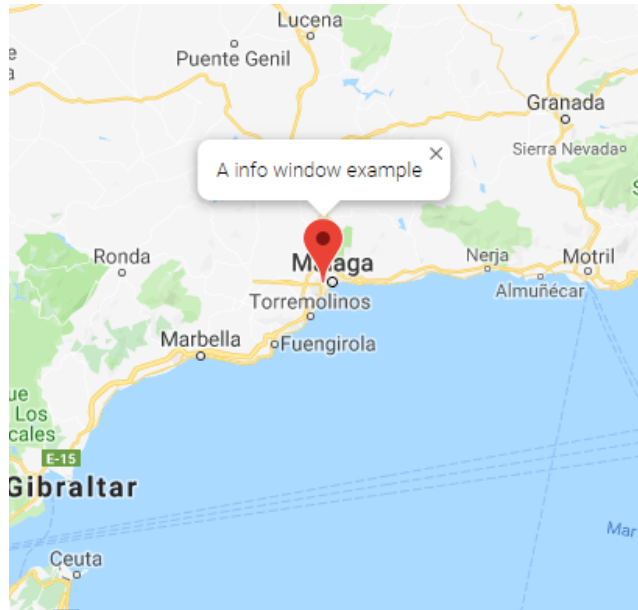


Figura 35. Resultado del código de la Figura 34

3.2. Diseño e implementación

3.2.1. Arquitectura completa del sistema

En la Figura 36 se muestra la arquitectura completa del sistema.

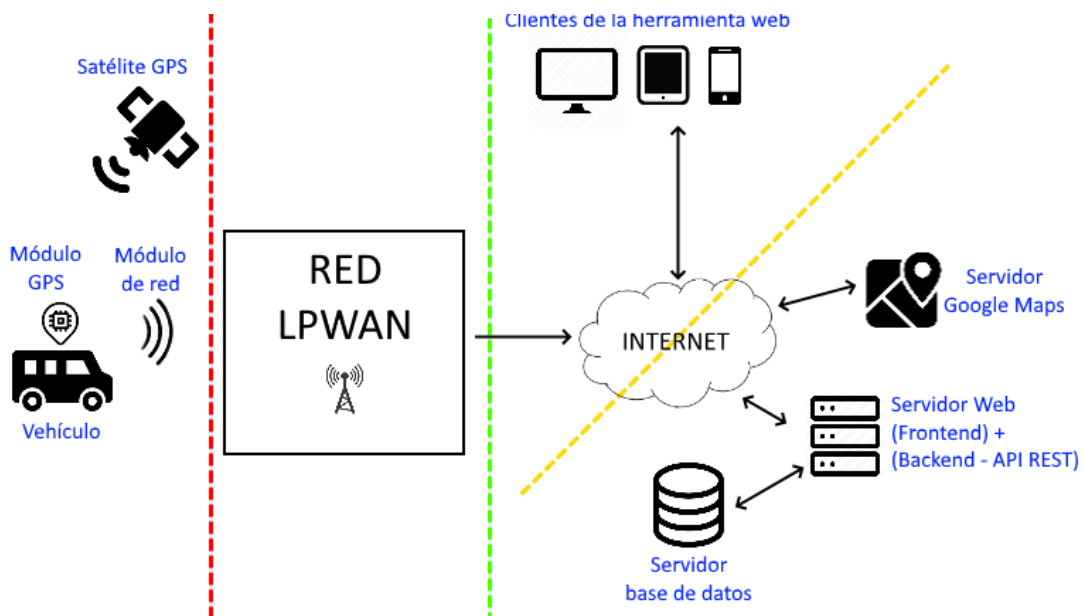


Figura 36. Arquitectura completa del sistema

El vehículo dispone de un módulo GPS que adquiere las coordenadas a través del posicionamiento global utilizando los satélites GPS. Mediando un módulo de red LTE-M transmite toda la información del vehículo a través de una red de comunicaciones LPWAN LTE-M. El sistema guarda la información en la base de datos a través de un servidor y con la programación del backend mediante protocolo AJAX HTTP y método POST.

Cuando el cliente quiere ver la información llama al servidor web mediante AJAX HTTP y método GET que le entrega los archivos de herramienta web (Frontend) al completo ya que se trata de una SPA (Single Page Application), es el propio navegador web del cliente quien renderiza y transforma los archivos entregados por el servidor web para mostrar el diseño de la herramienta web.

Mediante consultas a la API REST del Backend desde el navegador web del cliente con consultas AJAX HTTP (método GET, por lo general o POST según el caso) se consulta a la base de datos y se va actualizando la información en el navegador del cliente.

Como además hay que posicionar el vehículo en el mapa es necesario una consulta adicional mediante AJAX con método GET, esta vez al servidor de Google Maps para traer toda la información del mapa y posicionar correctamente el vehículo según las coordenadas obtenidas a lo largo del proceso, y por tanto, el servidor de Google Maps finalmente entrega al navegador del cliente la posición correcta donde colocar el vehículo en el mapa implementado en la herramienta web.

3.2.2.Estado de la red LPWAN

Como se indicó en la Figura 33 los vehículos abarcan una región por donde realizan sus movimientos de aproximadamente 205 Km².

Las antenas de la región de la Figura 33 deberán ser adaptadas a LTE-M (en caso de que no lo estén).

Para asegurar la correcta recepción de los datos tal y como se indica para LTE-M se puede tener un MCL de hasta 157 dB. Lo ideal es realizar pruebas de medidas en el mayor número de puntos posibles y a distintas horas del día dentro de la región de la Figura 33 y asegurar que en todos los lugares se obtiene medidas iguales o mejores al MCL (sino se cumple habría que añadir más antenas adaptadas a LTE-M) y además que se cumple la calidad de servicio y latencias ofrecidos por LTE-M, en este caso, se debe cumplir que el tiempo sea igual o inferior a 1 minuto.

3.2.3.Diseño del modelo de datos

En la Tabla 5 se muestra el modelo de los datos para un autobús y en la Figura 37. Ejemplo JSON modelo de un autobús un ejemplo del JSON resultante.

Clave	Tipo	
id	Integer	Estático
lineCode	Integer	Estático
vehicleType	String	Estático
fuel	String	Estático
firstStopCode	Integer	Estático
direction	Integer	Dinámico
lng	Integer	Dinámico
lat	Integer	Dinámico
lastUpdate	Date	Dinámico
driver	Integer	Dinámico
status	Boolean	Dinámico
lastWorkingTime	Integer	Calculado
totalWorkingTime	Integer	Calculado

Tabla 5. Modelo de los datos para un autobús

```

1  {
2      // Static data
3      "id": 1689,
4      "fuel": "diesel",
5      "vehicleType": "bus",
6      "firstStopCode": 1525,
7      "lineCode": 15,
8      // Dynamic data
9      "driver": 488,
10     "direction": 1,
11     "lng": -4.4479127,
12     "lat": 36.6892,
13     "lastUpdate": "2019-04-08 08:04:25",
14     "status": false,
15     // Computed data
16     "lastWorkingTime": 0,
17     "totalWorkingTime": 4020
18 }
19

```

Figura 37. Ejemplo JSON modelo de un autobús

En la Tabla 6 se muestra el modelo de los datos para un usuario y en la Figura 38. Ejemplo JSON modelo de un usuario un ejemplo del JSON resultante.

Clave	Tipo
id	Integer
username	String
password ⁸	String

Tabla 6. Modelo de datos de un usuario

⁸ La contraseña (password) se guarda encriptada.


```

1 {
2   "id": 1,
3   "username": "tfm",
4   "password": "tfm"
5 }
6

```

Figura 38. Ejemplo JSON modelo de un usuario

3.2.4. Diagrama de estados y navegación

La herramienta web consiste en una pantalla principal:

1. Inicio de sesión (Login),

donde el usuario introduce su usuario y contraseña, si es incorrecto le aparece un mensaje de error para volver a insertar el usuario y la contraseña. Si es correcto accede a las pantallas:

2. Tabla de listado de vehículos,
3. Mapa de geolocalización de vehículos.

Tanto en la pantalla 2 como en la pantalla 3 se puede cerrar la sesión para volver a la pantalla inicial de Inicio de sesión.

En cualquiera de las pantallas se puede cambiar de idioma. Por defecto el idioma principal es el español.

En la Figura 39 se especifica el diagrama de estados y navegación de la herramienta web.

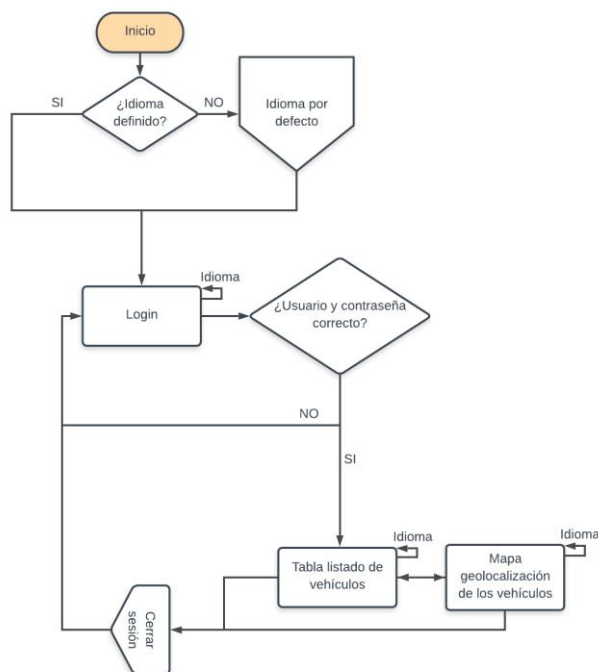


Figura 39. Diagrama de estados y navegación de la herramienta web

3.2.5. Esqueleto de la aplicación web

En la Figura 40 se muestra el esqueleto de la aplicación desarrollada. Esta estructura es importante debido a que es la utilizada en la mayoría de las aplicaciones web y concretamente con el framework VueJS y facilita la comprensión, la implementación y el desarrollo.

Las partes del esqueleto más relevantes a destacar son:

- El fichero `main.js` es el fichero inicial de configuración y carga de todo lo relacionado con el framework VueJS.
- En la carpeta **views** se encuentran las distintas pantallas/vistas asociadas a las rutas (fichero `router.js`) en las cuales tiene acceso un usuario.
- La carpeta **components** contiene componentes que se utilizan en distintas partes de la aplicación, es decir, trozos de código que pueden ser usadas según sea necesario en cualquier parte de la aplicación.
- En la carpeta **locale** se encuentran los distintos idiomas disponibles para la aplicación web.
- En la carpeta **public** se han guardado las imágenes estáticas de los distintos vehículos.
- En la carpeta **api** se encuentra la configuración de la URL y parámetros de llamada a la API REST.
- La carpeta **node_modules** contiene las dependencias de desarrollo para realizar la herramienta web.

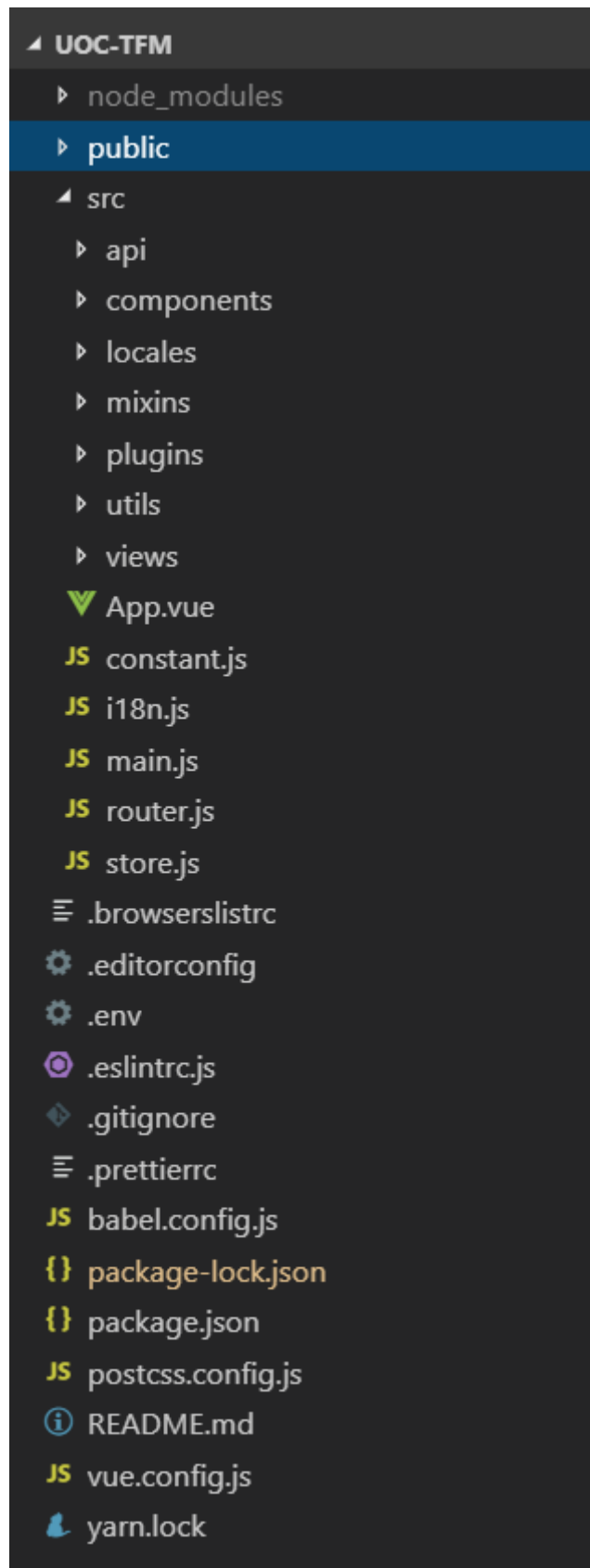


Figura 40. Esqueleto de la aplicación web

3.2.6. Librerías y dependencias

En el archivo **package.json** se recogen todas las librerías y dependencias del proyecto:

```
{
  "name": "uoc-tfm",
  "version": "1.0.0",
  "private": true,
  "scripts": {
    "serve": "vue-cli-service serve",
    "build": "vue-cli-service build",
    "lint": "vue-cli-service lint"
  },
  "dependencies": {
    "axios": "^0.18.0",
    "vue": "^2.6.6",
    "vue-i18n": "^8.9.0",
    "vue-i18n-filter": "^0.1.6",
    "vue-router": "^3.0.1",
    "vue2-google-maps": "^0.10.6",
    "vuetify": "^1.5.5",
    "vuex": "^3.0.1"
  },
  "devDependencies": {
    "@mdi/font": "^3.5.95",
    "@vue/cli-plugin-babel": "^3.5.0",
    "@vue/cli-plugin-eslint": "^3.5.0",
    "@vue/cli-plugin-pwa": "^3.5.0",
    "@vue/cli-service": "^3.5.0",
    "@vue/eslint-config-prettier": "^4.0.1",
    "babel-eslint": "^10.0.1",
    "eslint": "^5.8.0",
    "eslint-plugin-vue": "^5.0.0",
    "fibers": "^3.1.1",
    "sass": "^1.17.2",
    "sass-loader": "^7.1.0",
    "vue-cli-plugin-i18n": "^0.5.2",
    "vue-cli-plugin-vuetify": "^0.5.0",
    "vue-template-compiler": "^2.5.21"
  }
}
```

Como se ha indicado la herramienta web está basada en el framework VueJS, y por tanto se han añadido librerías y dependencias para el desarrollo y el pase a producción del proyecto para este framework.

Entre todas las dependencias de este proyecto es importante remarcar:

- **Vuetify** [44]: es un framework que combina la potencia de VueJS con la estética de Material Design. Permite acelerar el desarrollo de aplicaciones web complejas, incorporando una gran cantidad de componentes listos para usar, además de hacer la web responsiva.
- **vue2-google-maps** [45]: Es una librería que facilita la interacción entre VueJS y Google Maps para la creación del mapa, los marcadores y la información mostrada en el mapa.
- **vue-i18n** [46]: Esta librería facilita la internacionalización de la página para multitud de idiomas.

3.2.7. Presentación de la herramienta web

En la Figura 41 se observa la Pantalla de Inicio de sesión. Se puede escoger el idioma. El usuario debe introducir su nombre de usuario y contraseña correctamente para poder acceder a la herramienta web.

UOC: Trabajo fin de Máster

Herramienta web para la monitorización de vehículos

Por favor inicie sesión

Español ▾

Usuario
tfm

Contraseña
•••

INICIAR SESIÓN

Figura 41. Pantalla de Inicio de sesión

En la Figura 42 se observa la Pantalla Tabla de vehículos. Se puede escoger idioma en la parte superior derecha. Se puede cambiar a la pantalla de Mapa. Se puede Cerrar sesión. Aparece una tabla con el listado de vehículos y su información por cada uno de ellos. Además, se puede realizar búsquedas dentro de la propia información de la tabla.

Tabla de vehículos

TABLA DE VEHÍCULOS MAPA CERRAR SESIÓN Español

Listado de vehículos

Buscar en la tabla

Coordenadas	Id ↑	Tipo de vehículo	Id del conductor	Estado	Sentido	Parada inicial	Combustible	Última actualización	Código de línea	T. út. trabajo (min)	T. total (trabajo) (horas)
Lat. 36.71670, Long. -4.42898	1612	Microbús	695	En movimiento	1	416	Diesel	2019-05-19 12:31:05	1	74	3823
Lat. 36.71788, Long. -4.42879	1618	Microbús	Sin conductor	Parado	2	1463	Diesel	2019-05-11 22:07:05	31	0	4510
Lat. 36.66586, Long. -4.46679	1649	Autobús	650	En movimiento	2	1011	Diesel	2019-05-19 12:31:05	5	184	2302
Lat. 36.71672, Long. -4.35224	1671	Autobús	540	En movimiento	2	1119	Diesel	2019-05-19 12:31:05	11	217	4094
Lat. 36.74829, Long. -4.42635	1688	Autobús articulado	Sin conductor	Parado	2	1713	Diesel	2019-05-18 23:20:05	15	0	2537

Filas por página: 5 1-5 de 25

Figura 42. Pantalla Tabla de vehículos

En la Figura 43 se observa el Mapa con la geolocalización y estado de los vehículos. Se puede escoger idioma en la parte superior derecha. Se puede cambiar a la pantalla Tabla de vehículos. Se puede cerrar sesión. Aparece el mapa de Google Maps con los vehículos geolocalizados y coloreados en negro si están parados o en azul si se encuentran en movimiento.

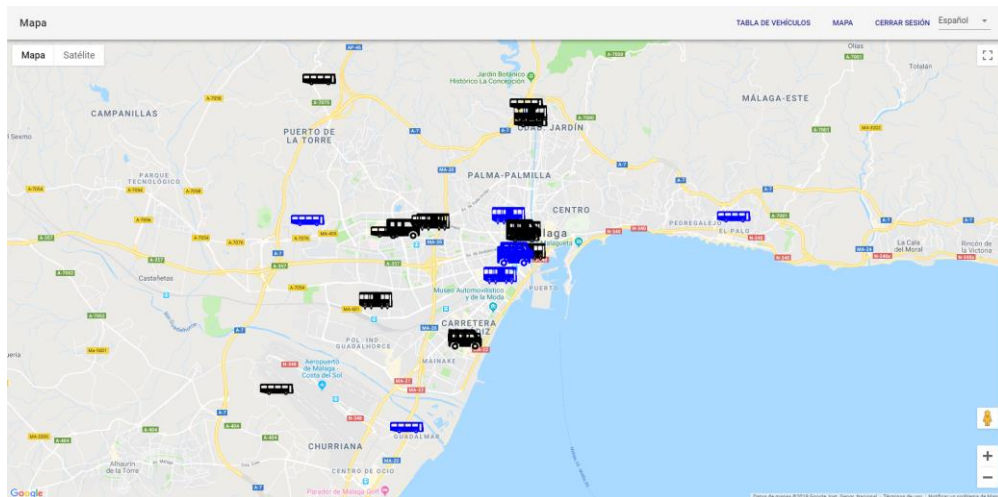


Figura 43. Mapa con la geolocalización y estado de los vehículos

En la Figura 44 se muestra el Mapa con la ventana de información de un vehículo activa.

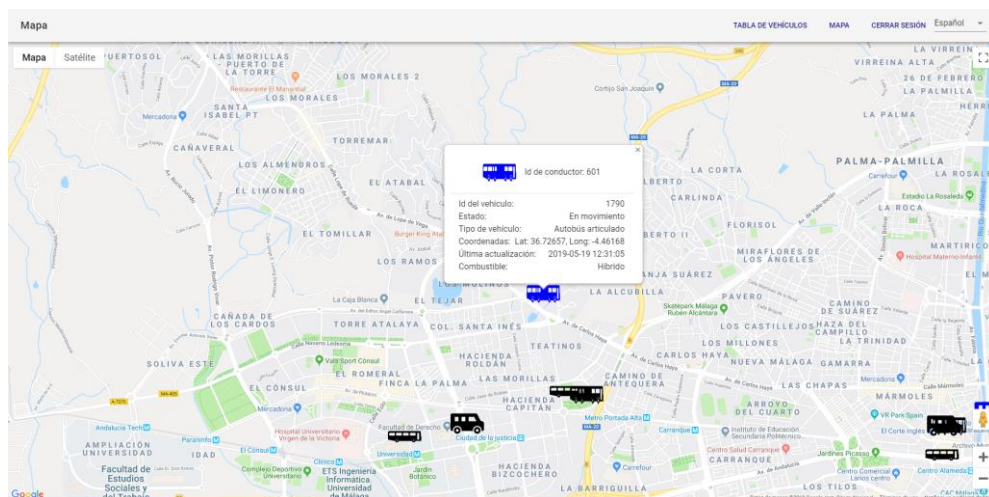


Figura 44. Mapa con ventana de información de un vehículo activa

3.3. Presupuesto

Se tienen en cuenta distintos conceptos para la elaboración del presupuesto como son los materiales, el software y el coste de desarrollo de la Herramienta Web. El coste de la adaptación y mantenimiento de la red LTE-M se entiende que será la empresa propietaria de la red la que se encargue de los gastos y, por tanto, cobrará una cuota mensual a cambio.

3.3.1. Recursos materiales

Son los costes asociados al material de trabajo del personal.

Concepto	Precio	Tiempo de uso	Total
Ordenador profesional (Licencia incluida)	1500 €	5 meses	1500 €
Office 365	7 €/mes	5 meses	35 €
Impresora	50		50 €
Libelium NB-IoT / Cat-M Development kit ⁹	1500 €	5 meses	1500 €
Tarjeta SIM	20 €/mes	5 meses	100 €
			3185 €

Tabla 7. Costes Recursos materiales

3.3.2. Software de desarrollo

Son los costes asociados al software de desarrollo utilizado, es decir, las tecnologías y herramienta para desarrollar la Herramienta Web.

Concepto	Precio	Total
IDE/Editor (VSCode)	0 €	0 €
VueJS	0 €	0 €
Node.js	0 €	0 €
Google Maps ¹⁰	0 €	0 €
		0 €

Tabla 8. Costes Software de desarrollo

3.3.3. Desarrollo de la Herramienta web

Son los costes asociados al desarrollo (requisitos y configuración, diseño e implementación y memoria).

⁹ Libelium NB-IoT / Cat-M Development kit <https://www.the-iot-marketplace.com/libelium-nb-iot-cat-m-development-kit>

¹⁰ Siempre que las consultas Dynamic Maps no sobrepasen las 28000 cargas al mes (200 €/mes gratis) <https://cloud.google.com/maps-platform/pricing/sheet/?hl=es>.

	Precio	Tiempo de trabajo	Total
Requisitos	100 €/día	2 días	200 €
Configuración	100 €/día	7 días	700 €
Diseño	100 €/día	7 días	700 €
Implementación	100 €/día	22 días	2200 €
Verificación	100 €/día	3 días	300 €
Memoria	100 €/día	18 días	1800 €
			5900 €

Tabla 9. Coste Desarrollo de la Herramienta Web

3.3.4. Coste total de desarrollo

Concepto	Precio
Recursos materiales	3185 €
Software	0 €
Desarrollo	5900 €
Total	9085 €
Total + IVA	10992.85 €

Tabla 10. Coste total

3.3.5. Costes en producción

Además, existen unos costes en producción (el coste asociado en este ejemplo a la producción se tiene en cuenta respecto a una flota de 25 vehículos). Además, existen costes fijos y costes variables.

3.3.5.1. Costes fijos

Concepto	Cantidad	Precio	Total
Sensores GPS	25	80 €	2000 €
Módulo LTE-M (SIM incluida)	25	20 €	500 €
			2500 €
			Total + IVA
			3025 €

Tabla 11. Costes fijos de en producción

3.3.5.2. Costes variables

	Cantidad	Precio	Total (IVA incluido)
Cloud (Servidor web + base de datos)	1	15 €/mes	15 €/mes
Tarifa de red LTE-M	25	20 €/mes	500 €/mes
			515 €/mes

Tabla 12. Costes variables en producción

3.3.6. Conclusión

Teniendo en cuenta el coste total de desarrollo de 10992.85 € se puede entender que es coste perfectamente viable.

En cuanto a los costes de producción también son viables ya que los costes de los servidores tienen una tarifa mensual y el mantenimiento y adaptación de la red LTE-M corre a cuenta de la empresa propietaria de la red a cambio de una tarifa mensual y se entiende que a medida que surja mayor competencia los precios irán abaratándose, inclusive se podría negociar con la empresa de la red una mejor tarifa a mayor número de vehículos conectados.

4. Verificación

Para la verificación se han llevado a cabo varias pruebas para indicar el correcto funcionamiento de la herramienta web.

4.1. Prueba 1. Inicio de sesión incorrecto

Como se observa en la Figura 45 falta un campo que es requerido por lo que no se activa el botón de inicio de sesión. En la Figura 46 se ha introducido usuario y contraseña, pero sin embargo son incorrectos por lo que no inicia la sesión y por tanto no entra en la herramienta web.

The screenshot shows a login form titled "UOC: Trabajo fin de Máster" and "Herramienta web para la monitorización de vehículos". It includes a language dropdown menu set to "Español", a "Usuario" field with "abc", and a "Contraseña" field. A red error message "El campo es obligatorio" is displayed below the password field. The "INICIAR SESIÓN" button is disabled (greyed out).

Figura 45. Falta un campo requerido

The screenshot shows the same login form as Figure 45. The "Contraseña" field now contains "..." and a red error message "El usuario o la contraseña son incorrectas." is displayed in a box below the fields. The "INICIAR SESIÓN" button is now active (dark blue).

Figura 46. El usuario o la contraseña son incorrectas

4.2. Prueba 2. Límite de número de vehículos consultados a la API REST

A través de una variable que se utiliza en la llamada a la API REST se le ha indicado el número de vehículos que debe traer:

```
{  
  offset = 0,  
  limit = API_ITEM_LIMIT  
}
```

- a) `export const API_ITEM_LIMIT = 25`
- b) `export const API_ITEM_LIMIT = Number.MAX_SAFE_INTEGER`

Con la opción a) se indica que se quiere pedir como máximo 25 vehículo a la base de datos,

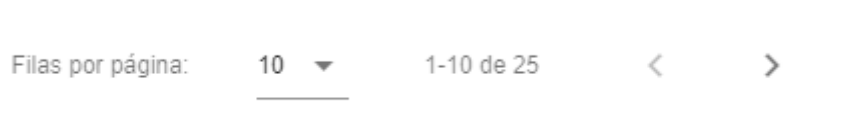


Figura 47. Llamada a la API con un máximo de 25 vehículos

en cambio, con la opción b) se indica que se quieren todos lo que existan en la base de datos.

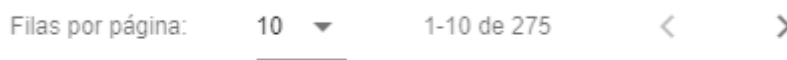


Figura 48. Llamada a la API para consultar todos los vehículos

Se puede verificar en la tabla de vehículos donde aparece el total de vehículos. 25 vehículos para el caso a) y 275 vehículos para el caso b).

4.3. Prueba 3. Movimiento de un vehículo al cabo de 1 minuto

En este apartado se aportan las Figura 49 y Figura 50 para demostrar cómo han cambiado las coordenadas de los vehículos al pasar 1 minuto. Se puede observar con claridad en el mapa y también en la ventana de información como las coordenadas y la última actualización cambian tras 1 minuto.

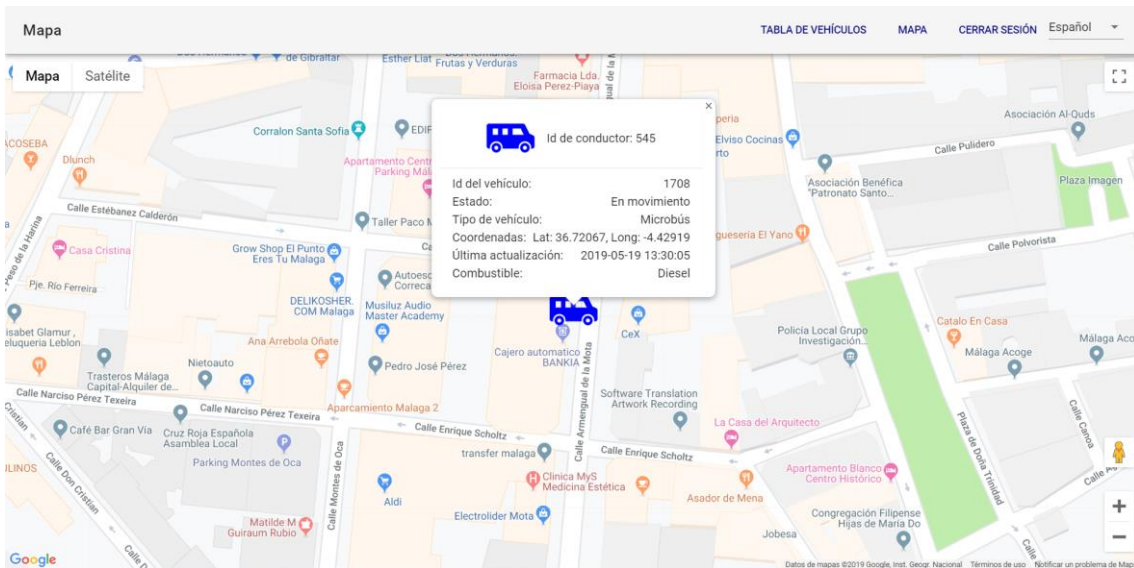


Figura 49. Posición de un vehículo "antes"

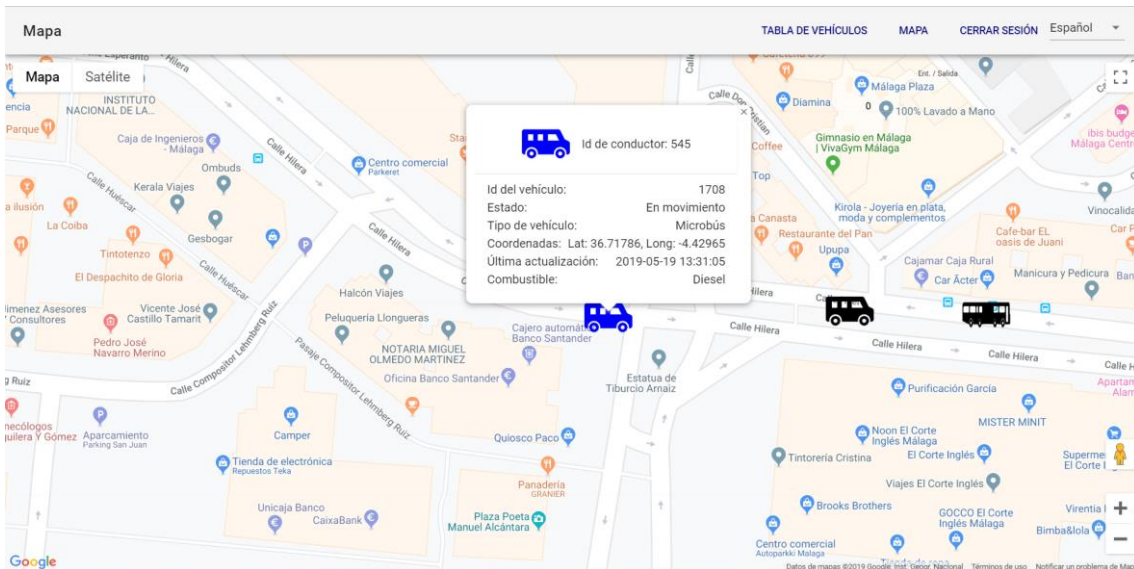


Figura 50. Posición de un vehículo "después de 1 minuto"

4.4. Prueba 4. Cambio de idioma

En las Figura 51 y Figura 52 se muestra la herramienta web en español y en inglés respectivamente y se observa claramente como ha cambiado el idioma de la herramienta web al seleccionar uno u otro idioma. Para ello solo es necesario seleccionar el idioma en la parte superior derecha de la herramienta web en las pantallas de Tabla de vehículos o Mapa. En la pantalla de inicio de sesión el selector de idioma se sitúa en la parte central de la pantalla.

Tabla de vehículos TABLA DE VEHÍCULOS MAPA CERRAR SESIÓN Español

Listado de vehículos Buscar en la tabla

Coordenadas	Id ↑	Tipo de vehículo	Id del conductor	Estado	Sentido	Parada inicial	Combustible	Última actualización	Código de línea	T. últ. trabajo (mi)
Lat: 36.73835, Long: -4.42143	1612	Microbús	695	En movimiento	1	151	Diesel	2019-05-19 13:36:05	1	
Lat: 36.71788, Long: -4.42879	1618	Microbús	Sin conductor	Parado	2	1463	Diesel	2019-05-11 22:07:05	31	
Lat: 36.75724, Long: -4.49545	1627	Autobús de dos plantas	Sin conductor	Parado	1	2115	Diesel	2019-05-17 23:36:05	21	
Lat: 36.71652, Long: -4.44051	1671	Autobús	540	En movimiento	1	473	Diesel	2019-05-19 13:36:05	11	
Lat: 36.74829, Long: -4.42635	1688	Autobús articulado	Sin conductor	Parado	2	1713	Diesel	2019-05-18 23:20:05	15	

Filas por página: 5 1-5 de 25

Figura 51. Herramienta web en español

Vehicle table VEHICLE TABLE MAP LOGOUT English

Vehicle list Search in table

Coordinates	Id ↑	Vehicle type	Driver id	Status	Direction	Firt stop code	Fuel	Last update	Line code	Last working time (min)	T. working time (f)
Lat: 36.73835, Long: -4.42143	1612	Microbús	695	In movement	1	151	Diesel	2019-05-19 13:36:05	1	177	
Lat: 36.71788, Long: -4.42879	1618	Microbús	No driver	Stopped	2	1463	Diesel	2019-05-11 22:07:05	31	0	
Lat: 36.75724, Long: -4.49545	1627	Double bus	No driver	Stopped	1	2115	Diesel	2019-05-17 23:36:05	21	0	
Lat: 36.71652, Long: -4.44051	1671	Bus	540	In movement	1	473	Diesel	2019-05-19 13:36:05	11	242	
Lat: 36.74829, Long: -4.42635	1688	Article bus	No driver	Stopped	2	1713	Diesel	2019-05-18 23:20:05	15	0	

Filas por página: 5 1-5 de 25

Figura 52. Herramienta web en inglés

4.5. Prueba 5. Diseño responsive

En las Figura 53, Figura 54 y Figura 55 se muestra la simulación de distintos diseños responsivos de la herramienta web. Se observa cómo se adapta al dispositivo en cuestión y como muestra por ejemplo la barra lateral izquierda o las opciones de barra superior cuando esta no está disponible porque la pantalla se estrecha.

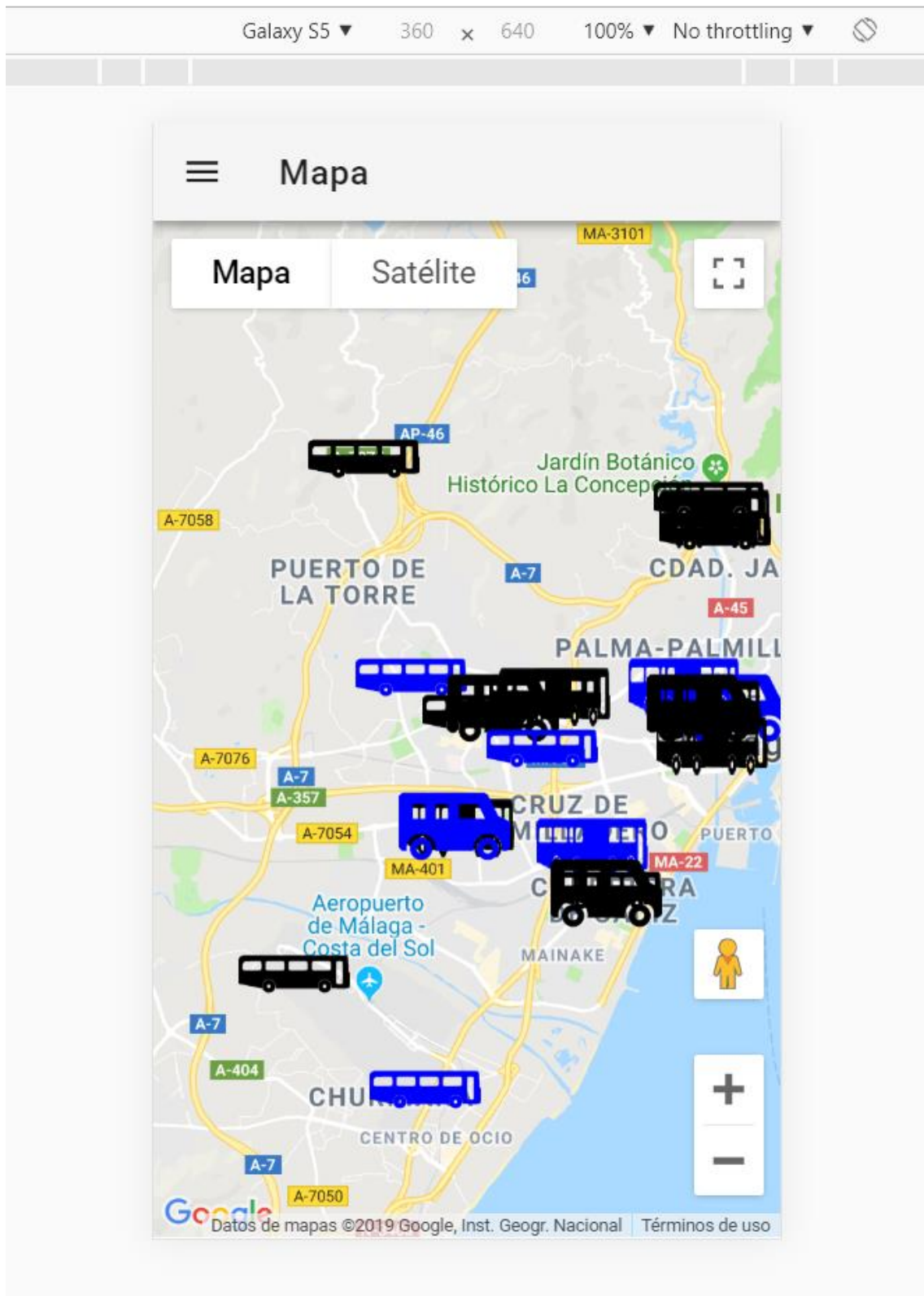


Figura 53. Simulación responsive en un dispositivo smartphone Android

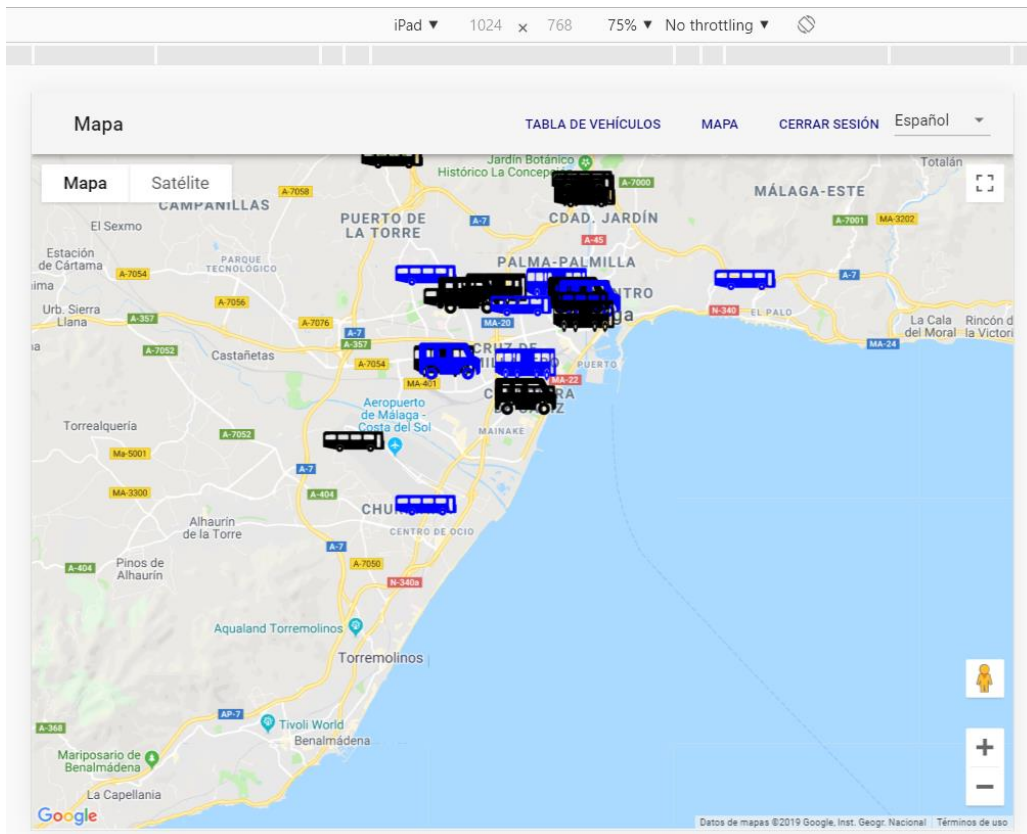


Figura 54. Simulación responsive en un dispositivo Tablet iPad

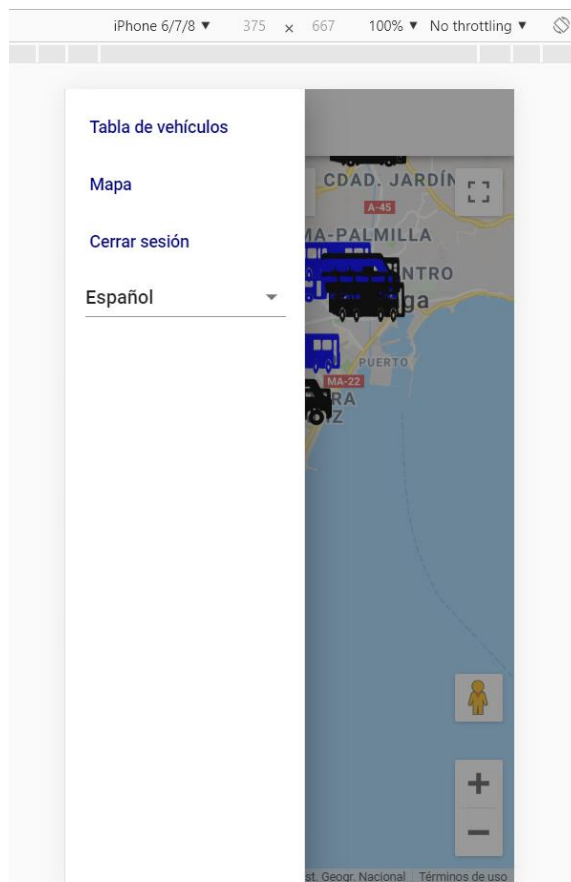


Figura 55. Simulación responsive en un dispositivo smartphone iOS

5. Conclusiones

En este TFM se ha descrito desde una perspectiva IoT el camino que lleva la información hasta que es mostrada. La información es recogida, recorre una red de transporte hasta que es almacenada en una base de datos y finalmente mediante una herramienta web, esta información es transformada y mostrada al usuario/cliente final para su mejor comprensión, es decir, el usuario final es el que tiene que reconocer y entender para qué sirve dicha información y hay que mostrárselo de una forma fácil y adecuada.

Se ha realizado estudio muy completo de las principales tecnologías LPWAN como son Sigfox y LoRa/LoRaWAN en la banda ISM no licenciada y NB-IoT y LTE-M en la banda licenciadas.

Tras el estudio y entendimiento de estas tecnologías no se puede decir que hay una tecnología LPWAN ideal, si no que cada una puede abarcar un espacio dependiendo del área de negocio a la que se quiera dedicar.

Hay que recalcar que Sigfox en el momento actual de realización de este TFM es la tecnología de este tipo más avanzada y parece que tiene el área de negocio más clara. Aunque desde un punto de vista objetivo y a medio plazo las grandes operadoras de la GSMA probablemente abarquen mejor este tipo de tecnologías y en un futuro próximo y puedan controlar el mercado.

Se ha realizado un estudio de las tecnologías de acceso y almacenamiento de la información, dónde se guarda la información y qué tecnologías son adecuadas para diseñar e implementar la herramienta utilizando tecnologías web del lado del cliente.

Por tanto, los objetivos planteados al principio del proyecto se han cumplido ya que la herramienta web es funcional, se han utilizado las tecnologías descritas y muestra la información de forma correcta cumpliendo con todas sus características, destacando principalmente el uso de las coordenadas para la geolocalización a través de Google Maps.

En cuanto a la planificación ha sufrido leves cambios para añadir más información y mejorar el estudio de las distintas tecnologías estudiadas y aplicadas en este TFM, pero siempre respetando los plazos y la metodología.

En cuanto a las líneas futuras la herramienta web es fácilmente ampliable, se pueden añadir nuevos idiomas y por supuesto más ventanas o pantallas, por ejemplo, aplicando técnicas de filtrado o técnicas de Big Data y mostrando estadísticas y gráficas.

Por otro lado, hay que comentar que pronto llegará la quinta generación de tecnologías inalámbricas de telefonía móvil (5G) y posiblemente será el momento en el que realmente todo empiece a estar conectado.

6. Glosario

3GPP	3rd Generation Partnership Project (Proyecto Asociación de Tercera Generación)
AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface (Interfaz de Programación de Aplicaciones)
CSS	Cascading Style Sheets (Hoja de Estilos en Cascada)
DBPSK	Differential Binary Phase Shift Keying (Modulación por Desplazamiento de Fase Binaria Diferencial)
E2E	End to End (Extremo a Extremo)
EPC	Evolved Packet Core
EPS	Evolved Packet System (Evolución del Core de Paquetes)
ETSI	European Telecommunications Standards Institute (Instituto Europeo de Normas de Telecomunicaciones)
ETSI	European Telecommunications Standards Institute (Instituto Europeo de Normas de Telecomunicaciones)
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
GFSK	Gaussian Frequency Shift Keying (Modulación por Desplazamiento de Frecuencia Gaussiana)
GNSS	Global Navigation Satellite System (Sistema Global de Navegación por Satélite)
GPS	Global Position System (Sistema de Posicionamiento Global)
GSM	Global System for Mobile communications (Sistema global para las comunicaciones móviles)
GSMA	Global System for Mobile communications Association (Asociación GSM)
HTML	HyperText Markup Language (Lenguaje de Marcas de Hipertexto)
HTTP	Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto)
IMS	IP Multimedia Subsystem (Subsistema I Multimedia)
IoT	Internet of Things (Internet de las Cosas)
JSON	JavaScript Object Notation (Notación de Objeto de JavaScript)
LPWAN	Low Power Wide Area Network (Red de Baja Potencia y Área Amplia)
LTE	Long Term Evolution
LTE-M	LTE Machine Type Communication
M2M	Machine-to-Machine (Máquina a Máquina)
MCL	Maximum Coupling Loss (Pérdida de acoplamiento máxima)
MVC	Model-View-Controller (Modelo-Vista-Controlador)
MVVM	Model-View-ViewModel (Modelo-Vista-Modelo de Vista)
NB-IoT	Narrowband-IoT
OFDM	Orthogonal Frequency Division Multiplexing (Multiplexación por División de Frecuencias Ortogonales)
PDN	Packet Data Network (Red de paquetes de datos)
REST	REpresentational State Transfer (Transferencia de Estado Representacional)

SC-FDMA	Single Carrier Frequency Division Multiple Access (Acceso Múltiple por División de Frecuencia de Portadora Única)
SF	Spreading Factor (Factor de Esparcimiento)
SPA	Single Page Application (Aplicación Página Única)
TCP/IP	Transmission Control Protocol / Internet Protocol
TTN	The Things Network
UE	User Equipment (Equipo de Usuario)
UMTS	Universal Mobile Telecommunications System (Sistema universal de telecomunicaciones móviles)
VoIP	Voice over Internet Protocol (Voz sobre protocolo de internet)
XML	Extensible Markup Language (Lenguaje de Marcado Extensible)

7. Bibliografía

- [1] Gabriel, «¿Cómo funcionan los dispositivos GPS? Trilateración vs Triangulación,» [En línea]. Available: <https://acolita.com/como-funcionan-los-dispositivos-gps-trilateracion-vs-triangulacion/>. [Último acceso: Abril 2019].
- [2] Y.2060, UIT-T, «Descripción general de Internet de los objetos,» 2012. [En línea]. Available: https://www.itu.int/rec/dologin_pub.asp?lang=s&id=T-REC-Y.2060-201206-!!!PDF-S&type=items. [Último acceso: Abril 2019].
- [3] S. Johan, R. Rueda, M. Jesús y P. Talavera, «Similitudes y diferencias entre Redes de Sensores Inalámbricas e Internet de las Cosas: Hacia una postura clarificadora,» *Revista Colombiana de Computación*, 2017.
- [4] K. Mekki, E. Bajic, F. Chaxel y F. Meyer, «A comparative study of LPWAN technologies for large-scale IoT deployment,» 2018.
- [5] C. Garrido-Hidalgo, D. Hortelano, L. Roda-Sánchez, T. Olivares y M. C. Ruiz, «Evaluando el Consumo de Redes BLE/LoRaWAN para IoT,» 2017.
- [6] Sigfox, «Sigfox Technical Overview,» Mayo 2017. [En línea]. Available: <https://www.disk91.com/wp-content/uploads/2017/05/4967675830228422064.pdf>. [Último acceso: Abril 2019].
- [7] Sigfox, «Sigfox Coverage,» [En línea]. Available: <https://www.sigfox.com/en/coverage>. [Último acceso: Abril 2019].
- [8] redestelecom, «La red de Sigfox ya cubre a 1.000 millones de personas en 60 países,» Abril 2019. [En línea]. Available: <http://www.redestelecom.es/internet/noticias/1111225001903/red-de-sigfox-ya-cubre-1000-millones-de-personas-60-paises.1.html>. [Último acceso: Abril 2019].
- [9] Sigfox, «Sigfox Spain,» [En línea]. Available: <https://partners.sigfox.com/companies/sigfox-spain>. [Último acceso: Abril 2019].
- [10] Securitas Direct, «Securitas Direct amplía su red de telecomunicaciones ATN y ya da cobertura a más del 95% de la población,» Abril 2016. [En línea]. Available: <https://www.securitasdirect.es/es/prensa/notas/securitas-direct-amplia-red-telecomunicaciones-atn>. [Último acceso: Abril 2019].
- [11] Libelium, «<http://www.libelium.com/development/waspmote/documentation/waspmote-sigfox-networking-guide/>,» [En línea]. Available: <http://www.libelium.com/development/waspmote/documentation/waspmote-sigfox-networking-guide/>. [Último acceso: Mayo 2019].
- [12] LoRa-Alliance, «A technical overview of LoRa and LoRaWAN,» Noviembre 2015. [En línea]. Available: https://www.tuv.com/media/corporate/products_1/electronic_components_and_lasers/TUeV_Rheinland_Overview_LoRa_and_LoRaWANtmp.pdf. [Último acceso: Abril 2019].
- [13] M. Saari, A. M. Baharudin, P. Sillberg, S. Hyrynsalmi y W. Yan, «LoRa - A Survey of Recent Research Trends,» *Int. Conv. Inf. Commun. Technol. Electron. Microelectron*, pp. 872-877, 2018.

- [14] Semtech, «SX1272/3/6/7/8: LoRa Modem. Designer's Guide,» 2013. [En línea]. Available: https://www.semtech.com/uploads/documents/LoraDesignGuide_STD.pdf#page=7. [Último acceso: Mayo 2019].
- [15] Redexia, «Redexia Red LoRa,» [En línea]. Available: <https://www.redexia.com/red-lora/>. [Último acceso: Abril 2019].
- [16] Libelium, «Libelium- LoRaWAN Networking guide,» [En línea]. Available: <http://www.libelium.com/development/waspmote/documentation/waspmote-lorawan-networking-guide/>. [Último acceso: Abril 2019].
- [17] LoRa Alliance, «LoRa - Member directory,» [En línea]. Available: <https://lora-alliance.org/member-directory>.
- [18] The Things Network, «Building a global open LoRaWAN network,» [En línea]. Available: <https://www.thethingsnetwork.org/>. [Último acceso: Abril 2019].
- [19] Turismo, Ministerio de Industria Energía y, «Estudio sobre los requisitos técnicos que permitan caracterizar la cobertura con tecnología LTE necesaria para proporcionar determinados servicios de datos,» [En línea]. Available: https://www.academia.edu/35997066/Estudio_de_requisitos_t%C3%A9cnicos_que_permitan_caracterizar_la_cobertura_con_tecnolog%C3%ADa_LTE. [Último acceso: Abril 2019].
- [20] S. Tabbane, «IoT Standards Part II: Part II: 3GPP Standards GPP Standards,» Septiembre 2018. [En línea]. Available: <https://www.itu.int/en/ITU-D/Regional-Presence/AsiaPacific/Documents/Events/2018/IoT-BDG/7.%20IoT%20Standards%20Part%20II%20-%20Sami%20Tabbane.pdf>. [Último acceso: Abril 2019].
- [21] R. Ligeró, «DIFERENCIAS ENTRE NB-IOT Y LTE-M,» [En línea]. Available: <https://accent-systems.com/es/blog/diferencias-nb-iot-lte-m/>. [Último acceso: Abril 2019].
- [22] M. Chen, Y. Miao, Y. Hao y K. Hwang, «Narrow Band Internet of Things,» 2017.
- [23] M. Luthra, R. Atri, M. Sadeghian, S. Malik y P. Rekhi, «Long Term Evolution for IoT (Narrow Band LTE-Cellular IOT),» Octubre 2016. [En línea]. Available: <https://www.slideshare.net/veermalik121/cellular-narrow-band-iot-iot-using-lte-technology?ref=http://www.techplayon.com/narrow-band-lte-narrow-band-iot-nb-iot/>. [Último acceso: Abril 2019].
- [24] G. Araniti, C. Campolo, M. Condoluci, A. Iera y A. Molinaro, «LTE for Vehicular Networking: A Survey,» *IEEE Communications Magazine*, nº 51, pp. 148-157, 2013.
- [25] A. D. Zayas y P. M. Gómez, «3GPP NB-IoT, tecnología y herramientas de medida,» Septiembre 2017. [En línea]. Available: <http://ocs.editorial.upv.es/index.php/JITEL/JITEL2017/paper/viewFile/6577/3183>. [Último acceso: Abril 2019].
- [26] GSMA, «GSMA Deployment Map,» [En línea]. Available: <https://www.gsma.com/iot/deployment-map/>. [Último acceso: Abril 2019].

- [27] Vodafone, «Internet of Things: NarrowBand IoT,» [En línea]. Available: <https://www.vodafone.es/c/empresas/grandes-clientes/es/soluciones/datos/internet-of-things/narrowband-iot/>. [Último acceso: Abril 2019].
- [28] Telefónica, «Telefónica España inicia el despliegue del Proyecto Territorio Rural Inteligente en Castilla y León basado en tecnologías NB-IoT,» [En línea]. Available: <https://www.telefonica.com/es/web/sala-de-prensa-/telefonica-espana-inicia-el-despliegue-del-proyecto-territorio-rural-inteligente-en-castilla-y-leon-basado-en-tecnologias-nb-iot>. [Último acceso: Abril 2019].
- [29] Libelium, «Libelium NB-IoT/LTE-M,» [En línea]. Available: http://www.libelium.com/downloads/documentation/waspmote_nb_iot_cat_m_networking_guide.pdf. [Último acceso: Mayo 2019].
- [30] J. del Valle Arias, «Congreso Nacional del Medio Ambiente Madrid del 26 al 29 de noviembre de 2018,» Madrid, 2018.
- [31] M. Hernandez, «Connectivity Now and Beyond; exploring Cat-M1, NB-IoT, and LPWAN Connections,» Julio 2018. [En línea]. [Último acceso: Abril 2019].
- [32] Wikipedia, «Wikipedia - Base de datos,» [En línea]. Available: https://es.wikipedia.org/wiki/Base_de_datos. [Último acceso: Abril 2019].
- [33] K. Chodorow, MongoDB: The Definitive Guide, O'Reilly Media, 2013.
- [34] BBVAOPEN4U, «API REST: qué es y cuáles son sus ventajas en el desarrollo de proyectos,» [En línea]. Available: <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>. [Último acceso: Abril 2019].
- [35] Wikipedia, «Wikipedia - Servidor web,» [En línea]. Available: https://es.wikipedia.org/wiki/Servidor_web. [Último acceso: Abril 2019].
- [36] J. Eguiluz, «Introducción a AJAX,» [En línea]. Available: <https://uniwebidad.com/libros/ajax>. [Último acceso: Mayo 2019].
- [37] D. Flanagan, JavaScript - The Definitive Guide, Sebastopol, CA: 5th ed., O'Reilly, 2006, p.497.
- [38] formandome, «formandome.es,» Abril 2014. [En línea]. Available: <http://www.formandome.es/javascript/spa-arquitectura-caracteristicas/>. [Último acceso: Mayo 2019].
- [39] L. Andrade, «ReactJS vs. AngularJS vs. VueJS - La pelea de los grandes,» [En línea]. Available: <https://medium.com/drakezair/reactjs-vs-angularjs-vs-vuejs-la-pelea-de-los-grandes-2018-5f4027e61cef>. [Último acceso: Abril 2019].
- [40] Node, [En línea]. Available: <https://nodejs.org/es/>. [Último acceso: 2019].
- [41] Vue CLI, [En línea]. Available: <https://cli.vuejs.org/guide/installation.html>. [Último acceso: 2019].
- [42] Yarn, [En línea]. Available: <https://yarnpkg.com/es-ES/docs/install#windows-stable>. [Último acceso: 2019].
- [43] Google, «Maps JavaScript API,» [En línea]. Available: <https://developers.google.com/maps/documentation/javascript/tutorial>. [Último acceso: Mayo 2019].

- [44] Vuetify, «Vuetify,» [En línea]. Available: <http://vuetifyjs.com/>. [Último acceso: Mayo 2019].
- [45] G. Leclerc, «vue-google-maps,» [En línea]. Available: <https://github.com/xkjyeah/vue-google-maps>. [Último acceso: Mayo 2019].
- [46] K. Kawaguchi, «vue-i18n,» [En línea]. Available: <https://github.com/kazupon/vue-i18n>. [Último acceso: Mayo 2019].
- [47] LoRa Alliance, «LoRa Specification,» 2015. [En línea]. Available: https://lora-alliance.org/sites/default/files/2018-05/2015_-_lorawan_specification_1r0_611_1.pdf#page=34. [Último acceso: Mayo 2019].
- [48] akirasan, «Enviar coordenadas GPS por LoRaWAN TTN,» Octubre 2018. [En línea]. Available: <http://akirasan.net/enviar-coordenadas-gps-por-lorawan-the-things-network/>. [Último acceso: Mayo 2019].
- [49] Techplayon, «Techplayon.com,» [En línea]. Available: <http://www.techplayon.com/maximum-coupling-loss-mcl-and-maximum-path-loss-mpl/>. [Último acceso: Junio 2019].
- [50] G. Threepwood, «Qué es la trilateración satelital,» Octubre 2013. [En línea]. Available: <https://www.youbioit.com/es/article/15626/que-es-la-trilateracion-satelital>. [Último acceso: Abril 2019].
- [51] E. Huerta, A. Mangiaterra y G. Noguera, GPS: Posicionamiento Satelital, UNR EDITORA, 2005.
- [52] LoRa Alliance, «NB-IoT vs LoRa Technology,» Septiembre 2016. [En línea]. Available: <https://lora-alliance.org/>. [Último acceso: Abril 2019].
- [53] Embitel, «Role of Cloud Backend in IoT and Basics of IoT Cloud Applications,» [En línea]. Available: <https://www.embitel.com/blog/embedded-blog/role-of-cloud-backend-in-iot-and-basics-of-iot-cloud-applications>. [Último acceso: Abril 2019].
- [54] Wikipedia, «Wikipedia - HTTP,» [En línea]. Available: https://es.wikipedia.org/wiki/Protocolo_de_transferencia_de_hipertexto. [Último acceso: Abril 2019].
- [55] M. Centenaro, L. Vangelista, A. Zanella y a. M. Zorzi, «Long-range communications in unlicensed bands: The rising stars in the IoT and smart city scenarios,» 2016.
- [56] S. Tabbane, «IoT Standards Part I: IoT Technology and Architecture,» Septiembre 2018. [En línea]. Available: Sami TABBANE. [Último acceso: Abril 2019].
- [57] Bricolabs, «Lora Bricolabs,» [En línea]. Available: https://bricolabs.cc/wiki/guias/lora_ttn#el_spreading_factor_sf_de_lora. [Último acceso: Mayo 2019].
- [58] VueJS, «VueJS,» [En línea]. Available: <https://vuejs.org/>. [Último acceso: Mayo 2019].

8. Anexos

8.1. Ejemplo Sigfox. Regulación europea 1% en la banda ISM

La cantidad de mensajes por día permitidos en la red Sigfox es una aplicación directa de la regulación europea ETSI.

Una hora tiene $60 \cdot 60 = 3600$.

El 1% de 3,600 es 36 segundos, por lo que un dispositivo puede emitir durante 36 segundos por hora.

Un mensaje de Sigfox tarda 6 segundos en enviarse [6], por lo tanto, un dispositivo puede enviar un máximo de $36/6 = 6$ mensajes por hora.

Lo que significa un total de $6 \cdot 24 = 144$ mensajes por día.

Sigfox mantiene 4 mensajes para el uso del protocolo, lo que permite 140 mensajes por día¹¹.

8.2. Ejemplo LoRaWAN. Regulación europea del 1% en la banda ISM

Debido a la multitud de cálculos se recomienda utilizar la calculadora [14]. Se puede descargar desde el link <https://sx1272-lora-calculator.software.informer.com/download/>. La calculadora tiene el aspecto que se muestra en la Figura 56.

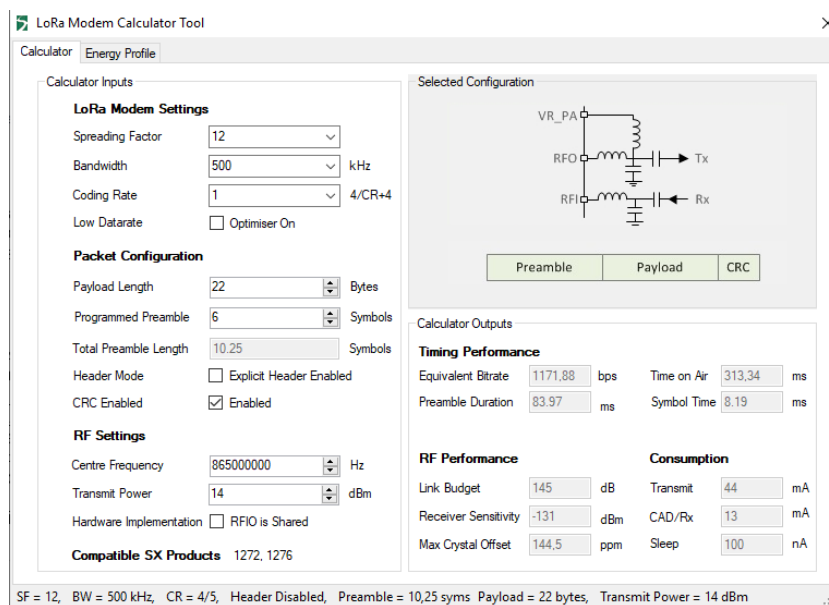


Figura 56. Calculadora LoRa

Según [14] [47] [48] el tamaño mínimo de payload de LoRaWAN es de 13 bytes y para la geolocalización por GPS codificando al máximo el mensaje y con 7 decimales de precisión para la latitud y longitud ocupa 8 bytes, además de 1 bytes para otra información como por ejemplo la identificación.

¹¹ Este cálculo es solo un ejemplo de lo que se hace en la región de EMEA (Europa, Oriente Medio y África). Dependiendo de la ubicación, las limitaciones pueden ser diferentes.

Por lo tanto, se tienen $13 + 8 + 1 = 22$ bytes.
 $22 * 8 \text{ bit/byte} = 176$ bits de tamaño mínimo de payload para enviar la geolocalización de un vehículo.

Una hora tiene $60*60 = 3600$.

El 1% de 3,600 es 36 segundos, por lo que un dispositivo puede emitir durante 36 segundos por hora.

Como se observa en la Figura 56 de la calculadora LoRa (y utilizando valores recomendados en [14] [47] [48], SF12, BW = 500KHz) se obtiene un tiempo de envío de 313.34 ms. Por tanto, $36/0.31434 = 114.64$ mensajes por hora.

Lo que significa un total de $114.64*24 = 2751$ mensajes al día.

8.3.El Spreading Factor (SF) de LoRa

El Spreading Factor (factor de propagación), indica cuanto se “esparce” el mensaje en el tiempo. El SF parte de 7, mínimo esparcimiento, al 12, máximo [14] [47].

A mayor SF mayor radio de cobertura y sensibilidad de recepción pero el mensaje tarda más en enviarse y por tanto se pueden enviar menos mensaje por la regla del 1% [47] [48].

Donde R_b es la tasa de bit o régimen binario, BW es el ancho de banda y SF es el Spreading Factor.

$$R_b = SF * \frac{BW}{2^{SF}}$$

Ecuación 1. Cálculo tasa de bit LoRa [14]

8.4.MCL (Pérdida de acoplamiento máxima)

La pérdida de acoplamiento máxima ha sido elegida por 3GPP como la métrica para evaluar la cobertura de una tecnología de acceso de radio. En teoría, se puede definir como la pérdida máxima en el nivel de potencia conducida que un sistema puede tolerar y aún puede estar en funcionamiento (definido por un nivel de potencia recibido mínimo aceptable). El MCL se puede calcular como la diferencia entre los niveles de potencia conducidos medidos en los puertos de la antena transmisora y receptora como el punto de referencia, la ganancia direccional de la antena no se considera al calcular el MCL.

$$MCL = P_{TX} - (Noise Figure + SINR + Noise Floor)$$

Ecuación 2. Pérdida de acoplamiento máxima [49]

$$Receiver Sensitivity = Noise Figure (NF) + SINR + Noise Floor$$

Ecuación 3. Sensibilidad del receptor [49]

$$Noise Floor = -174 + 10 * \log_{10}(Bandwidth)$$

Ecuación 4. Ruido [49]

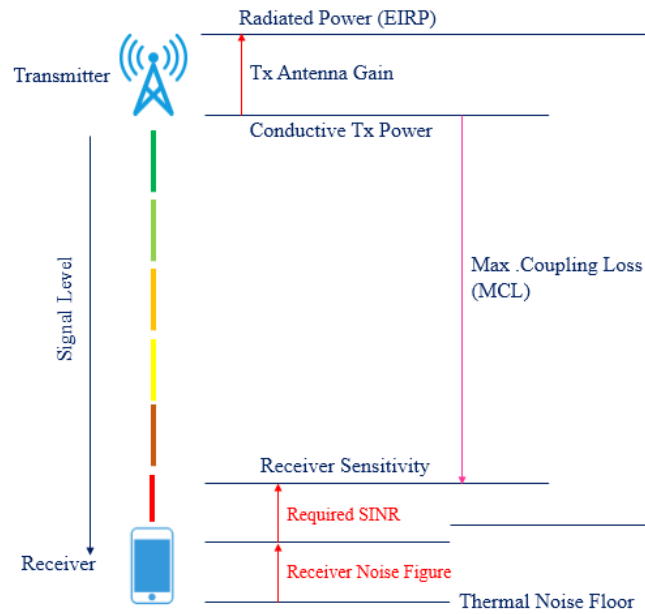


Figura 57. Cálculo MCL [49]

8.5. Registro Google Cloud Platform para usar la API de Google Maps

Para este proyecto es imprescindible el uso de Google Maps para ello es necesario registrarse en Google Cloud Platform y obtener la API Key de Google Maps Platform¹² (Figura 58).

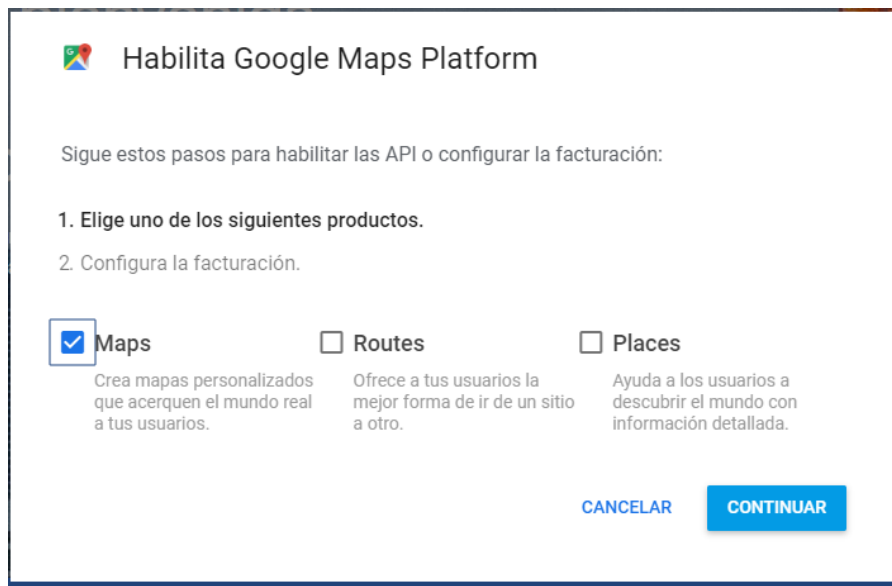


Figura 58. Habilitar Google Maps Platform

Para ello es necesario tener una cuenta de Gmail y registrarse siguiendo los siguientes pasos¹³:

¹² A fecha de este TFM Google ofrece 300\$ a modo de prueba para usar en un máximo de un año.

¹³ A fecha de este TFM.

1. Ir a la consola de Google Cloud Platform <https://cloud.google.com/console/google/maps-apis/overview>.
2. En el menú desplegable Proyecto, crear un proyecto nuevo.
3. En el menú de navegación, seleccione API y servicios → Credenciales.
4. En la página Credenciales, haga clic en Crear credenciales → clave API.
5. En el cuadro de diálogo, haga clic en Restringir clave. A continuación, el cuadro de diálogo creado por la clave API mostrará su nueva clave API creada.
6. (Este paso es opcional) en la página de clave API, en Restricciones de clave, hay que establecer las restricciones de la aplicación.
 - Seleccione referencias HTTP (sitios web).
 - Agregue las referencias (siga las instrucciones).
 - Clic en Guardar.

En la Figura 59 se muestra habilitada la API Key para Google.

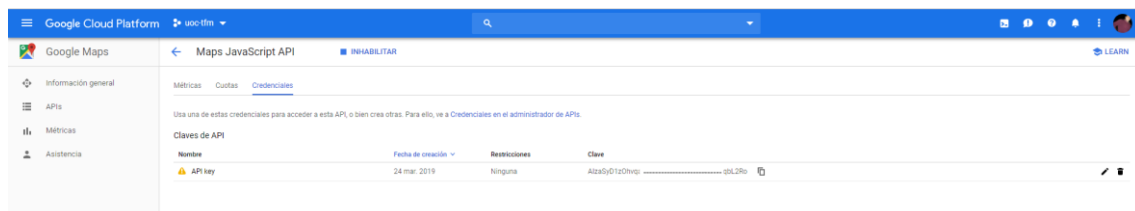


Figura 59. Google Cloud Platform. Credenciales para Google Maps JavaScript API

Las claves de la API de Google Maps consisten en una cadena de texto con un aspecto similar al siguiente¹⁴: AlzaSyD1zOhvqa*****-*****qbL2Ro.

8.6. Instalación de dependencias y otras opciones del proyecto

El repositorio del proyecto de la Herramienta Web se encuentra en Bitbucket en el enlace: <https://bitbucket.org/pacocom/uoc-tfm/src/master/> y está disponible para todo aquel que quiera visualizarlo.

El proyecto se ha desarrollado con la ayuda de Vue CLI, por lo tanto, es necesario añadir la dependencia:

- `npm install -g @vue/cli`

Para instalar las demás dependencias que se encuentran en el archivo package.json solo es necesario ejecutar:

- `yarn install`

Durante el desarrollo es necesario visualizar como está quedando la herramienta web para ello hay que utilizar:

- `yarn run serve`

¹⁴ La API Key se muestra solo parcialmente cambiando algunos caracteres por * para evitar su uso inadecuado.

Además, se puede estilizar el código para que tenga los mismos espacios, tabulaciones, orden de importaciones, ..., utilizando el comando:

- yarn run lint

Para poner el código en producción hay que utilizar el comando:

- yarn run build

El usuario para acceder a la herramienta web es **"tfm"** (sin comillas) y contraseña es **"tfm"** (sin comillas).