



**Universidad de Valladolid**



**ESCUELA DE INGENIERÍAS  
INDUSTRIALES**

**UNIVERSIDAD DE VALLADOLID**

**ESCUELA DE INGENIERIAS INDUSTRIALES**

**Grado en Ingeniería en Electrónica Industrial y Automática**

# **Diseño de Sistemas WIFI en Trenes de Alta Velocidad**

**Autor:**

**Alonso Barbero, Carlos**

**Tutor:**

**Baeyens Lázaro, Enrique  
Departamento de Ingeniería de  
Sistemas y Automática**

**Valladolid, julio 2015.**



Universidad de Valladolid

# Diseño de Sistemas WIFI en Trenes de Alta Velocidad





## AGRADECIMIENTOS

*A mis padres por apoyarme en todo momento y nunca permitir que me rindiera.*



Universidad de Valladolid

# Diseño de Sistemas WIFI en Trenes de Alta Velocidad





## **PALABRAS CLAVE:**

Internet, WIFI, 3G/4G, Router, Trenes, Vehículo, Alta velocidad, AVE, Programación, C#, Visual Studio, Google Maps.

## **RESUMEN:**

En este Trabajo Fin de Grado se realiza el diseño de un sistema que proporcione acceso a internet a los usuarios en un tren de alta velocidad. Se busca conseguir un sistema inalámbrico (WIFI) que permita un gran número de conexiones simultáneas con un ancho de banda suficiente para todas ellas.

En este proyecto se buscará una tecnología capaz de cumplir estos requisitos, realizando un estudio de las tecnologías disponibles actualmente (*WiMAX*, satélite, 3G/4G). Las diferentes tecnologías propuestas tienen la característica de que son capaces de trabajar en situaciones de alta velocidad (310 km/h). Se escogerá un dispositivo que opere con dicha tecnología (3G/4G) y se procederá a su configuración.

La segunda parte de este proyecto consiste en desarrollar una aplicación para realizar pruebas de campo con el dispositivo escogido. Estas pruebas de campo tendrán el objetivo de analizar el rendimiento del dispositivo así como los niveles de cobertura de un trayecto de una línea regular de alta velocidad.



Universidad de Valladolid

# Diseño de Sistemas WIFI en Trenes de Alta Velocidad





## KEYWORDS:

Internet, WIFI, 3G / 4G Router, Trains, Vehicles, High-speed, AVE, Programming, C #, Visual Studio, Google Maps.

## ABSTRACT:

A design of a system that provides Internet access to users in a high-speed train is performed in this End of Degree Project. The target is to set up a wireless (WIFI) system that allows a large number of simultaneous connections and enough bandwidth for all of them.

A research among currently available technologies will be carried out aiming for a solution which meets all these requirements (WiMAX, satellite, 3G/4G). The ones that have been proposed are typified by their suitability to work at high speed (310 km/h). We will select a device that works with this technology (3G / 4G) and proceed to its configuration.

The second part of this project is to develop an application for field testing with the chosen device. These field tests will aim to analyze the device's performance and coverage levels of a regular high-speed route.



Universidad de Valladolid

# Diseño de Sistemas WIFI en Trenes de Alta Velocidad





## ÍNDICE DE CONTENIDO

1	INTRODUCCIÓN .....	19
1.1	JUSTIFICACIÓN DEL PROYECTO .....	22
1.2	OBJETIVOS.....	23
1.3	ESTRUCTURA DE LA MEMORIA.....	24
1.3.1	Introducción.....	24
1.3.2	Estudio detallado de la solución escogida.....	24
1.3.3	Aplicación software para pruebas de campo.....	24
1.3.4	Realización de pruebas de campo.....	25
1.3.5	Estudio económico.....	25
1.3.6	Pliego de condiciones .....	25
1.3.7	Conclusiones y líneas futuras .....	25
1.3.8	Bibliografía.....	26
1.3.9	Anexos.....	26
2	ESTUDIO DETALLADO DE LA SOLUCIÓN ESCOGIDA .....	29
2.1	SOLUCIONES DISPONIBLES.....	29
2.2	TECNOLOGÍA WiMax.....	30
2.2.1	Ventajas y Desventajas.....	31
2.3	INTERNET POR SATÉLITE .....	32
2.3.1	Ventajas y Desventajas.....	33
2.4	TECNOLOGÍA 3G Y 4G .....	34
2.4.1	Tecnología 3G.....	34
2.4.2	Tecnología 4G.....	35
2.4.3	Ventajas y Desventajas.....	36
2.5	DISPOSITIVOS SELECCIONADOS A PRIORI.....	39
2.5.1	Teldat H1-RAIL Router .....	44
2.5.2	Icomera X6 .....	45
2.5.3	NETMODULE NB3700 MOBILE & WLAN .....	46
2.6	DISPOSITIVO ELEGIDO .....	52
2.7	CONEXIONADO HARDWARE.....	53



2.7.1	Redes LAN ( <i>Ethernet</i> ).....	54
2.7.2	Alimentación .....	55
2.7.3	Otras Conexiones.....	56
3	APLICACIÓN SOFTWARE PARA PRUEBAS DE CAMPO. ....	57
3.1	OBJETIVOS DE LA APLICACIÓN SOFTWARE.....	59
3.2	OBTENCIÓN DE DATOS DEL ROUTER NB3700.....	60
3.2.1	COMANDOS Y PARÁMETROS DISPONIBLES .....	60
3.2.2	Puerto serie.....	71
3.2.3	Servidor <i>PHP</i> . ....	72
3.3	ENTORNO Y LENGUAJE DE PROGRAMACIÓN ESCOGIDOS. ....	76
3.4	LIBRERÍAS Y HERRAMIENTAS ESPECIALES NECESARIAS.....	81
3.4.1	Librerías propias.....	81
3.4.2	<i>GMap .NET</i> .....	84
3.4.3	Compresión en formato <i>ZIP</i> .....	87
3.5	ESTRUCTURA GENERAL DEL PROGRAMA.....	88
3.5.1	Estado Actual.....	89
3.5.2	Control de trayecto. ....	91
3.5.3	Opciones.....	91
3.6	ESTADO ACTUAL.....	92
3.7	CONTROL DE TRAYECTO.....	96
3.7.1	Posición actual.....	96
3.7.2	Trayecto en tiempo Real. ....	98
3.7.3	Generador de Caché. ....	104
3.7.4	Visualizar Informes. ....	105
3.7.5	Visualizar Gráficas.....	107
3.7.6	Enviar Datos por Email.....	108
3.8	OPCIONES.....	109
3.9	FICHEROS GENERADOS.....	110
3.9.1	Fichero de datos <i>.CSV</i> .....	112
3.9.2	Fichero de coordenadas <i>.KML</i> .....	113



3.10	ORGANIZACIÓN DE LOS FICHEROS Y MODIFICACIÓN DE LAS VARIABLES. ....	116
4	REALIZACIÓN DE PRUEBAS DE CAMPO.....	118
4.1	OBJETIVO DE LAS PRUEBAS DE CAMPO. ....	120
4.2	PRIMERA PRUEBA. ....	122
4.3	SEGUNDA PRUEBA. ....	123
4.4	Resultados y conclusiones. ....	124
5	ESTUDIO ECONÓMICO .....	129
5.1	COSTES DIRECTOS .....	129
5.1.1	Costes de personal .....	129
5.1.2	Costes de material .....	131
5.2	COSTES INDIRECTOS.....	132
5.3	COSTES TOTALES.....	133
6	CONCLUSIONES Y LINEAS FUTURAS DE TRABAJO.....	135
6.1	CONCLUSIONES .....	135
6.2	LÍNEAS FUTURAS DE TRABAJO .....	137
7	BIBLIOGRAFÍA.....	141
	ANEXO A: MANUAL DE CONFIGURACIÓN DEL SISTEMA.....	147
	ANEXO B: DISEÑO DE LA CAJA SOPORTE PARA EL SISTEMA.....	170



## ÍNDICE DE FIGURAS

Figura 2.1. Funcionamiento del <i>WiMax</i> . .....	30
Figura 2.2 Funcionamiento básico de Internet por satélite. ....	32
Figura 2.3. Ventajas de la tecnología 4G frente a la 3G .....	36
Figura 2.4. Mapa de la cobertura 3G en España 2013 (porcentajes de población).....	42
Figura 2.5. Mapa de la cobertura 4G en España 2013 (porcentajes de población).....	42
Figura 2.6. <i>Router NET MODULE NB3700 Mobile &amp; WLAN</i> . ....	48
Figura 2.7. Situación de los dispositivos en el tren a priori.....	53
Figura 2.8. Conexión <i>Ethernet</i> de los <i>routers</i> . ....	55
Figura 2.9. Montaje particular destinado a la alimentación de nuestro sistema. ....	56
Figura 2.10. Conexión <i>Ethernet</i> entre los 3 dispositivos <i>router</i> . ....	57
3.1. Conexión SSH mediante el software <i>PuTTY</i> .....	60
Figura 3.2. Información generada por el comando <i>status -h wan</i> .....	64
Figura 3.3. Información generada por el comando <i>status -h wwan</i> .....	65
Figura 3.4. Información generada por el comando <i>status -h lan</i> .....	68
Figura 3.5. Información generada por el comando <i>status -h info</i> .....	69
Figura 3.6. Configuración del puerto serie desde nuestro dispositivo .....	71
Figura 3.7. Activación del servidor web para la transmisión de datos .....	72
Figura 3.8. Instrucciones en lenguaje <i>PHP</i> para la petición de datos mediante servidor web.....	73
Figura 3.9. Ejemplo de texto plano devuelto por nuestro servidor web al recibir un comando <i>PHP</i> .....	75
Figura 3.10. Texto plano obtenido de nuestro servidor web.....	78
Figura 3.11. Fragmento de código que nos permite leer del servidor web y seleccionar un dato en particular de todo el texto.....	79
Figura 3.12. Ejemplo de representaciones gráficas con el recurso <i>MSChart</i> . 80	
Figura 3.13. Fragmento del código usado para enviar los resultados por correo electrónico.....	80
Figura 3.14. Lista de todas las librerías usadas en nuestra aplicación <i>software</i> .....	82
3.15. Configuración del objeto de <i>GMapControl</i> .....	85
Figura 3.16. Ejemplo de inicialización del objeto <i>GMapControl</i> aquí llamado " <i>gmap</i> " .....	85
Figura 3.17. Comparación entre cuatro de los mejores servidores de aplicaciones de mapas con los que se puede trabajar en <i>GMap .NET</i> .....	86



Figura 3.18. Diferencia entre la función ruta (rojo) y la función polígono (azul) .....87

Figura 3.19. Fragmento de código para comprimir una carpeta en el formato de compresión *ZIP* ..... 88

Figura 3.20. Diseño del menú principal de nuestra aplicación..... 89

Figura 3.21. Diseño del programa "Estado Actual" ..... 90

Figura 3.22. Diseño del submenú "Control Trayecto"..... 91

Figura 3.23. Diseño del programa "Opciones" ..... 92

Figura 3.24. Diferentes partes del programa "Estado Actual" ..... 93

Figura 3.25. Ejemplo de actualización de comandos *PHP* ante el evento de cambio de *router*..... 94

Figura 3.26. Código asociado al evento de cada botón..... 94

Figura 3.27. Ejemplo de la información mostrada por pantalla del programa "Estado Actual" ..... 95

Figura 3.28. Ejemplo del programa "Posición Actual" en funcionamiento ..... 96

Figura 3.29. Fragmento del código selector de color a partir de la escala de colores definida anteriormente..... 97

Figura 3.30. Ejemplo del mensaje de error en el cuadro de texto ..... 98

Figura 3.31. Desglose de los componentes del programa "Trayecto En Tiempo Real" ..... 99

Figura 3.32. Actualización de las variables para dibujar una trayectoria continua ..... 101

Figura 3.33. Ejemplo de una trayectoria representada en el mapa ..... 101

Figura 3.34. Ejemplo de los datos presentados en el cuadro de texto inferior ..... 102

Figura 3.35. Ejemplo de las cuatro gráficas que aparecen en nuestro programa. .... 103

Figura 3.36. Diseño del programa "Generador de Caché" ..... 104

Figura 3.37. Ejemplo de funcionamiento del programa "Visualizador de Informes" ..... 105

Figura 3.38. Diseño del programa "Enviar Datos por Email" ..... 108

Figura 3.39. Diseño del programa "Opciones" ..... 109

Figura 3.40. Ejemplo de los ficheros generados por nuestra aplicación..... 111

Figura 3.41. Ejemplo de un archivo *CSV* generado en unas pruebas de campo y abierto con una hoja de cálculo ..... 112

Figura 3.42. Ejemplo de la interpretación de un documento *KML* en *Google Earth*..... 114

Figura 3.43. Ejemplo de la estructura de nuestros archivos *KML* ..... 114

Figura 3.44. Archivos auxiliares que usa la aplicación *software*..... 116

Figura 3.45. Contenido del archivo "datos.cfg"..... 117



Figura 4.1. Imagen de la furgoneta especializada usada en las pruebas de campo, en la que se puede apreciar la situación de las difentes antenas...	119
Figura 4.2. Detalle de la ruta realizada en la primera prueba de campo .....	122
Figura 4.3. Ejemplo de la ruta realizada en la segunda prueba de campo ..	123
Figura 4.4 Ejemplo de servicios ofrecidos por tres compañías telefónicas diferentes.....	127
Figura 6.1. Gráfica de la cobertura frente a la velocidad del vehículo.....	136
Figura 6.2. Ejemplo de un archivo de configuración del <i>router</i> .....	138



## ÍNDICE DE TABLAS

Tabla 2.1. Características de los diferentes modelos de del rango <i>NETMODULE NB3700</i> .....	52
Tabla 2.2. Puertos y conexiones disponibles en nuestro dispositivo .....	54
Tabla 3.1. Información generada por el comando <i>status -h gnss</i> .....	61
Tabla 3.2. Información generada por el comando <i>status -h neigh</i> .....	62
Tabla 3.3. Asociación de niveles de cobertura a una escala de colores .....	67
Tabla 3.4. Lista de comandos disponibles en nuestro <i>router</i> . .....	70
Tabla 3.5. Comandos <i>PHP</i> .....	74
Tabla 3.6. Colores usados en el documento <i>KML</i> .....	115
Tabla 5.1. Horas imputadas a cada miembro durante el proyecto .....	130
Tabla 5.2. Coste/hora de cada trabajador del equipo .....	130
Tabla 5.3. Coste total del personal del proyecto .....	131
Tabla 5.4. <i>Hardware</i> para la realización del proyecto .....	131
Tabla 5.5. <i>Software</i> para la realización del proyecto .....	132
Tabla 5.6. Desglose de los costes del proyecto .....	133



Universidad de Valladolid

# Diseño de Sistemas WIFI en Trenes de Alta Velocidad





## 1 INTRODUCCIÓN

En la actualidad internet representa el medio de comunicación más utilizado. La sociedad está acostumbrada a internet como algo cotidiano que forma parte de nuestro día a día. Hoy en día se ha convertido en una necesidad el poder acceder a internet en cualquier lugar. Por este motivo desarrollar un sistema que proporcione internet en los trenes de alta velocidad, y así aumentar las prestaciones y comodidades para sus usuarios es una necesidad importante.

Una de las soluciones más populares para proporcionar internet a un gran número de usuarios es la tecnología WIFI, que permite una gran cantidad de conexiones dentro de una gran superficie. De esta forma la tecnología WIFI es uno de los métodos más efectivos para proporcionar internet a los usuarios de un tren de alta velocidad, ya que en la actualidad existen un gran número de dispositivos de uso cotidiano que dependen de la conexión WIFI (tabletas, móviles, ordenadores portátiles, etc.).

El objetivo de esta nueva tecnología fue designar una marca que permitiese fomentar más fácilmente la tecnología inalámbrica y asegurar la compatibilidad de equipos. De esta forma, en el año 2000, la WECA certificaba la interoperabilidad de equipos según la norma IEEE 802.11b, bajo la marca WIFI. Esto quería decir que el usuario tenía la garantía de que todos los equipos que tuvieran el sello WIFI podían trabajar juntos sin problemas, independientemente del fabricante de cada uno de ellos.

Existen diversos tipos de WIFI, basado cada uno de ellos en un estándar IEEE 802.11 propiamente aprobado y son los siguientes. Los estándares *IEEE 802.11b*, *IEEE 802.11g* e *IEEE 802.11n* disfrutaban de una aceptación internacional debido a que la banda de 2.4 GHz está disponible en casi todo el mundo, con una velocidad de hasta 11 Mbit/s, 54 Mbit/s y 300 Mbit/s, respectivamente.

En la actualidad ya se maneja también el estándar IEEE 802.11a, conocido también como WIFI 5, el cual opera en la banda de 5 GHz y que disfruta de una operatividad con canales relativamente limpio, ya que la banda de 5 GHz ha sido recientemente habilitada y, además, no existen otras tecnologías (Bluetooth, microondas) que la utilicen actualmente, por lo tanto existen muy pocas interferencias. Si bien, cabe destacar que su alcance es



algo menor que el de los estándares que trabajan a 2.4 GHz (aproximadamente un 10 %), debido a que la frecuencia es mayor y a mayores frecuencias, menores serán los alcances.

En cuanto a los dispositivos que están relacionados con el WIFI se puede decir que existen dos grupos o dos tipos de dispositivos WIFI según la función que cumplen:

- Dispositivos de **distribución o de red**: *Routers*, puntos de acceso y repetidores.
- Dispositivos **terminales**: Que en general son las tarjetas receptoras que están integradas en el PC o a cualquier dispositivo móvil, ya sean internas (tarjetas PCI) o bien USB.

## DISPOSITIVOS DE DISTRIBUCIÓN O DE RED

- Los **puntos de acceso** son dispositivos que generan un servicio, que podría definirse como una red WIFI a la que se pueden conectar otros dispositivos. Los puntos de acceso permiten, en resumen, conectar dispositivos en forma inalámbrica a una red existente. Pueden agregarse más puntos de acceso a una red para generar redes de cobertura más amplia, o conectar antenas más grandes que amplifiquen la señal.
- Los **repetidores** inalámbricos son equipos que se utilizan para extender la cobertura de una red inalámbrica, éstos se conectan a una red existente que tiene señal más débil y crean una señal limpia a la que se pueden conectar los equipos dentro de su alcance. Algunos de ellos funcionan también como punto de acceso.
- Los **router** inalámbricos son dispositivos compuestos, especialmente diseñados para redes pequeñas (hogar o pequeña oficina). Estos dispositivos incluyen, un *router* (encargado de interconectar redes, por ejemplo, nuestra red del hogar con Internet), un punto de acceso



(explicado más arriba) y generalmente un switch que permite conectar algunos equipos vía cable (*Ethernet* y USB). Su tarea es tomar la conexión a Internet, y brindar a través de ella acceso a todos los equipos que conectemos, sea por cable o en forma inalámbrica.

### DISPOSITIVOS TERMINALES

Que abarcan tres tipos mayoritarios: tarjetas PCI, tarjetas PCMCIA y tarjetas USB:

- El WIFI puede ser desactivado por un terminal del dispositivo.
- Las tarjetas PCI para WIFI se agregan, o ya vienen instaladas en el propio dispositivo, a los PC. Hoy en día están perdiendo terreno debido a las tarjetas USB. Dentro de este grupo también pueden agregarse las tarjetas MiniPCI que vienen integradas en casi cualquier ordenador portátil disponible hoy en el mercado.
- Las tarjetas PCMCIA son un modelo que se utilizó mucho en los primeros ordenadores portátiles, aunque están cayendo en desuso, debido a la integración de tarjeta inalámbricas internas en estos ordenadores. La mayor parte de estas tarjetas solo son capaces de llegar hasta la tecnología IEEE 802.11b de WIFI, no permitiendo por tanto disfrutar de una velocidad de transmisión demasiado elevada.
- Las tarjetas USB para WIFI son el tipo de tarjeta más común que existe en las tiendas y más sencillo de conectar a un PC, ya sea de sobremesa o portátil, haciendo uso de todas las ventajas que tiene la tecnología USB. Hoy en día puede encontrarse incluso tarjetas USB con el estándar 802.11n, el cual es el último estándar liberado para redes inalámbricas.

Se puede ver como actualmente la tecnología WIFI está presente en la mayoría de los dispositivos de comunicación. Hoy en día se puede observar como poco a poco la mayoría de lugares públicos (bares, hoteles, plazas...)



ofrecen una red WIFI como servicio adicional para sus clientes. En el futuro se puede predecir que existirá un acceso a internet en casi cualquier lugar, esto incluirá los trenes de alta velocidad.

## 1.1 JUSTIFICACIÓN DEL PROYECTO

El presente trabajo de fin de grado radica en un proyecto que surgió a partir de una oferta del Centro Tecnológico *CARTIF* (Parque Tecnológico de Boecillo, Valladolid). *CARTIF* es un centro tecnológico, especializado en dar soluciones integrales a las empresas. El principal objetivo de este centro es el de contribuir al desarrollo de su entorno económico y social a través del uso y fomento de la innovación tecnológica, mediante el desarrollo y la difusión de la investigación.

La idea del proyecto surgió al observar una oportunidad de implementar un sistema WIFI en los trenes de alta velocidad en España. La fuerte demanda de los usuarios de dicho servicio impulso al Centro Tecnológico *CARTIF* junto con una empresa asociada *MAREV ENGINEERING* a desarrollar una alternativa rápida, económica y eficaz capaz de solventar esta necesidad.

Esta demanda viene dada debido a que en la sociedad en que nos encontramos actualmente ya sea por motivos de trabajo o de ocio, la conexión a Internet desde cualquier punto dónde nos encontremos es muy importante. Tan importante que la calidad de los servicios es aumentada potencialmente si dispone en este caso de una red WIFI que proporcione a los usuarios de trenes acceso a Internet de alta velocidad durante el recorrido del mismo.

También hay que destacar que otros países de la Unión Europea ya llevan años con esta tecnología implantada en sus trenes de alta velocidad, por lo que existe cierta urgencia por parte de España de implementar esta tecnología e igualar a estos países en las prestaciones de sus trenes de alta velocidad.

Por tanto la motivación de implementar un sistema WIFI en trenes de alta velocidad es bastante grande, y ha sido lo que ha impulsado este proyecto y por extensión, este trabajo de fin de grado.



## 1.2 OBJETIVOS

El objetivo principal, es desarrollar y validar un sistema que proporcione conexión a Internet sin cables (WIFI) a los usuarios de trenes de alta velocidad durante todo el recorrido. Para validar dicho sistema se desarrollara un *software* de control que interactuará directamente con el sistema desarrollado. Una vez estén configurados los equipos y terminado el *software* se procederá a realizar unas pruebas de campo donde se comprobara el funcionamiento de todo el sistema y las prestaciones que pueda llegar a ofrecer.

El acceso a Internet del sistema consistirá en la tecnología que consideremos más oportuna para el proyecto, y el acceso de los usuarios será a través de de un modem WIFI. El sistema estará basado en equipos comerciales.

Los puntos generales y los objetivos a desarrollar en el proyecto principalmente serán dar servicio de acceso a Internet de forma sencilla a los usuarios de trenes de alta velocidad (AVE), definir una solución cuya finalidad sea facilitar la instalación en cualquier vehículo, detallándola a nivel de equipos comerciales y manuales (*hardware* y *software* de los equipos).

Los puntos fundamentales que busca este proyecto:

- Solución de fácil implementación en cualquier tren de alta velocidad y la posible implementación en otro tipo de vehículos, por ejemplo autobuses metropolitanos.
- Definir los análisis y las pruebas a realizar antes de la instalación de los equipos con el fin de validar la idoneidad y la viabilidad de la solución propuesta.
- Evaluar la compatibilidad con otros servicios de valor añadido. (Ej.: Televisión a la carta, juegos on-line, etc.).
- Desarrollo de un *software* capaz de interactuar con los dispositivos.
- Implementar en el *software* una parte dedicada a la realización de pruebas de campo y su posterior análisis.
- Realización de diversas pruebas para poder establecer los límites de nuestro sistema y mejorar su configuración.



- Conclusiones y líneas futuras.

## 1.3 ESTRUCTURA DE LA MEMORIA

En este apartado se puede encontrar un breve resumen de los diferentes puntos que se incluye en la memoria de este trabajo de fin de grado.

### 1.3.1 Introducción

Se muestra una breve explicación de la necesidad de realizar este trabajo de fin de grado. También se explica las motivaciones y objetivos que se persiguen a lo largo del proyecto que se ha realizado con el centro tecnológico *CARTIF*, haciendo especial hincapié en la importancia de la implementación de un sistema WIFI que proporcione acceso a internet a los usuarios de los trenes de alta velocidad.

### 1.3.2 Estudio detallado de la solución escogida.

En este capítulo de la memoria se hace un estudio de las posibles soluciones que se pueden implementar, teniendo en cuenta las restricciones que tiene instalar este tipo de sistemas en un tren de alta velocidad. Una vez realizado este estudio se escogerá la solución y su dispositivo asociado que mejor se adapte a nuestras necesidades. Por último se hará un completo análisis del dispositivo escogido para poder realizar una configuración completa de este.

### 1.3.3 Aplicación *software* para pruebas de campo.

En este capítulo se realiza una aplicación *software* capaz de interactuar con nuestro dispositivo. Esta aplicación estará orientada a extraer la información proporcionada por nuestro dispositivo y usarla para proporcionar un estudio del rendimiento del dispositivo en unas pruebas de campo específicas.



## 1.3.4 Realización de pruebas de campo.

Una vez desarrollado todo el sistema, se plantearán una serie de pruebas de campo que buscarán poner a prueba el sistema. Dichas pruebas consistirán en una serie de trayectos con una furgoneta diseñada especialmente para prueba de campo. En estos trayectos se evaluarán zonas con diferentes niveles de cobertura para evaluar el comportamiento del sistema en cada una de ellas.

## 1.3.5 Estudio económico

Se realiza un cálculo exacto del coste que posiblemente tendría la realización física del proyecto, teniendo en cuenta tanto los costes materiales, como los costes de personal, así como también se incluyen los costes indirectos, los beneficios, etc.

## 1.3.6 Pliego de condiciones

En este capítulo se muestran las condiciones que debe cumplir el proyecto ligado al trabajo de fin de grado, como plazos de presentación, plazos de elaboración, etc.

## 1.3.7 Conclusiones y líneas futuras

En este capítulo se plantea un resumen del desarrollo del proyecto. Se explican los resultados obtenidos, pasando por los problemas que han ido surgiendo así como las soluciones adaptadas a cada uno de ellos. Por último se plasma posibles mejoras y nuevas implementaciones que se pueden implementar en el proyecto.



## 1.3.8 Bibliografía

En este capítulo se recogen todas las fuentes consultadas para la realización y documentación del proyecto. Estas fuentes se pueden agrupar en libros técnicos específicos, manuales y páginas web relacionadas con el campo que se trabaja en este proyecto.

## 1.3.9 Anexos

Se adjuntarán aquellos elementos que son necesarios para la comprensión de este trabajo de fin de grado. Por motivos de confidencialidad no se incluye el código de la aplicación *software* desarrollada en los anexos.



# DESARROLLO DEL TFG



Universidad de Valladolid

# Diseño de Sistemas WIFI en Trenes de Alta Velocidad





## 2 ESTUDIO DETALLADO DE LA SOLUCIÓN ESCOGIDA

En este capítulo analizaremos las soluciones disponibles en la actualidad, y escogeremos la que mejor se adapte a los requisitos acordados con la empresa interesada (*CARTIF* y *MAREV*). Una vez escogida la solución, la analizaremos para poder desarrollar el sistema y el *software* alrededor de ella.

Este capítulo es un resumen del proyecto de final de carrera de D. Jorge Estrada Robles, con quien realice conjuntamente este proyecto en el Centro Tecnológico *CARTIF*.

### 2.1 SOLUCIONES DISPONIBLES

Para cumplir el objetivo de desarrollar y validar un sistema que proporcione conexión a Internet sin cables (WIFI) a los usuarios de trenes de alta velocidad durante todo el recorrido, se han barajado y se han estudiado inicialmente todas las posibles soluciones existentes en el mercado para obtener tales fines.

En este estudio inicial, para que se cumplan las expectativas de este proyecto que se presenta, se han analizado las diferentes soluciones que el mercado nos ofrece para equipar de conexión WIFI a un tren de alta velocidad.

Las soluciones que el mercado nos ofrece actualmente, se basan principalmente en tres grandes grupos, de los cuales, efectuaremos un análisis y un estudio más o menos determinante, observando las ventajas y los inconvenientes que nos presentan las diferentes tecnologías para finalmente llegar a la solución más idónea. Las tres tecnologías que se han barajado y por lo tanto forma los tres grandes grupos que hemos mencionado anteriormente son: *WiMax*, internet por satélite, *3G/4G*.

## 2.2 TECNOLOGÍA WiMax

En primer lugar, se dará una definición sobre la tecnología *WiMax* y a continuación se profundizará en sus características, para analizar las ventajas y desventajas y comprobar si dicha tecnología es la más idónea para el proyecto.

Durante los últimos años se han creado expectativas y realidades entorno al acceso de banda ancha de última generación. El cableado representa altos costos de instalación que no siempre justifican su tendido hasta áreas rurales o geográficamente inaccesibles.

Llevar servicios ADSL a estas áreas no es económicamente efectivo para los operadores de telefonía. El operador de cable tradicional aún se encuentra en el proceso de transición hacia el transporte de datos, y la tecnología móvil sólo permite tasas de velocidad limitadas.

Estas y otras limitaciones tecnológicas y topográficas, unidas a la rápida expansión del acceso a Internet, han motivado el desarrollo de este nuevo estándar inalámbrico denominado *WiMax*.



Figura 2.1. Funcionamiento del *WiMax*.  
Fuente: <http://www.domodesk.com/>.



## 2.2.1 Ventajas y Desventajas

Entre las **ventajas** más importantes de *WiMax* se encuentran:

- Numerosos servicios: es posible la utilización de otros servicios añadidos como datos, vídeos, *VoIP*, etc.
- Su calidad claramente visible, ya que una sola estación puede dar soporte a cientos de usuarios.
- Alta seguridad: este tipo de internet incluye medidas de seguridad tanto a nivel de usuarios como en la encriptación.
- Gran ancho de banda: la tecnología *WiMax* proporciona un gran ancho de línea llegando a admitir más de 60 conexiones. Además los anchos de banda son totalmente configurables y no están sometidos al espectro.
- Internet rural: esta tecnología es la más utilizada en aquellos lugares en los que resulta muy caro hacer llegar el ADSL o la fibra óptica.

Entre las principales **desventajas** de *WiMax* se encuentran:

- Instalación: el uso de la tecnología *WiMax* requiere la instalación de una pequeña antena exterior adecuadamente instalada por el operador correspondiente, haciendo que el coste de instalación y coste de operación sea elevado.
- Cobertura: sólo es posible utilizar esta tecnología en aquellos lugares en los que un operador proporcione cobertura y tenga las antenas necesarias instaladas. Además, es necesaria una visión directa entre la antena de usuario y la estación base para el estándar *WiMax* que opera en la banda 10-66GHz lo que supone un problema en las ciudades y obliga a emplazar repetidores para evitar zonas de sombra.
- Interferencias: la conexión a la red mediante *WiMax* puede verse afectada por diversos agentes como ondas, interferencias, clima, etc.
- Consumo: este tipo de tecnología consume mucha electricidad.

## 2.3 INTERNET POR SATÉLITE

El Internet por satélite es una solución de acceso a Internet a alta velocidad, moderna y fiable, y que cuenta con anchos de banda superiores a otras tecnologías. No es la más conocida ni la más usada, tampoco la más barata ni la más sencilla, pero es el mejor sistema de acceso a Internet. Es, por lo tanto, la segunda tecnología, después de la tecnología *WiMax* para el acceso rápido a Internet desde lugares remotos, especialmente aquellos que se encuentran en el ámbito rural y en las zonas de alta montaña.

El acceso a internet vía satélite está diseñado para satisfacer las necesidades de acceso a Internet de usuarios que busquen disponer de una conexión a Internet en banda ancha y que se encuentren fuera de la cobertura de la red.

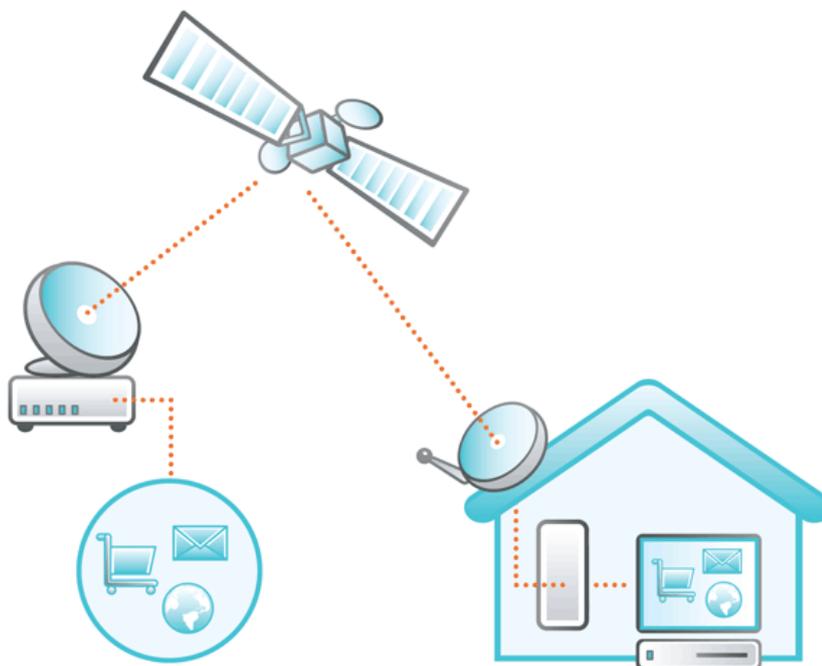


Figura 2.2 Funcionamiento básico de Internet por satélite.  
Fuente: <http://bucket.cdndtl.co.uk/>.



## 2.3.1 Ventajas y Desventajas

Al igual que la tecnología anterior, el Internet por satélite tiene ventajas y desventajas, las cuales, a continuación mencionaremos:

Entre las principales **ventajas** podemos encontrar:

- Situación y movilidad: Puede recibir señales de Internet de banda ancha desde cualquier punto de la Tierra. Es una tecnología ideal para aquellas personas que trabajan o viven en zonas remotas o rurales, dónde la cobertura de las redes telefónicas no alcanza. Es muy útil y además es muy práctico para mantener la conexión a Internet cuando nos desplazamos por cualquier medio de transporte, ya sea vehículo, tren o barco.
- Accesibilidad: Otra de las principales ventajas, es que este tipo de tecnología es muy beneficiosa para las empresas, debido a que proporciona conexiones ininterrumpidas. Las organizaciones, los negocios y las personas pueden mantener el contacto entre sí aún en situaciones cuando los proveedores de Internet estén dañados, desactivados o no tengan cobertura en el entorno donde nos encontramos. Por lo que resumiendo, esta tecnología es apta y es una gran solución para las empresas u organizaciones que tienen oficinas en todo el mundo y desea mantenerlas conectadas.

Entre las principales **desventajas** podemos encontrar:

- Aerodinámica: Quizá esta es una de las principales desventajas del uso de internet por satélite en los trenes de alta velocidad. El problema reside en que un tren de alta velocidad tiene un diseño aerodinámico en que cualquier elemento que sobresalga o altere esta estructura hará que el tren no funcione a su máxima velocidad. Por eso la colocación de una antena de satélite podría alterar la estructura aerodinámica. Cabe destacar que algunos diseños de trenes de alta velocidad ya incluyen un pequeño agujero para la implementación de dichas antenas.
- Conexión lenta: Como mencionaremos posteriormente en las debilidades del Internet por satélite, debido a que la señal debe recorrer grandes distancias hasta contactar con el satélite, se produce un retraso o latencia entre la petición de los datos y la recepción de la respuesta. Este retraso se puede admitir cuando únicamente necesitamos conexiones básicas a Internet, es decir, envío y recepción



de mensajes y correos electrónicos o simplemente navegar por páginas web. Pero en la mayor parte de las ocasiones puede ser lento para aquellas aplicaciones que precisan de un acceso a Internet en tiempo real, como puede ser llamadas por *Skype* (*VoIP*), videoconferencias, videojuegos on-line, etc.

- **Gastos:** Esta tecnología es mucho más cara si la comparamos con otras tecnologías, por denominarlas de alguna manera, terrestres. Realizando un promedio de los datos a los que se tiene acceso, el coste de esta tecnología es entre dos y tres veces más costosa que cualquier otro servicio, desde el año 2010. Sin añadir además, los costes que conlleva la instalación de los equipos necesarios para su funcionamiento.
- **Meteorología:** Por último, el Internet por satélite puede verse afectado por las perturbaciones ocasionadas por la meteorología, tales como la lluvia, la cual puede ocasionar interferencias en las señales y provocar velocidades de carga y de descarga más lentas, así como fallos en la conexión a Internet.

## 2.4 TECNOLOGÍA 3G Y 4G

Son dos tecnologías que están ligadas entre sí, ya que la tecnología 4G es la sucesora de la tecnología 3G y por lo tanto comparten características y algunos cuantos más aspectos en común. Pero para tener más claro lo que son estos dos tipos de tecnologías, veremos por separado las características de cada una de ellas.

### 2.4.1 Tecnología 3G

En primer lugar cabe destacar que la siglas de 3G surge como la abreviación de la tecnología de tercera generación móvil, la cual permite al usuario navegar por Internet a alta velocidad sin la utilización de cables y la transmisión de voz y datos a través del Servicio Universal de Telecomunicaciones Móviles (UMTS).



Los servicios asociados con esta tercera generación proporcionan la posibilidad de transferir tanto voz y datos (Llamada telefónica / video-llamada), cómo únicamente transferir datos sin voz (mensajería instantánea, e-mail, descarga de aplicaciones, etc.).

Por otro lado, aunque esta tecnología inicialmente estaba destinada a su uso en terminales de telefonía móvil, rápidamente las propias operadoras de estos dispositivos comenzaron a proporcionar servicios de conexión a Internet mediante USB con los que disponer de Internet en cualquier lugar sin necesidad de disponer de un teléfono. Por lo tanto, con este servicio de 3G podríamos tener conexión a Internet desde cualquier ordenador, tablet o cualquier otro tipo de dispositivo existente hoy en día.

Además, con este tipo de tecnología es preciso en todos los casos anteriores el uso de una tarjeta *SIM*, aunque el uso del número del teléfono móvil asociado a la tarjeta o recibir llamadas pueda estar bloqueado o asociado a un contrato.

## 2.4.2 Tecnología 4G

Por último la tecnología más actual y la que se encuentra en desarrollo en la actualidad, es la tecnología 4G, denominada así por ser las siglas de tecnología de cuarta generación y simplemente es una evolución de la tecnología 3G.

La tecnología 4G está basada completamente en el protocolo IP, denominada como un sistema de sistemas y una red de redes, que se alcanza gracias a la convergencia entre las redes de cables e inalámbricas. Esta tecnología puede ser usada por módems inalámbricos, telefonía actual (Smartphone) y otros dispositivos móviles.

Esta tecnología denominada también de alta velocidad se apoya en dos tecnologías, *WiMax*, de la cual ya hemos hablado anteriormente y una nueva tecnología denominada *LTE*.

La principal diferencia con las generaciones anteriores es la capacidad que esta nueva tecnología tiene para proveer velocidades de acceso mayores de 100 Mbit/s en movimiento y 1 Gbit/s en reposo, manteniendo una calidad de servicio de punta a punta de alta seguridad que permitirá ofrecer servicios



de cualquier clase en cualquier momento, en cualquier lugar, con el mínimo coste posible.

Mientras que la tecnología 3G actual permite sólo disponer de algunos megabits por segundo desde tu móvil, la tecnología 4G permite una banda ancha que puede alcanzar hasta más de 100 megabits, el equivalente de una conexión de Internet ADSL de una línea fija de un hogar.



Figura 2.3. Ventajas de la tecnología 4G frente a la 3G  
Fuente: <http://ten-tec-ent-vir.blogspot.com.es/>

### 2.4.3 Ventajas y Desventajas

Como tenemos dos tipos de tecnologías que funcionan de manera sustitutiva, analizaremos las ventajas y desventajas de cada una de ellas por separado.

#### VENTAJAS Y DESVENTAJAS (3G)

Entre las **ventajas** de la tecnología 3G, podemos encontrar las siguientes:

- Calidad, fiabilidad y velocidad: Son tres de las grandes ventajas que ofrece la tecnología 3G. Y es que la tecnología 3G nos



proporciona una mejor calidad y fiabilidad y además una mayor velocidad de transmisión de datos, con velocidades de 384 Kbps (Kilo bytes por segundo), lo que supone ser casi siete veces más rápida que una conexión telefónica estándar (56 Kbps) y un además, el ancho de banda es superior, lo que permite realizar conexiones de datos y voz (video llamadas).

- Movilidad: Permite el acceso permanente a Internet desde cualquier lugar.
- Seguridad: Ofrece mayor seguridad en el momento de realizar la conexión, dado que la red es autenticada por el organismo competente en cada lugar y de esta manera el usuario puede asegurarse que se está conectando a una red segura.

Entre las **desventajas** de la tecnología 3G encontramos:

- Cobertura: La cobertura de estos servicios es mucho más limitada que la red utilizada por las tecnologías anteriores (GSM). Es decir, dependiendo de nuestra localización, puede haber zonas donde la tecnología 3G no tenga acceso.
- Velocidad: La velocidad de transferencia de datos varía en función de la cobertura, es decir, a menor cobertura disminuye la intensidad de datos que se pueden transferir. Además, la velocidad de transferencia de datos también se verá afectada si el dispositivo conectado a la tecnología 3G se encuentra en movimiento.
- Costes: El costo de infraestructura de la tecnología 3G es elevado, ya que precisa de la instalación de antenas compatibles con la tecnología 3G.

### VENTAJAS Y DESVENTAJAS (4G)



La tecnología 4G como hemos visto anteriormente es un desarrollo de la tecnología 3G, por lo tanto unifica todas las **ventajas** de la tecnología anterior, pero además podemos ver observar las siguientes ventajas, las cuales siempre van a estar asociadas a la velocidad de la transmisión de los datos:

- Velocidad: La velocidad de transmisión de datos es de entre 5 y 10 veces mayor que la de la tecnología 3G.
- Latencia: El tiempo de respuesta (latencia), es hasta cuatro veces menor y por lo tanto las conexiones o transferencias de datos y de voz en tiempo real son mucho mejores.
- Congestión: Otra ventaja del 4G, no muy conocida, es que disminuye la congestión de las redes. Por lo que más usuarios pueden estar conectados al mismo tiempo en una zona. O lo que es lo mismo, el 4G puede minimizar el tradicional colapso telefónico en grandes eventos.

Como **desventajas** de esta tecnología, podemos resumir, que mantiene las mismas que la tecnología 3G, debido a que la tecnología 4G, es la evolución en cuanto a velocidad se refiere de la generación anterior.



## 2.5 DISPOSITIVOS SELECCIONADOS A PRIORI

Una vez realizado un estudio de las diferentes tecnologías disponibles, se realizarán una serie de comparativas y se determinará cuál se adapta mejor a los requisitos del proyecto. Estos requisitos se basan en las peticiones de las dos empresas participantes (*CARTIF* Y *MAREV*).

La finalidad del presente proyecto es encontrar un sistema que cumpla los siguientes aspectos básicos:

- Implementación rápida y sencilla.
- Proporcionar WIFI a los pasajeros de vehículos o medios de transporte públicos (trenes, autobuses,...) pero principalmente de trenes de alta velocidad (AVE).
- Instalación sencilla que no altere la estructura aerodinámica de un tren de alta velocidad.
- Versatilidad para la posible implementación en otros medios de transportes, como autobuses urbanos.

A continuación analizaremos, con respecto a los requisitos anteriores, las diferentes tecnologías vistas con anterioridad.

- **Tecnología WiMax:** Como hemos visto con anterioridad la tecnología *WiMax* tiene puntos fuertes (ventajas) y puntos débiles (desventajas). Pero para la finalidad que queremos nuestro sistema, los puntos débiles tienen un mayor peso. Inicialmente podemos observar que es una tecnología que nos proporciona cobertura en aquellos lugares donde el acceso a Internet es imposible. También que cubre superficies de grandes dimensiones (varios kilómetros).

Pero lo que hace que esta tecnología no sea la elegida es principalmente el coste de las infraestructuras, es decir, el coste de instalación de antenas o repetidores de la señal de *WiMax* para que la cobertura siempre sea óptima durante el trayecto de los vehículos donde se implanta esta tecnología y la instalación de antenas de recepción de la señal.

Otro de los inconvenientes es que en España, que es el país para el que se está desarrollando el sistema, existen varios operadores tanto locales como regionales, por lo que en cada zona que el usuario utiliza *WiMax* tendría que cambiar de operador, algo que es actualmente



complicado, ya que cada uno utiliza sus propias infraestructuras en cada zona. Además el estándar que utilizan la mayoría de los dispositivos móviles es el WIFI, por lo que para que pueda ser utilizada esta tecnología es necesario que los dispositivos sean compatibles con el *WiMax*.

Y por último, lo que hace que este tipo de tecnología no sea elegida para buscar una solución al problema propuesto viene dado por el punto de vista medioambiental. Y es que la tecnología *WiMax*, tiene un alto consumo eléctrico y por lo tanto energético.

- **Internet por Satélite:** Este tipo de tecnología es una de las soluciones más factibles para el proyecto que se propone. El Internet por satélite nos proporciona grandes anchos de banda y conexiones ininterrumpidas a grandes velocidades y desde el lugar más remoto que nos podamos imaginar.

Estas cualidades de esta tecnología son las que hacen que el Internet por satélite sea una de las soluciones más factibles para el proyecto, pero por contra, no es una tecnología extremadamente sencilla ni barata.

Las conexiones a Internet pueden llegar a ser lentas debido a las grandes distancias que debe de recorrer la señal hasta encontrar el satélite, lo que ocasiona un aumento considerable de la latencia, impidiendo o haciendo más costosa la transmisión de voz.

Además el gran impedimento de esta tecnología es el coste, el cual como hemos visto anteriormente, es de aproximadamente el triple. La meteorología es otro de los factores a tener en cuenta, ya que las interferencias que provocan algunos aspectos meteorológicos pueden ser determinantes.

No obstante, los factores positivos de esta tecnología para dar solución al problema que presenta este proyecto son muy importantes, pero la búsqueda de financiación para estudiar y configurar esta tecnología ante nuestro proyecto (No es una tecnología barata) y debido a que uno de los principales aspectos que se buscaban en el



proyecto no es factible (Instalación rápida y sencilla), nos ha obligado a no seleccionar el Internet por satélite como la tecnología necesaria para dar solución al proyecto.

- **Tecnología 3G/4G:** Este tipo de tecnología es el que hemos seleccionado para poner dar solución al proyecto. Ha sido elegida porque es una tecnología fiable (cuenta con buenos protocolos de seguridad), y además tiene gran calidad y velocidad (7 veces más rápida que una conexión a Internet estándar aproximadamente).

Cuenta con grandes anchos de banda y tiempos de latencia muy pequeños, lo que permite conexiones de voz y datos (video-llamadas) y además nos permite movilidad, es decir, podemos tener conexión a Internet desde cualquier punto en cualquier momento, debido a la existencia de dispositivos, que de diferentes maneras logra la conexión a Internet.

El principal inconveniente que presentaba la tecnología 3G/4G era el coste de las infraestructuras (antenas, repetidores, etc.) pero actualmente ese problema está solventado, ya que los operadores telefónicos en los últimos años han invertido grandes cantidades de dinero en este tipo de infraestructuras. Por lo que actualmente podemos contar con tecnología 3G en la totalidad del país y de tecnología 4G en los principales núcleos urbanos tal y como se muestra en las siguientes figuras:

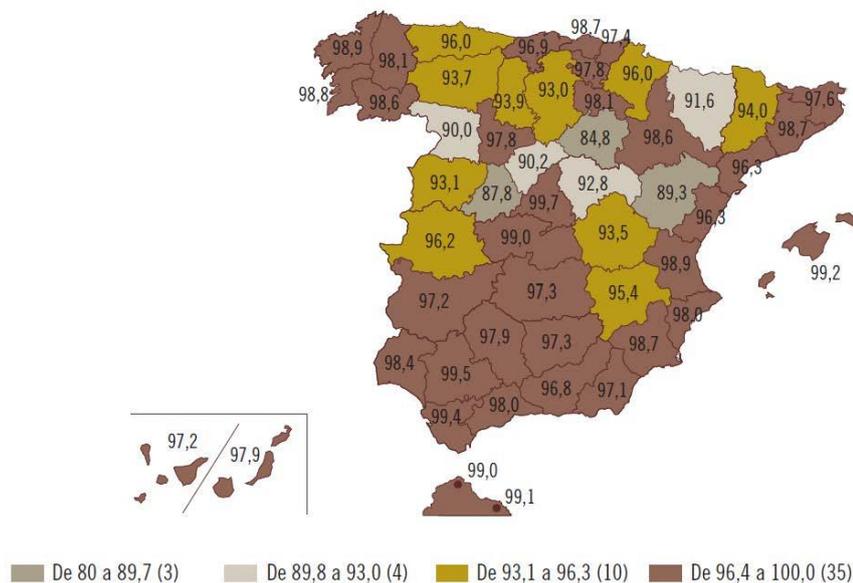


Figura 2.4. Mapa de la cobertura 3G en España 2013 (porcentajes de población).  
Fuente: <http://cnmcblog.es/>

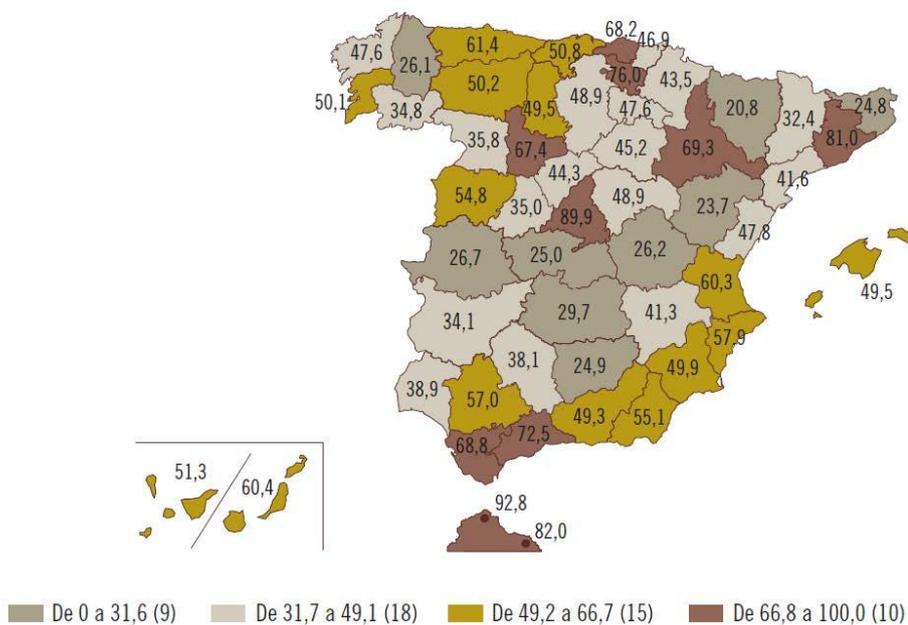


Figura 2.5. Mapa de la cobertura 4G en España 2013 (porcentajes de población).  
Fuente: <http://cnmcblog.es/>



Otro de los inconvenientes que se apreciaban en este tipo de tecnología era que la velocidad variaba en función de la cobertura. Pero como podemos observar, actualmente tenemos cobertura 3G en la mayor parte del país, por lo que este inconveniente tampoco presenta o es un factor de peso suficiente como para declinar esta tecnología y finalmente nos decantamos por la tecnología 3G y 4G (LTE).

Ahora que ya sabemos el tipo de tecnología que vamos a utilizar (Tecnología 3G/4G), procedemos a ver cuáles son los dispositivos comerciales que utilizan esta tecnología y así elegir el más idóneo a nuestros intereses y el cual cumple con las premisas que habíamos previsto en el inicio del proyecto.

Como hemos visto anteriormente la tecnología que hemos seleccionado para buscar solución al presente proyecto es la propia tecnología 3G/4G y por lo tanto una vez seleccionada, procedemos a ver qué dispositivos nos encontramos en el mercado que nos puedan proporcionar WIFI a partir de tecnología 3G/4G.

En efecto, los dispositivos que más se adecúan a las características que buscamos son los *routers* inalámbricos. Estos dispositivos son capaces de proporcionar al menos una red WIFI a través de tarjetas SIM 3G/4G de forma totalmente inalámbrica.

Además, es necesario que los dispositivos cumplan con las normativas específicas y habituales en los trenes, por lo que el dispositivo seleccionado debe de ser un *router* específico o diseñado para ser utilizado en aplicaciones ferroviarias. Por tanto, los candidatos que hemos seleccionado y que hemos considerado óptimos a priori para poder llegar a cumplir el objetivo del proyecto entre otros, son los siguientes:

- *TEL DAT H1-RAIL ROUTER.*
- *ICOMERA X6.*
- *NET MODULE NB3700 MOBILE & WLAN.*



## 2.5.1 Teldat H1-RAIL Router

Es un *router* inalámbrico robusto cuya finalidad es la de ofrecer servicios de banda ancha en trenes. Habilita un ancho de banda seguro para trenes proporcionando servicios multimedia (seguridad, telemetría y acceso a Internet para los pasajeros entre otros).

El *Teldat H1-Rail* es una plataforma de comunicaciones que permite alta disponibilidad, fiabilidad y conectividad celular de banda ancha. En él se combinan un diseño mecánico robusto, adecuado para la instalación en bastidores a bordo de los trenes, con un ancho de banda versátil (WAN inalámbrica para conectar otros dispositivos y WIFI).

En cuanto al *software* del *router*, podemos decir que se impulsa por un *software* de *Internetworking* de *Teldat (CIT)* el cual ofrece una arquitectura robusta de protocolo IP de clase empresarial para la aplicación eficiente de varios servicios VPN administrados por un acceso móvil. El *router* además, se puede gestionar de forma centralizada por la plataforma de gestión de red de *Teldat (TeldaGES)*.

### CARACTERÍSTICAS MÁS IMPORTANTES DEL TELDAT H1-RAIL

A continuación se detalla de forma breve algunas de las características más importantes del *Teldat H1-Rail*.

- Múltiples interfaces para un ancho de banda 3G/4G, con la finalidad de proporcionar una verdadera conectividad de alta velocidad en las aplicaciones a bordo.
- Soporte para múltiples tarjetas SIM, para que en el caso de fallar una de ellas se pueda obtener banda ancha de una alternativa.
- Diseño mecánico y eléctrico lo suficientemente robusto como para su instalación óptima en cabinas de trenes.
- Interfaz 802.11n (WIFI) embebido con modo de operación configurable entre punto de acceso o cliente.



- Dispone de 4 puertos *Ethernet* con la finalidad de ampliar la red *LAN*, la cual dará servicio a otros dispositivos tales como cámaras de seguridad, ordenadores, etc.
- *GPS* con las últimas innovaciones.

### 2.5.2 *Icomera X6*

Al igual que el anterior *router*, el *Icomera X6* es un *router* innovador que permite que los diferentes medios de transporte queden completamente conectados y además es ideal para una amplia gama de aplicaciones en el propio ámbito de los transportes. Incorpora un *software* patentado que combina múltiples redes para proporcionarnos la conexión más rápida y más fiable para que los pasajeros o los usuarios de los medios de transporte (concretamente trenes) queden satisfechos.

#### CARACTERÍSTICAS ICOMERA X6

Las principales características del *Icomera X6* son las siguientes:

- Incorpora 4 módems *LTE* y capacidad *WIFI*, además de un módem adicional o ranura *WIFI* para futuras expansiones.
- Soporte para dos tarjetas *SIM* de tal manera que los operadores puedan reducir costes en zonas o escenarios fronterizos.
- El *Icomera X6* además, incorpora la norma *IP65*, por lo que es un *router* ideal para cualquier ambiente tanto sucio como húmedo.
- Está diseñado para todas las aplicaciones ferroviarias incluyendo soporte para servicios a los pasajeros.



- Cuenta con 4 GB de memoria *RAM* y un procesador a 2.26 GHz, por lo que este *router* permite el uso de grandes aplicaciones que se pueden ejecutar al mismo tiempo desde el mismo.

### 2.5.3 NETMODULE NB3700 MOBILE & WLAN

Los *Routers* para trenes de *NETMODULE* están diseñados para la comunicación Internet utilizando conexión móvil GPRS (2G), UMTS (3G) y *LTE* (4G).

El *NB3700* ofrece conexión por cable a su *Switch Ethernet* de 5 conexiones *Ethernet M12*. Los *Routers* para trenes de *NETMODULE* cumplen con la normativa EN50155 y demás características que son obligatorias para la mayoría de las aplicaciones en trenes. Soportando las últimas tecnologías *WAN/LAN* (incluyendo *GSM*, *UMTS*, *LTE*, *WLAN*), *GPS* y llamadas de voz, estos *Routers* ofrecen una gran conectividad con un envío de datos robusto entre los enlaces de banda ancha usando el protocolo móvil IP.

Concretamente, el *NETMODULE NB3700 Mobile & WLAN* proporciona acceso inalámbrico a Internet para trenes. Este *router* puede ser usado para crear una red *WLAN* para los pasajeros del tren o para cualquier otro equipo electrónico a bordo del propio tren.

El *NB3700* es el módulo de comunicación más idóneo que nos proporciona *NETMODULE* para ser utilizado en trenes y para cualquier otra aplicación en medios de transporte públicos. Y además este equipo es perfecto siempre y cuando las finalidades que le demos al módulo sean las siguientes:

- Control remoto.
- Condiciones de monitoreo.
- Adquisición de datos.
- Sistemas de información para pasajeros y señalización en forma digital.



Gracias a su característica única de subredes basadas en puertos, el NB3700 es capaz de separar diferentes de redes de aplicación, es decir, cada red puede ser utilizada para un fin diferente, evitando así interferencias entre las redes utilizadas y garantizando la comunicación.

El dispositivo además dispone del soporte necesario para permitir priorizar el tráfico, evitando que tareas de menor prioridad (menos importantes) puedan bloquear o retrasar datos o actividades de una mayor prioridad.

Por otro lado, el *router* dispone de una potente "suite" de protocolo VPN (*Virtual Private Network*), el cual permite que el ordenador en la red envíe y reciba datos sobre redes compartidas o públicas como si fuera una red privada con toda la funcionalidad, seguridad y políticas de gestión de una red privada.

Además podemos conectar varios dispositivos directamente al NB3700 usando los cinco puertos *Ethernet* conmutados de los que dispone en su parte frontal.

El NB3700 está equipado con múltiples módulos 3G/4G por lo que el ancho de banda total puede ser extendido según las necesidades de cada aplicación, especialmente en aquellas aplicaciones en las que haya que proporcionar WIFI a pasajeros, pudiendo incrementar el rendimiento para que el usuario tenga un mejor servicio.

En cuanto al *software* del *router*, está basado según el fabricante en componentes correctamente probados, incluyendo un sistema operativo *Linux* embebido y protocolos de comunicación estandarizados.

La configuración del NB3700 de NETMODULE se realiza por medio de un navegador vía web o mediante líneas de comando, pudiendo realizar actualizaciones de *software* o nuevas configuraciones de forma autónoma y remota mediante la elaboración de programas de interfaz específicos para tales fines.



Figura 2.6. Router NET MODULE NB3700 Mobile & WLAN.  
Fuente: <http://www.netmodule.com/>.

Las dos ranuras para insertar tarjetas *SIM* junto con su sofisticado controlador de enlaces *WAN* ofrecen alta disponibilidad de conexiones de red entre múltiples proveedores de la red.

Por último, el *router* está cualificado para operar bajo condiciones medioambientales extremas y está definido bajo la norma EN50155 (Aplicaciones ferroviarias. Equipos electrónicos utilizados sobre material rodante). Soporta temperaturas en un rango de  $-40\text{ }^{\circ}\text{C}$  hasta  $+70\text{ }^{\circ}\text{C}$ , alcanzando picos que soporta  $+85\text{ }^{\circ}\text{C}$  durante un máximo de 15 minutos).

## CARACTERÍSTICAS DEL NB3700

A continuación se van a detallar las características más importantes del *router NB3700*, centrándose en los siguientes puntos:

- *WWAN*.
- *WLAN*.
- *GPS*.
- *ETHERNET SWITCH*.
- *ENTRADAS Y SALIDAS DIGITALES*.
- *USB Y PUERTO SERIE*.
- *ALIMENTACIÓN*.
- *ENTORNO Y MEDIOAMBIENTE*.
- *TIEMPO MEDIO ENTRE FALLOS (MTBF)*.
- *ESTÁNDARES FERROVIARIOS*.



- MODELOS DE LOS DISPOSITIVOS.

## WWAN

Dispone de un módulo que puede operar en los modos *LTE (4G)*, *UMTS (3G)* y *GSM (2G)*, las frecuencias en las que opera son:

- LTE/4G: 2600 MHz, 2100 MHz, 1900 MHz, 1800 MHz, 900 MHz, 850 MHz, 800 MHz
- HSPA+/UMTS/3G: 2100 MHz, 1900 MHz, 900 MHz, 850 MHz
- GSM/2G: 1900 MHz, 1800 MHz, 900 MHz, 850 MHz

Los máximos ratios de transmisión de datos son de 100 Mbps de bajada y 50 Mbps de subida para la tecnología *LTE*, de 42 Mbps de bajada y 5.76 Mbps de subida para la tecnología *HSPA+* y de 21 Mbps de bajada y 5.76 Mbps de subida para las tecnologías *UMTS (3G)* y *GSM (2G)*.

Sigue el estándar IEEE802.11 a/b/g/n de hasta 300 Mbps a 2.4GHz MIMO, pudiéndose usar tanto como "punto de acceso" como "cliente", admitiendo un máximo de usuarios en el modo de punto de acceso de 54 si la seguridad es *WPA* y de 110 usuarios si la seguridad es del tipo *WPA2*.

## GPS

Receptor *GPS* que funciona con el estándar de flujo de datos NMEA 0183 y con una sensibilidad de seguimiento típica de -154 dBm.

## ETHERNET SWITCH

El dispositivo además dispone de 5 puertos *Ethernet switch* 10/100Mbps con un conector del tipo *M12* de 4 polos, con modo auto MDIX, con el cual se elimina la necesidad de utilizar cables específicos para cada conexión ya que permite al receptor detectar la señal que está recibiendo y adecuarse a la misma.



## ENTRADAS Y SALIDAS DIGITALES

El NB3700 de NETMODULE dispone de 2 entradas digitales, que consideran de nivel 0 a los valores entre 0-4 V<sub>DC</sub> y que consideran de nivel 1 a los valores comprendidos entre 7.2-40 V<sub>DC</sub>. Y también dispone de 2 salidas digitales de 0-60 V<sub>DC</sub> y 1 Amperio, consumiendo una potencia máxima en las conmutaciones de 60W.

## USB Y PUERTO SERIE

Dispone también de un puerto USB para realizar actualizaciones y otras acciones sobre el dispositivo que podremos ver más adelante y puerto serie RS-232, el cual este último es opcional, dependiendo del tipo de dispositivo utilizado.

## ALIMENTACIÓN

Admite tensiones nominales de alimentación de 24 V<sub>DC</sub>, 36 V<sub>DC</sub> y 48 V<sub>DC</sub> de acuerdo con la normativa vista anteriormente EN50155 y comprendidas en un rango de 12-60 V<sub>DC</sub> (-15%/+5%), con un consumo máximo de 15W.

## ENTORNO Y MEDIOAMBIENTE

Cumpliendo con la norma EN50155 TX, el *router* está desarrollado para trabajar de forma normal a temperaturas comprendidas entre los -40 °C y los +70 °C (+85 °C durante 15 minutos) y además con un grado de protección IP40, es decir, los elementos sólidos que puedan penetrar en el dispositivo son menores de 1mm y no está protegido frente al ingreso de agua, por lo que el líquido entrará en el equipo en poco tiempo.

## TIEMPO MEDIO ENTRE FALLOS (MTBF)



El tiempo medio estimado entre fallos que nos proporciona el fabricante sobre dispositivos de este tipo es de entre 117'000 y 296'000 horas, en función del modelo que estemos utilizando concretamente.

## ESTÁNDARES FERROVIARIOS

Como ya hemos venido narrando con anterioridad, al ser un *router* destinado a su implementación en medios de transporte, concretamente en trenes, debe de cumplir una serie de normas destinadas a este tipo de entornos especiales. Por tanto, el *router* cumple con la norma que hemos venido describiendo con anterioridad EN50155, la CEN/TS 45545 (protección contra incendios en vehículos ferroviarios), la IEC 60068-2-6;1982 (Vibraciones) y la IEC 60068-2-27;1987 (Choques e impactos).

## MODELOS DE LOS DISPOSITIVOS

Dentro de la gama de *NET MODULE NB3700 Mobile & WLAN*, podemos encontrar 6 modelos distintos del dispositivo, los cuales describimos a continuación:

- NB3700-R.
- NB3700-U: Es un *router UMTS (3G)*.
- NB3700-U-G: Es un *router UMTS (3G)* e incorpora *GPS*.
- NB3700-UW-G: Es un *router* equivalente en funciones al anterior, es decir, consta de *3G* y *GPS* y además ofrece servicio *WLAN (WIFI)* 802.11 bgn operando tanto en el modo de punto de acceso como de cliente.
- NB3700-L-G: Es un *router* que nos ofrece la posibilidad de operar con tecnología *LTE (4G)* e incorpora *GPS*.



- NB3700-LW-G: Ofrece las mismas características que el *router* anterior, ofreciendo además servicio *WLAN* (WIFI) 802.11 bgn, operando tanto en el modo de punto de acceso como de cliente.

Model	UMTS	LTE	WLAN
NB3700-R (Wireline)			
NB3700-W (WLAN)			●
NB3700-U (UMTS)	●		
NB3700-UW (UMTS & WLAN)	●		●
NB3700-2UW (UMTS & WLAN)	2x		●
NB3700-L (LTE)	●	●	
NB3700-LW (LTE & WLAN)	●	●	●
NB3700-2LW (LTE & WLAN)	2x	2x	●

Tabla 2.1. Características de los diferentes modelos de del rango *NETMODULE NB3700*

## 2.6 DISPOSITIVO ELEGIDO

Una vez hemos realizado un estudio sobre las diferentes soluciones, tecnologías y sistemas que hay en el mercado, se ha tomado la decisión de seleccionar los equipos/dispositivos con los cuales se va a desarrollar el proyecto.

En primer lugar, se han seleccionado los equipos o dispositivos que se van a emplear para el desarrollo del proyecto. Desde Centro Tecnológico *CARTIF* y con la colaboración de la empresa *MAREV ENGINEERING* hemos seleccionado el dispositivo que nos proporciona *NETMODULE* y concretamente en su variante de, "*equipo utilizado en vehículos ferroviarios*".

Por tanto, se ha elegido el *router NB3700 Mobile & WLAN* para poder dotar de WIFI a pasajeros de trenes de alta velocidad, buscando una alternativa más económica y simple a la que ya existe en el mercado actualmente.

Concretando en la solución del proyecto, se han elegido tres *routers* de la familia *NB3700* de *NETMODULE*. La explicación de porqué se han elegido tres de estos dispositivos es que la idea principal es la de colocar el *router* "Maestro" en el centro del vagón del tren y los dos restantes, que actuarán como "Esclavos" en los extremos del mismo. (Uno en cada extremo).

Con la finalidad de que los dispositivos sean capaces de manejar las últimas tecnologías *4G*. Se ha decidido que los dispositivos elegidos hayan sido el *NB3700 -LW-G* (Maestro), el cual proporcionará la señal *WIFI* a los pasajeros a través de la señal de la tecnología *4G* (o tecnologías anteriores) y dos módulos *NB3700-L-G* (Esclavos), los cuales nos servirán para aumentar el ancho de banda, es decir, repartir las conexiones de los usuarios de *WIFI* entre los tres dispositivos *router*.

Para hacernos una idea inicial, la colocación de los dispositivos a priori es la siguiente:



Figura 2.7. Situación de los dispositivos en el tren a priori.

## 2.7 CONEXIONADO HARDWARE

Una vez seleccionado y analizado nuestro dispositivo procederemos al encendido de este y a su cableado. Tras un análisis inicial del dispositivo podemos ver en las especificaciones del producto los puertos y conectores que existen en cada uno de los dispositivos junto con todas las especificaciones necesarias.



Label	Panel	Function
SIM 1	Front	SIM 1, it can be assigned dynamically to any modem by configuration.
SIM 2	Front	SIM 2, it can be assigned dynamically to any modem by configuration.
USB	Front	USB 2.0 host port, can be used as USB device server or for software/configuration updates.
Ethernet 1-4	Front	Ethernet switch ports, can be used as LAN or WAN interface.
Ethernet 5	Front	Additional Ethernet switch port (represents an RS232 port for NB3700-4ES)
	Front	Earth protection connector (earthing is optional), connected to the ground of the power supply $V_{GND}$ . If used, connect a yellow-green marked cable with at least $6mm^2$ copper area. Avoid corrosion and protect the screws against loosening
Power	Front	Power supply (galvanically isolated)
Digital I/O	Front	Galvanically isolated digital I/O M12 connector
<b>Mob 1</b>	Front	TNC female connector for mobile antenna 1
<b>WLAN 1</b>	Front	TNC female connector for first WLAN antenna (main)
<b>GPS</b>	Front	TNC female connector for GPS antenna
<b>Mob 2</b>	Front	TNC female connector for mobile antenna 2
<b>WLAN 2</b>	Front	TNC female connector for second WLAN antenna (diversity)
Reset	Front	Reset button, press at least 3 seconds for reboot and at least 5 second for a factory reset. The start of the factory reset is confirmed by all LEDs lighting up for a second. The button can be released then again.

Tabla 2.2. Puertos y conexiones disponibles en nuestro dispositivo  
Fuente: <http://www.netmodule.com/>

## 2.7.1 Redes LAN (Ethernet)

Siguiendo las especificaciones de los dispositivos, cada uno de los módulos NB3700 utilizados está formado por 5 puertos *Ethernet* (M12), mediante los cuales podemos conectarnos a otros dispositivos ya sean *routers* u otros diferentes. En nuestro caso conectaremos los 3 *routers* en

paralelo, es decir, el *router* principal es el encargado de administrar los 2 *routers* secundarios.

Con la siguiente imagen podemos ver de forma visual como sería el conexionado de los dispositivos entre sí:



Figura 2.8. Conexión *Ethernet* de los *routers*.

Como se puede observar en la figura anterior, la conexión es en paralelo, es decir, el puerto *Ethernet* 2 del *router* maestro, está conectado al puerto *Ethernet* 2 de uno de los *router* esclavo (*LAN* 2), al cual en ocasiones llamaremos *router* 2 y el puerto *Ethernet* 3 del *router* maestro, está conectado al puerto *Ethernet* 2 del otro *router* esclavo (*LAN* 3), al cual también llamaremos en ocasiones *router* 3.

Esta conexión se realiza mediante conectores *M12 Ethernet*, macho-macho, de 4 polos de tipo *D-Coded* como los que se pueden ver a continuación:

Además, se puede apreciar que los puertos *Ethernet* 1 están libres y es debido a que es un puerto que está destinado a la configuración del equipo y aunque funciona de forma similar a los demás, se ha dejado libre para simplificar la instalación del sistema.

## 2.7.2 Alimentación

Pensando en una futura instalación y para poder realizar las pruebas de campo posteriores de forma correcta se ha ideado un sistema de alimentación capaz de alimentar los tres dispositivos *router* de forma

simultánea. La alimentación será de 24 V<sub>DC</sub> para cada uno de los *routers*, y la fuente de alimentación será alimentada a 230V en alterna.

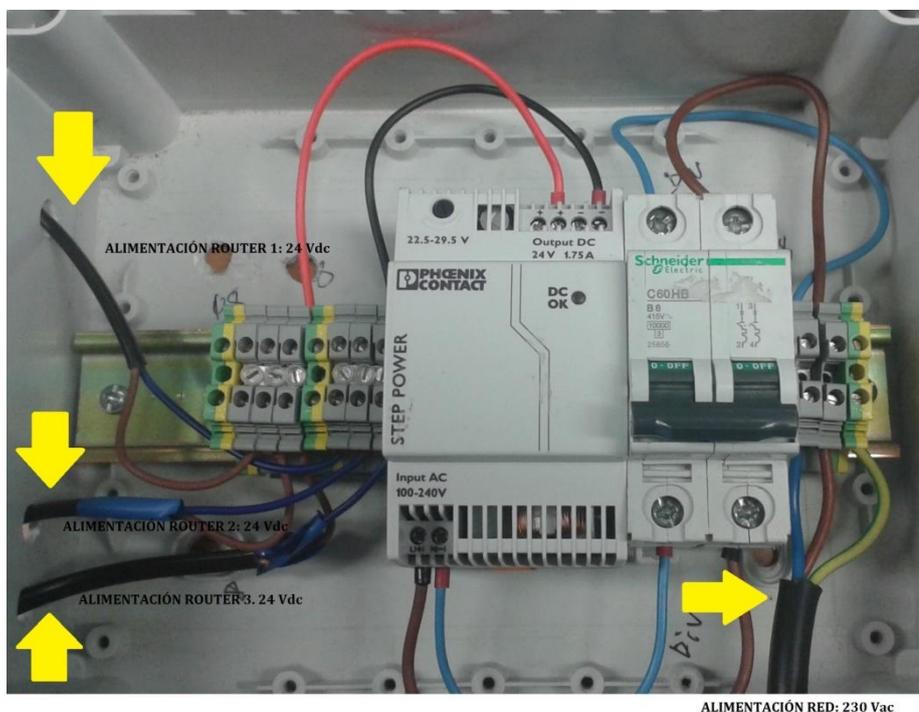


Figura 2.9. Montaje particular destinado a la alimentación de nuestro sistema.

Si hacemos una breve descripción del sistema de alimentación, podemos observar que la entrada de 230 V<sub>AC</sub> está conectada a un interruptor magnetotérmico, a la salida de este conectamos la entrada de la fuente de alimentación, la salida de la fuente de alimentación será de 24 V<sub>DC</sub> y la cual repartimos en tres salidas para alimentar los tres *routers*.

### 2.7.3 Otras Conexiones

Para finalizar la instalación del *hardware*, en cuanto a conexiones físicas se refiere, quedan por destacar las conexiones de los dispositivos necesarios para el correcto funcionamiento del sistema. Estos dispositivos son:

- Antena GPS.

- Antenas WIFI.
- Antenas celulares (Para móviles).
- Adaptador *Ethernet M12* a RJ45 para programar los *routers* desde PC.

Únicamente precisaremos de una antena *GPS*, la cual conectaremos al *router* principal (maestro) mediante un conector *TNC* del que dispone el *router*. Del mismo modo y mediante el mismo tipo de conector (*TNC*) conectamos las antenas WIFI (únicamente en el módulo *NB3700-LW-G*, que es el único que dispone de ellas) y por último se conecta una antena por *router*, ya que cada *router* dispone de un único módem aunque sea capaz de soportar dos tarjetas *SIM*.

El adaptador *Ethernet M12* a RJ45 es necesario para realizar la configuración interna de los dispositivos a través de ordenador. Por ello anteriormente se ha decidido dejar libre el puerto *Ethernet 1* de los tres *routers*. Ya que únicamente se puede programar cada dispositivo mediante ese puerto *Ethernet*. Por tanto, cada vez que se desee modificar la configuración de uno de los *routers* será necesario conectarse mediante uno de esos puertos *Ethernet 1* al PC.



Figura 2.10. Conexión *Ethernet* entre los 3 dispositivos *router*.

### 3 APLICACIÓN SOFTWARE PARA PRUEBAS DE CAMPO.

La aplicación *software* es un pilar fundamental de este TFG, ya que será la encargada de hacer las veces de intermediario entre el usuario y el sistema compuesto por nuestro *routers*.



Desde *CARTIF* se sugiere una aplicación *software* que permita interactuar al usuario de forma gráfica con el sistema. Este requisito nos deja muchas opciones para el lenguaje y el entorno de desarrollo.

El *software* tiene que seguir una serie de objetivos que acotaran las posibles opciones de programación que se usaran. El más decisivo será la compatibilidad con librerías Gmap que nos permitirán la representación de rutas mediante coordenadas *GPS* en mapas de *Google* o *Bing*. Esto hará que el lenguaje escogido para el desarrollo haya sido *C#* con *Visual Studio* como entorno de desarrollo.

A medida que se ha ido desarrollando el proyecto desde *CARTIF* y *MAREV* se han ido sugiriendo una serie de mejoras que se han ido implementando en el proyecto, como puede ser la representación gráfica en tiempo real de los datos que ha ido tomando el sistema.



## 3.1 OBJETIVOS DE LA APLICACIÓN SOFTWARE.

Para desarrollar la aplicación lo primero que nos planteamos fue una serie de objetivos que nos servirán de base para la construcción de nuestro *software*. Estos objetivos fueron fijados por las dos empresas que colaboraron en este proyecto.

A continuación se desarrollaran brevemente los objetivos principales:

- Obtención de todos los datos posibles del sistema. Nuestros tres *routers* nos pueden proporcionar una gran variedad de datos, desde las coordenadas *GPS* en ese mismo instante como las direcciones *MAC* de cada uno de los usuarios conectados a nuestra red *WIFI*. Uno de los objetivos de nuestra aplicación *software* es poder tomar dichos datos y manipularlos para desarrollar diversas tareas, como una ruta en tiempo real plasmada sobre un mapa interactivo.
- Interactuación con librerías *Gmap* para localizar las coordenadas *GPS* en un mapa. De este modo se persigue poder plasmar en un mapa la ruta que se esté realizando en tiempo real.
- Diferenciar por una escala de colores los diferentes niveles de cobertura y poderlos plasmar en la ruta que se esté realizando.
- Interacción con cada uno de los elementos presentes en nuestro sistema, como son los tres *Routers* y los posibles usuarios conectados.
- Otro objetivo fundamental es presentar una aplicación intuitiva y sencilla capaz de ser manejada por usuarios inexpertos.

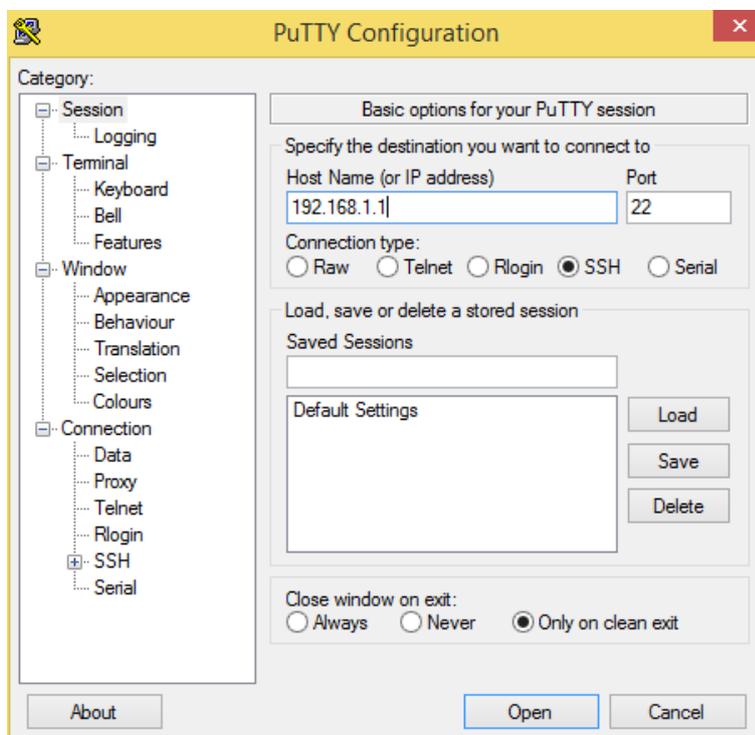
Estos son los objetivos más destacados que debe cumplir nuestra aplicación. A lo largo de este capítulo se mostrara como se han ido resolviendo los diferentes requisitos así como algún requisito que surgió durante el desarrollo del proyecto.

## 3.2 OBTENCIÓN DE DATOS DEL ROUTER NB3700.

El objetivo inicial que debe superar nuestra aplicación es el de obtener datos de un *router*. Para ello el fabricante nos propone una serie de métodos que evaluaremos y finalmente usaremos el que más se adapte a nuestro proyecto.

### 3.2.1 COMANDOS Y PARÁMETROS DISPONIBLES

Mediante una conexión SSH con nuestro *router* y un programa (*PuTTY*), conseguimos obtener los datos internos que genera el *router*. Este proceso es muy sencillo. Hay que conectar el *router* ya configurado al puerto *Ethernet* del ordenador y mediante el *PuTTY* realizamos una conexión SSH. Nos pedirá un usuario y una contraseña que hemos asignado al dispositivo. Una vez realizado esto podremos introducirle una serie de comandos que nos proporciona el fabricante del dispositivo para obtener diferentes datos internos que genera el dispositivo.



3.1. Conexión SSH mediante el software PuTTY

A continuación destacaremos los comandos que más nos interesan, ya que nos proporcionarían todos los datos que usaremos. Cada uno de los comandos nos proporcionaría una serie de datos de los cuales solo escogeremos los que necesitamos. La idea inicial de nuestro programa es generar estos datos y seleccionar y almacenar los que representen interés para analizar con nuestra aplicación.

## GNSS

Este comando nos proporciona una serie de parámetros obtenidos del satélite *GPS*. De todos los datos que obtenemos mediante este comando nos interesan los siguientes:

### GNSS Status

Description	Value
Administrative state:	enabled
Operational state:	up
System:	GPS
Latitude:	49.007482
Longitude:	8.415052
Altitude:	139.70
Satellites in view:	12
Speed:	0.000 m/s horizontal, 0.000 m/s vertical
Dilution of precision:	0.90 horizontal 0.80 vertical 1.30 positional
Satellites used:	6

Tabla 3.1. Información generada por el comando `status -h gnss`

- Con Latitude obtenemos la coordenada correspondiente con la latitud actual proporcionada por un satélite *GPS*. Este dato junto con la longitud nos servirá para obtener la situación actual de nuestro equipo. Con esto podremos representar en tiempo real la trayectoria que siga nuestro tren de alta velocidad. Esto junto a una herramienta de mapa interactivo nos permitirá representar esta ruta en un mapa.



- Con Longitude obtenemos la coordenada correspondiente con la longitud actual proporcionada por un satélite *GPS*. Como se acaba de explicar anteriormente este dato nos servirá para representar una ruta en tiempo real.
- Con Altitude obtenemos la coordenada correspondiente con la altitud actual proporcionada por un satélite *GPS*. Este dato nos permitirá principalmente para realizar estudios entre la relación de la altura con otras variables como pueden ser los niveles de cobertura, el tipo de servicio móvil, etc. Uno de los estudios que se pueden realizar con estos datos es un estudio gráfico comparando el nivel de cobertura con la altura a lo largo del recorrido.
- Con Speed obtenemos una velocidad promedio que se obtiene en función de la variación de la longitud y la latitud en función del tiempo. Al igual que pasa con la altitud este dato es muy interesante de cara a realizar estudios comparando este dato con otros de nuestro interés. Una de los estudios que más nos interesa es la variación de la velocidad con el nivel de cobertura, ya que es uno de los problemas más importantes que se plantean en un tren de alta velocidad ya que estos alcanzan velocidades muy elevadas.

## NEIGH

Con este comando conseguimos una lista detallada de los usuarios conectados a nuestro sistema. Este comando nos será muy útil ya que analizar cómo afecta el número de usuarios conectados a la calidad de nuestro sistema WIFI es un dato fundamental en este proyecto. Gracias a este comando podemos saber diversos datos de los usuarios conectados:

### Neighborhood

Address	MAC	Name	Description
192.168.2.2	3C:97:0E:23:4C:FA	192.168.2.2	ARP discovered host

Tabla 3.2. Información generada por el comando `status -h neigh`



- La dirección IP (Address) nos proporciona la IP fija asignada por el servidor DHCP a cada uno de los dispositivos conectados. Una dirección IP es una etiqueta numérica que identifica, de manera lógica y jerárquica, a una interfaz de un dispositivo dentro de una red que utilice el protocolo IP (*Internet Protocol*). Esta IP será única para cada dispositivo.
- La dirección MAC nos proporciona la MAC propia y única de cada dispositivo conectado. La dirección MAC es un identificador de 48 bits que corresponde de forma única a una tarjeta o dispositivo de red. Se conoce también como dirección física, y es única para cada dispositivo.
- Por último el nombre y la descripción que corresponden al asignado por el usuario en su propio equipo.

Todos estos elementos nos permitirán tener un control exhaustivo de nuestra red en todo momento. Gracias a la dirección IP y la MAC podremos eliminar usuarios conflictivos o restringir a determinados grupos el acceso a algún servicio particular.

## WAN

El comando “wan” nos permitirá ver determinados datos propios de algunas de las redes creadas por nuestro sistema. Estas son variadas pero a nosotros la que nos interesaran serán las redes WIFI. Como ya se puede ver, con este comando podremos tener control de la red WIFI que estará presente en nuestro tren de alta velocidad. De todos los datos que nos presenta este comando nos interesa quedarnos con los siguientes:



## WLAN1

Description	Value
Administrative state	enabled
Operational state	up
Link is up since	2015-05-13 14:32:54
Signal strength	-69 dBm (good)
Network	HomeLan
Station	08:96:D7:90:CB:E2
Bitrate	162 Mbit/s
IP address	192.168.1.104
Gateway	192.168.1.1
Transfer rate down / up	8.09 Kbit/s / 1.11 Kbit/s
Data downloaded / uploaded	104.09 MB / 183.32 MB <input type="button" value="Reset"/>

Figura 3.2. Información generada por el comando `status -h wan`

- Con el dato Operational State tendremos siempre la opción de saber cómo se encuentra nuestra red WIFI en todo momento, encendida, apagada o con algún tipo de error.
- Con el dato Bitrate obtendremos la tasa de bits que se refiere al número de bits que se transmiten por unidad de tiempo a través de un sistema de transmisión digital o entre dos dispositivos digitales. Así pues, es la velocidad de transferencia de datos aproximada.
- Con el dato Data downloaded / uploaded obtendremos la cantidad de paquetes descargados y subidos a la red a través de nuestro sistema WIFI. Este valor se resetea cada vez que se reinicia o apaga el sistema.

## WWAN

Para obtener la información de nuestro sistema 3G/4G utilizaremos el comando `WWAN` que nos proporciona diversos datos acerca de este sistema y las dos tarjetas *SIM* que admite nuestro sistema. Todos estos datos nos servirán para saber el estado de cada red generada por cada tarjeta *SIM* así como una serie de datos muy interesante para nuestras pruebas de campo. A

continuación se muestran los datos que más nos interesaran en este proyecto:

Mobile1	
Description	Value
Status	registeredInHomeNetwork
Modem	Mobile1
SIM	SIM1
Service type	LTE
Network	o2 - de
Signal	-8.5 dB (medium)

Figura 3.3. Información generada por el comando `status -h wwan`

- El parámetro Modem nos indica el modulo físico sobre el que está funcionando nuestra red *SIM*. Este dato es interesante en el caso de poder a optar a dos modem diferentes dentro de un mismo dispositivo.
- El parámetro SIM nos indica la tarjeta *SIM* de la que se están especificando los parámetros. Como ya se comento en el capítulo anterior, todos los dispositivos que utilizamos tienen la opción de trabajar con dos tarjetas *SIM*. Estas tarjetas funcionarán de forma sustitutiva, es decir, por defecto funcionará la *SIM1* y en caso de que esta tenga algún tipo de error o un fallo en la señal entrará a trabajar la *SIM2*. Por lo tanto este dato nos dará de forma indirecta la tarjeta que está funcionando en ese instante, para así poder pasar a formar parte de nuestro estudio.
- El parámetro Service type nos proporciona el tipo de servicio que nos está proporcionando nuestra tarjeta *SIM*. Este dato es un pilar fundamental de nuestro estudio ya que podremos distinguir entre servicios 3G (GSM) y 4G (LTE). Recordemos que es de gran interés para este proyecto el conseguir el mayor ancho de banda posible, por lo que nos interesara movernos en zonas de cobertura 4G.
- El parámetro Network nos permite conocer la red de telefonía que estamos usando: Vodafone, orange, movistar, etc. Este dato nos permitirá en un futuro hacer una comparativa de las diferentes



compañías en un mismo trayecto, analizando su cobertura, su ancho de banda, etc.

- El parámetro Signal nos da el nivel de intensidad de la cobertura en ese momento. Este nivel nos va a servir para demasiadas funciones como el de dibujar un trayecto de diferentes niveles de cobertura asignados a diferentes colores. Llegados a este punto podemos presentar el cuadro de niveles de cobertura asociados a sus respectivos colores. Estos colores están basados en una escala básica de colores donde el rojo representa la menor cobertura y el azul la mayor. Recordemos que la cobertura es una unidad de potencia (W) pero que se suele expresada en dbm. La fórmula para la conversión de estos datos es la siguiente:

$$P[dbm] = 10 * \log_{10}(1000 * P[W])$$

La tabla asociada a los niveles de cobertura es la siguiente:

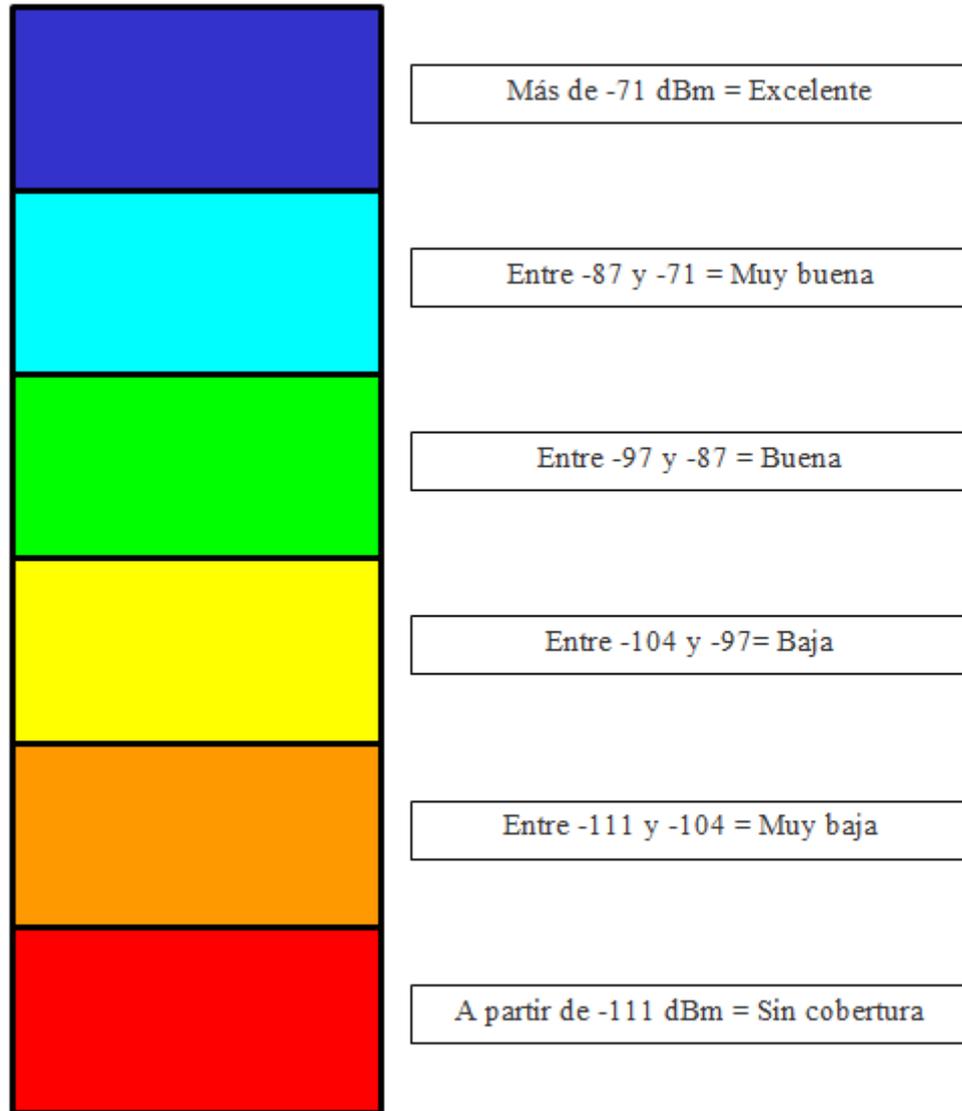


Tabla 3.3. Asociación de niveles de cobertura a una escala de colores

## LAN

El comando *LAN* nos permitirá obtener los datos de cada una de las redes locales que estén creadas en nuestro sistema. Como ya se comentó en el anterior capítulo se deben crear diferentes redes *LAN* para interconectar nuestros dispositivos. De esta forma se conseguirá sumar los anchos de bandas. Cada conexión entrante de un nuevo usuario será asignado a una *LAN* diferente y obtendrá conexión a internet por medio de la



*SIM* del dispositivo asignado a esa *LAN*. Así pues se consigue distribuir a los usuarios entre los tres dispositivos por lo que se reparte el ancho de banda.

Los datos que más nos interesaran del comando *LAN* serán los siguientes:

LAN1	LAN2	LAN1-1	LAN1-2	LAN2-1
<b>Description</b>		<b>Value</b>		
State		up		
Address		192.168.10.1		
MAC		00:11:2B:00:84:D8		
DHCP service		active		
DNS service		active		
MTU		1500		
Packets received / sent		0 / 0		
Bytes received / sent		0 B / 0 B		
Errors on reception / transmission		0 / 0		

Figura 3.4. Información generada por el comando `status -h lan`

- El parámetro State nos indica si el servicio *LAN* esta activo (UP). En caso contrario podríamos llegar a tener algún tipo de fallo de alimentación o de conexión de nuestro *router* esclavo con el *router* maestro.
- El parámetro Bytes received/sent nos proporciona el tráfico de bytes desde el último reseteo. Este dato nos servirá para realizar un análisis sobre qué red *LAN* ha sido más usada, si la distribución entre ellas es regular, etc.

## INFO

Por último este comando nos proporcionará información básica del sistema. Destacaremos dos datos principales:



System	Modules	Software
<b>System Information</b>		
Product name:	NetModule Router	
Product type:	NB1600	
Hardware version:	V3.1	
Serial number:	00112B0084D8	
CPU:	MPC830x RDB e300c3 (266.666664MHz)	
RAM:	64 MB (9.4 MB free)	
Flash:	128 MB (23.24 MB available)	
Temperature:	58.5 degree	
System time:	2015-05-13 15:29:06	
Uptime:	6:38	
Load average:	0.60, 0.30, 0.24	

Figura 3.5. Información generada por el comando `status -h info`

- El parámetro Temperature nos da la temperatura en el interior del dispositivo. Hemos considerado este dato para líneas futuras de trabajo, ya que una de ellas sería crear una caja para colocar los dispositivos y nos interesará tener un control de esta temperatura para añadir métodos de disipación si fuera necesario en caso de sobrecalentamiento.
- El parámetro System time nos proporcionara la hora interna del dispositivo, la cual será útil a la hora de generar informes y comparar resultados.

Como se puede ver el sistema nos permite obtener la información necesaria para el desarrollo de nuestro *software*.

Ahora solo queda hacer la operación de obtener información de forma automática. Para eso plantearemos varios métodos, de los cuales escogeremos el que más se adapte a nuestros requisitos.



COMANDO	DESCRIPCIÓN
summary	Con este comando se muestra un breve informe general.
Info	Con este comando se muestra la información del sistema y su configuración
config	Con este comando se muestra la información detallada de la configuración
system	Con este comando se muestra la información detallada de nuestro sistema
wwan	Con este comando se muestra la información del modulo WWAN
wlan	Con este comando se muestra la información del modulo WLAN
gnss	Con este comando se muestra la información del modulo GPS
eth	Con este comando se muestra la información del interfaz ethernet
lan	Con este comando se muestra la información de nuestras redes LAN
wan	Con este comando se muestra la información de nuestras redes WAN
mobileip	Con este comando se muestra el estado de nuestro modulo MobileIP
neigh	Con este comando se muéstrala información de todos los usuarios conectados a nuestro sistema
location	Con este comando se muestra la información GPS de nuestra localización actual

Tabla 3.4. Lista de comandos disponibles en nuestro *router*.



## 3.2.2 Puerto serie.

La primera opción que se nos plantea es obtener los datos por puerto serie, ya que el fabricante del dispositivo nos permite esta posibilidad.

Un puerto serie o puerto serial es una interfaz de comunicaciones de datos digitales, frecuentemente utilizado por computadoras y periféricos, donde la información es transmitida bit a bit enviando un solo bit a la vez. La transmisión en serie se puede explicar usando una analogía con las carreteras. Una carretera tradicional de un sólo carril por sentido sería como la transmisión en serie siendo los vehículos los bits que circulan por el cable.



Port	Protocol	Used by
SERIAL1	RS232	login console

Refresh

Figura 3.6. Configuración del puerto serie desde nuestro dispositivo

A nivel de programación nos interesan dos pines del puerto serie, ya que serán los encargados de comunicarse con nuestro dispositivo. Por uno leeremos y por otro escribiremos:

- El pin TX: Señal de recepción. Son los datos que se reciben. (entrada)
- El pin RX: Señal de transmisión. Por aquí salen los datos. (salida)

Esta opción aunque muy practica, plantea dos problemas fundamentales que nos harán desecharla como opción de comunicación *software*-sistema.

- El primer obstáculo que se nos plantea es que tendríamos que programar internamente el dispositivo (mediante el uso del SDK) para que proporcionara por el puerto serie los datos pertinentes. Esto generaría dos capas de *software* en dos sitios diferentes y podría generar futuros problemas y complicaciones.

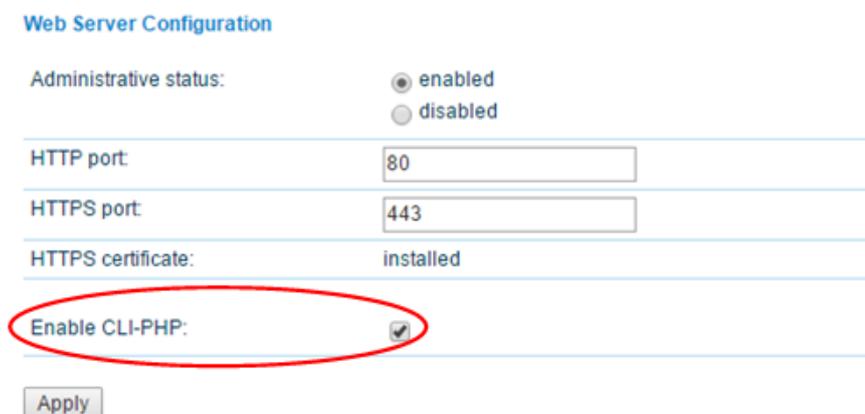


- El segundo inconveniente que aparece es la necesidad de disponer de un conversor USB-RS232 para obtener los datos de nuestro dispositivo. Ya que el dispositivo presenta un puerto USB interno que hace las veces de puerto serie. Añadido a esto, necesitaríamos un equipo con puerto RS232 para funcionar con este adaptador.

Basándonos en estos argumentos se ha desechado la opción de transmisión de datos mediante puerto serie.

### 3.2.3 Servidor *PHP*.

Habiendo desechado la opción de transmisión por puerto serie y repasando las características de nuestro equipo nos encontramos con una opción de servidor web. Esta opción es muy práctica y versátil y no necesita más configuración que activar una casilla en el *router*.



The screenshot shows a configuration page titled "Web Server Configuration". It includes several settings:

- Administrative status:  enabled,  disabled
- HTTP port: 80
- HTTPS port: 443
- HTTPS certificate: installed
- Enable CLI-PHP:  (This checkbox is circled in red in the original image)

An "Apply" button is located at the bottom left of the configuration area.

Figura 3.7. Activación del servidor web para la transmisión de datos

*PHP* es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un **servidor web** con un módulo de procesador de *PHP* que genera la página Web resultante. De esta forma nosotros solo tendremos que



introducir a nuestro servidor web una petición de datos en lenguaje *PHP*, y nuestro servidor web generará una página con los datos que se han pedido.

En el manual del dispositivo encontramos las instrucciones concretas para realizar una comunicación mediante el servidor web. Estas instrucciones corresponden al lenguaje *PHP*. Este método respeta los comandos explicados anteriormente para la obtención de datos. De esta forma el comando que usaremos será el comando

```
status - Display status information

Key usage:
  command=status [&arg0=<section>]

Notes:
  Available sections can be retrieved by running
  command=status&arg0=h.
  Please note that the status summary can be displayed without
  authentication.

Examples:
  http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
  admin01&command=status&arg0=h

  http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd=
  admin01&command=status&arg0=summary

  http://192.168.1.1/cli.php?version=2&output=html&command=status
```

Figura 3.8. Instrucciones en lenguaje *PHP* para la petición de datos mediante servidor web

Analizando estos datos podemos analizar y crear instrucciones para cada comando que usemos. Podríamos decir que una petición al servidor web se compone de dos partes (en *PHP* & representa un separador):

- **Cabecera:** Esta cabecera será común a todas las instrucciones y consistirá de varias partes:

➤ `http://192.168.3.1/cli.php?version=2&output=html&`

Representará la dirección IP del dispositivo al cual se dirija la petición. Recordemos que tenemos tres dispositivos con tres servidores web diferentes entre sí. Cambiando esta IP



podremos pedir la misma instrucción a los diferentes dispositivos.

➤ `usr=USUARIO&pwd=CONTRASEÑA&`

Con esta parte se representara el usuario y contraseña que se haya configurado inicialmente

El resto de texto no remarcado en negrita no varía entre los diferentes *routers*. Solo el texto en negrita puede variar entre los diferentes *routers* de nuestro sistema.

- **Comando:** Esta parte corresponderá con el comando que queremos transmitir al cliente *PHP* para que nos proporcione los datos correctos:

COMANDO	INSTRUCCIÓN EN PHP
GNSS	<code>&amp;command=status&amp;arg0=-s&amp;arg1=gnss</code>
NEIGH	<code>&amp;command=status&amp;arg0=-s&amp;arg1=neigh</code>
WAN	<code>&amp;command=status&amp;arg0=-s&amp;arg1=wan</code>
WWAN	<code>command=status&amp;arg0=-s&amp;arg1=wwan</code>
LAN	<code>&amp;command=status&amp;arg0=-s&amp;arg1=lan</code>
INFO	<code>&amp;command=status&amp;arg0=-s&amp;arg1=info</code>

Tabla 3.5. Comandos *PHP*

De esta forma un ejemplo de comando completo, uniendo ambas partes, sería el siguiente:

`http://192.168.3.1/cli.php?version=2&output=html&usr=USUARIO&pwd=CONTRASEÑA&command=status&arg0=-s&arg1=info`

El servidor web al recibir este comando por medio de un navegador web genera una página de texto plano en el cual aparecen los datos que se requerían:



```
System: NB2700 hostname (00:11:22:AA:BB:CC)
WAN1: WWAN1 is up (10.0.0.1, Mobile1, UMTS, -83 dBm, LAI 12345)
GPS: lat 47.377894, lon 8.540055, alt 282.200
OVPN: client on tun0 is up (10.0.8.4)
DIO: IN1=off, IN2=off, OUT1=on, OUT2=off
```

Figura 3.9. Ejemplo de texto plano devuelto por nuestro servidor web al recibir un comando *PHP*

De este modo solo necesitamos una aplicación *software* que sea capaz de leer de un navegador web texto plano, seleccionar el dato y procesarlo.



### 3.3 ENTORNO Y LENGUAJE DE PROGRAMACIÓN ESCOGIDOS.

La elección del lenguaje de programación y su entorno de desarrollo es fundamental. Necesitamos un lenguaje y entorno de programación que cumpla con los requisitos presentados anteriormente. De entre varias opciones como java, python, etc., hemos elegido usar *C#* como lenguaje y *Visual Studio* como entorno. Dados los conocimientos de programación adquiridos a lo largo de la carrera, *C#* es uno de los lenguajes que mejor se adaptaban.

*C#* es un lenguaje de programación orientado a objetos (nos permitirá usar objetos gráficos) desarrollado y estandarizado por *Microsoft* como parte de su plataforma *.NET*. *C#* es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Su sintaxis básica deriva de *C/C++* y utiliza el modelo de objetos de la plataforma *.NET*, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

*C#* se basa en *.NET*, que es un *marco de trabajo* (marco de trabajo) de que permita un rápido desarrollo de aplicaciones. La finalidad de este *marco de trabajo* es ofrecer una manera rápida y económica, a la vez que segura y robusta, de desarrollar aplicaciones permitiendo una integración más rápida y ágil entre empresas y un acceso más simple y universal a todo tipo de información desde cualquier tipo de dispositivo.

*Microsoft Visual Studio* es un entorno (*IDE*) para sistemas operativos *Windows*. Soporta múltiples lenguajes de programación tales como *C++*, *C#*, *Visual Basic .NET*, *F#*, *Java*, *Python*, *Ruby*, *PHP*; al igual que entornos de desarrollo web como *ASP.NET MVC*, *Django*, etc.

*Visual Studio* permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma *.NET*. Así se pueden crear aplicaciones que se comuniquen entre estaciones de trabajo, páginas web, dispositivos móviles, dispositivos embebidos, consolas, etc.



Una vez seleccionada nuestra aplicación describiremos brevemente como esta elección cumple con nuestros requisitos.

## SOFTWARE VISUAL.

La aplicación que se va a desarrollar tiene que estar en un entorno gráfico que facilite al usuario un fácil manejo y comprensión de esta. La finalidad de esta aplicación *software* es que pueda ser manejada por un usuario inexperto. El usuario manejará de forma intuitiva una aplicación gracias a los elementos que aporta un entorno grafico con botones, ventanas, selectores, etc. Con *C#* tendremos acceso a una gran cantidad de objetos gráficos que nos permitirán desarrollar una aplicación intuitiva.

## COMPATIBILIDAD CON LIBRERÍAS GMAP.

Nuestra aplicación *software* deberá funcionar en unas pruebas de campo en las que tendrá que dibujar la ruta seguida por el tren de alta velocidad y tomar valores. Por lo tanto se necesita una librería capaz de dibujar en un mapa una serie de coordenadas *GPS*. Para esto se han escogido un conjunto de librerías gratuitas llamadas “*Great Maps*” (*GMap .NET*). Estas librerías son totalmente compatibles con *C#*.

## CAPACIDAD DE LEER Y PROCESAR TEXTO PLANO GENERADO POR UN SERVIDOR WEB.

Para acceder a los datos necesitaremos que nuestra aplicación *software* envíe una solicitud vía web a un servidor *PHP* alojado en el *router*. Una vez enviada esta solicitud tendrá que leer la respuesta y de ella obtener los datos requeridos. Para ello *C#* tiene una serie de librerías que permiten hacer peticiones web y almacenar la respuesta. Una vez se tiene esa respuesta del servidor web, se tiene que buscar el dato en concreto que se quiera almacenar. Por ejemplo si yo quiero obtener las coordenadas *GPS* actuales (latitud, longitud y altura), tendría que enviarle el comando “*GNSS*” al servidor web.



Una vez enviado ese comando el servidor web me proporcionara una serie de datos de los cuales solo me interesan las coordenadas GPS. Como nosotros conocemos la estructura del texto plano que nos proporciona el servidor web, podemos acotar donde se encuentra nuestro dato.

```
GNSS1_LATITUDE=-4.717873
GNSS1_LONGITUDE=41.518854
GNSS1_ALTITUDE=716.50
GNSS1_HORIZONTAL_SPEED=0.00
GNSS1_VERTICAL_SPEED=0.00
```

Figura 3.10. Texto plano obtenido de nuestro servidor web

Como se puede apreciar en la imagen para obtener la latitud actual, deberíamos crear un programa que obtenga el dato que se encuentra entre “GNSS1\_LATITUDE=” y “GNSS1\_LONGITUDE=”, Y así con el resto de datos que queramos almacenar. De esta forma tenemos un método que de forma automática realiza el proceso de almacenar datos proporcionados por un servidor web. Este proceso requiere de conocer la estructura de cada respuesta de nuestro servidor ante cualquiera de los comandos nombrados anteriormente. En la siguiente imagen podemos apreciar las dos funciones necesarias para realizar una petición web y obtener los datos.



```
// Función para obtener el texto plano del servidor Web
public static string conectar_url(string url)
{
    WebRequest request = WebRequest.Create(url);
    request.Credentials = CredentialCache.DefaultCredentials;
    WebResponse response = request.GetResponse();
    Stream dataStream = response.GetResponseStream();
    StreamReader reader = new StreamReader(dataStream);
    string responseFromServer = reader.ReadToEnd();
    reader.Close();
    response.Close();
    return (responseFromServer);
}

// Función para seleccionar los datos dentro del texto plano
public static string sacar_datos(string desde, string hasta, string responseFromServer)
{
    Thread.CurrentThread.CurrentCulture = new CultureInfo("en-US");
    string str = responseFromServer;
    int first = str.IndexOf(desde);
    int last = str.IndexOf(hasta); //
    string str2 = str.Substring(first, last - first);
    char[] delimiter = { '"', '\t', '\n' };
    string[] words = str2.Split(delimiter, StringSplitOptions.RemoveEmptyEntries);
    string dato_limpio = words[1];
    return dato_limpio;
}
```

Figura 3.11. Fragmento de código que nos permite leer del servidor web y seleccionar un dato en particular de todo el texto

## COMPATIBILIDAD CON LIBRERÍAS PARA REPRESENTAR VALORES DE FORMA GRÁFICA.

De cara a los resultados de las pruebas de campo se requiere una comparación gráfica de los resultados obtenidos. C# gracias a su compatibilidad con *Microsoft Office* nos permite usar una librería asociada al recurso del sistema usado por *Microsoft Office Excel*, llamado *MSChart*. De esta forma contaremos con una serie de objetos que nos permitirán representar una serie de gráficas para comparar nuestros resultados.

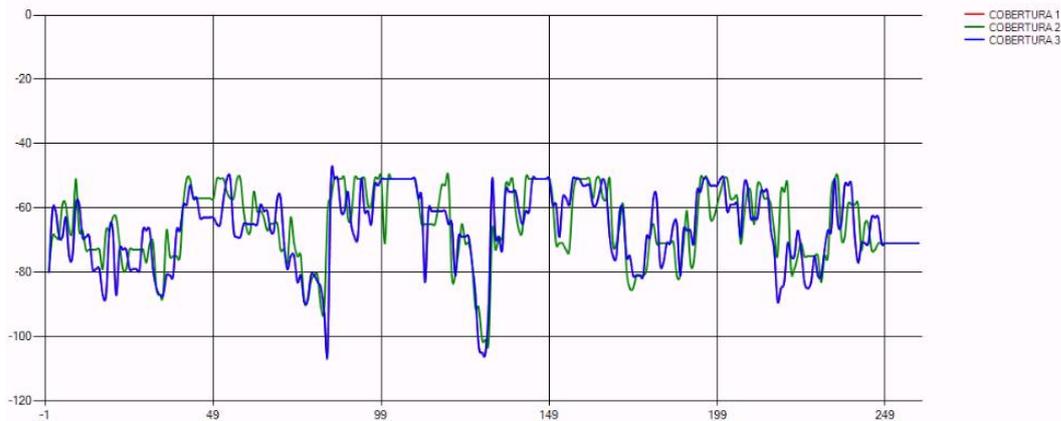


Figura 3.12. Ejemplo de representaciones gráficas con el recurso *MSChart*

## COMPATIBILIDAD CON LIBRERÍAS CAPACES DE ENVIAR CORREOS ELECTRÓNICOS.

Una de las mejoras que se añadió a posteriori fue el poder enviar los resultados por email. Esta mejora se implemento para mejorar el procesamiento de estos, ya que el usuario sería capaz de enviar los datos a su cuenta de correo electrónico personal y así poder analizarlos en cualquier momento sin tener que copiarlos del equipo de pruebas.

```
//Usando la clase Correo
Correos Cr = new Correos();
MailMessage mnsj = new MailMessage();
//Obtiene la fecha del sistema para anotarla en el correo
DateTime fecha_hora = DateTime.Now;
//Asunto...
mnsj.Subject = "ENVIO DE DATOS EL " + fecha_hora;
mnsj.From = new MailAddress("prueba.3700@gmail.com", "");
//Email para la direccion introducida por el Usuario
mnsj.To.Add(new MailAddress(direccion.Text));
//Obtiene la direccion donde se esta ejecutando el programa y la toma como ruta base
string directorioActual = Path.GetDirectoryName(
    Assembly.GetExecutingAssembly().GetModules()[0].FullyQualifiedName)+"\\data";
System.IO.File.Copy(directorioActual
    + "\\datos_de_medidas.zip", directorioActual + "\\datos_de_medidas3.zip", true);
mnsj.Attachments.Add(new Attachment(directorioActual + "\\datos_de_medidas3.zip"));
mnsj.Body = "Trayecto finalizado: " + fecha_hora;
//Envia correo..
Cr.MandarCorreo(mnsj);
mnsj.Dispose();
```

Figura 3.13. Fragmento del código usado para enviar los resultados por correo electrónico



## COMPATIBILIDAD CON LIBRERÍAS DE COMPRESIÓN DE ARCHIVOS EN .ZIP.

Otra de las mejoras que surgieron a posteriori fue la de comprimir todos los datos en un archivo con formato de compresión *ZIP*. Esta mejora se diseñó para mejorar el envío de correos electrónicos desde el programa ya que de esta forma solo se adjuntaría un archivo al correo y se ahorraría espacio por lo que se enviaría más rápido el correo.

Para realizar esta compresión de archivos recurrimos a una librería externa gratuita. La librería se llama *SharpZipLib* y añade un *archivo.dll* que añade una serie de opciones relacionadas con el formato de compresión *ZIP*.

### **3.4 LIBRERÍAS Y HERRAMIENTAS ESPECIALES NECESARIAS.**

A lo largo del desarrollo de nuestra aplicación *software* se han necesitado diferentes librerías y herramientas. A medida que se iba necesitando una librería o herramienta se iba añadiendo al programa.

En los siguientes apartados se mostraran todas las librerías y herramientas usadas.

#### **3.4.1 Librerías propias.**

En informática, una librería (biblioteca) es un conjunto de implementaciones funcionales, codificadas en un lenguaje de programación, que ofrece una interfaz bien definida para la funcionalidad que se invoca.

A diferencia de un programa ejecutable, el comportamiento que implementa una biblioteca no espera ser utilizada de forma autónoma (un programa sí: tiene un punto de entrada principal), sino que su fin es ser utilizada por otros programas, independientes y de forma simultánea. Por otra parte, el comportamiento de una biblioteca no tiene por qué diferenciarse en demasía del que pudiera especificarse en un programa. Es más, unas bibliotecas pueden requerir de otras para funcionar, pues el comportamiento



que definen refina, o altera, el comportamiento de la biblioteca original; o bien la hace disponible para otra tecnología o lenguaje de programación.

Las bibliotecas pueden vincularse a un programa (o a otra biblioteca) en distintos puntos del desarrollo o la ejecución, según el tipo de vínculo que se quiera establecer, tal y como se detalla en el apartado de "Tipos".

La mayoría de los sistemas operativos modernos proporcionan bibliotecas que implementan los servicios del sistema. De esta manera, estos servicios se han convertido en una "materia prima" que cualquier aplicación moderna espera que el sistema operativo ofrezca. Como tal, la mayor parte del código utilizado por las aplicaciones modernas se ofrece en estas bibliotecas.

En este apartado se hará mención a las librerías generales propias de .NET que hemos necesitado, así como las librerías externas que hemos tenido que implementar.

Cada librería incluida en nuestra aplicación cumple una o varias funciones. Como ya se comentó con anterioridad .NET permite añadir librerías externas mientras que sean compatibles. Esto nos abre un gran abanico de posibilidades a la hora de desarrollar nuestra aplicación *software*.

```
using System;
using System.Globalization;
using System.Threading;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.Windows.Forms;
using GMap.NET;
using GMap.NET.WindowsForms;
using System.IO;
using System.Net;
using System.Reflection;
using System.Net.Mail;
using System.IO.Compression;
using ICSharpCode.SharpZipLib.Core;
using ICSharpCode.SharpZipLib.Zip;
using ICSharpCode.SharpZipLib.Checksums;
```

Figura 3.14. Lista de todas las librerías usadas en nuestra aplicación *software*

A continuación se describirá brevemente la finalidad de algunas de las librerías propias de *Visual Studio*

- System.Globalization: La librería *System.Globalization* contiene clases que definen información relativa a la referencia cultural, incluido el



idioma, el país o región, los calendarios utilizados, los modelos de formato de fecha, divisa y números, y el criterio de ordenación de las cadenas. Estas clases resultan útiles para escribir aplicaciones globalizadas (de uso internacional).

- *System.Threading*: La librería *System.Threading* proporciona clases e interfaces que permiten la programación multiproceso. Además de las clases para sincronizar las actividades y el acceso a los datos (Mutex, Monitor, Interlocked, AutoResetEvent de subprocesos, etc.), esta librería incluye una clase de ThreadPool que le permita usar un grupo de subprocesos proporcionados, y una clase de Timer que ejecute métodos de devolución de llamada en subprocesos threadpool.
- *System.Collections.Generic*: La librería *System.Collections.Generic* contiene interfaces y clases que definen colecciones genéricas, permitiendo que los usuarios creen colecciones fuertemente tipadas para proporcionar una mayor seguridad de tipos y un rendimiento mejor que los de las colecciones no genéricas fuertemente tipadas.
- *System.ComponentModel*: La librería *System.ComponentModel* proporciona clases que se utilizan para implementar el comportamiento de componentes y controles en tiempo de diseño y en tiempo de ejecución. Esta librería incluye las clases e interfaces necesarias para implementar atributos, convertidores de tipos, enlaces a orígenes de datos y componentes de licencia.
- *System.Drawing*: La librería *System.Drawing* proporciona acceso a funcionalidad de gráficos básica de GDI, la interfaz de diseño de gráficos de *Windows*.
- *System.Windows.Forms*: La librería *System.Windows.Forms* contiene clases para crear aplicaciones basadas en *Windows* que aprovechan las ventajas de las características avanzadas de la interfaz de usuario disponibles en el sistema operativo *Microsoft Windows*.
- *System.IO*: La librería *System.IO* contiene tipos que permiten leer y escribir en archivos y flujos de datos, así como tipos que proporcionan compatibilidad básica con los archivos y directorios



- *System.Reflection*: La librería *System.Reflection* contiene tipos que recuperan información sobre los ensamblados, módulos, miembros, parámetros y otras entidades del código administrado examinando sus metadatos. Estos tipos se pueden utilizar también para manipular instancias de tipos cargados.
- *System.Net.Mail*: La librería *System.Net.Mail* contiene clases que se utilizan para enviar correo electrónico a un servidor *SMTP* (Simple Mail Transfer Protocol, Protocolo simple de transferencia de correo) para la entrega.

### 3.4.2 *GMap .NET*

Uno de los principales recursos con los que se ha trabajado en este proyecto ha sido la herramienta *GMap .NET*. *GMap .NET* es una plataforma libre para el desarrollo de aplicaciones usando mapas. Las utilidades de esta plataforma son inmensas. Desde trazado de ruta, geolocalización, representación de polígonos en mapas, etc.

A lo largo del desarrollo de la aplicación *software* este recurso ha sido usado para diversos fines, entre ellos la representación de una ruta en tiempo real.

Para incluir esta herramienta en *Visual Studio* basta con agregar dos librerías proporcionadas por la web oficial de la herramienta. Una vez agregadas podemos hacer uso de esta herramienta. El objeto más interesante que aporta *GMap .NET* es el ***GMapControl*** que nos permite agregar un mapa en blanco totalmente configurable. En la siguiente imagen se muestra un cuadro de diálogo para realizar la configuración básica de este objeto.

GMap.NET	
Bearing	0
CanDragMap	True
GrayScaleMode	False
MarkersEnabled	True
MaxZoom	18
MinZoom	2
MouseWheelZoomType	MousePositionAndCenter
NegativeMode	False
PolygonsEnabled	True
RoutesEnabled	True
ShowTileGridLines	False
Zoom	5

3.15. Configuración del objeto de *GMapControl*

Igual que la mayoría de objetos de *Visual Studio*, *GMapControl* requiere de una inicialización:

```
private void Form1_Load(object sender, EventArgs e)
{
    // Initialize map:
    gmap.MapProvider = GMap.NET.MapProviders.BingMapProvider.Instance;
    GMap.NET.GMaps.Instance.Mode = GMap.NET.AccessMode.ServerOnly;
    gmap.SetCurrentPositionByKeywords("Maputo, Mozambique");
}
```

Figura 3.16. Ejemplo de inicialización del objeto *GMapControl* aquí llamado "gmap"

Una de las grandes ventajas de *GMap .NET* es la compatibilidad con diferentes servidores de aplicaciones de mapas. Desde la configuración del objeto *GMapControl* se puede escoger el servidor deseado. A continuación se muestra una pequeña lista de ellos:

- *CloudMadeMapProvider*
- *GoogleMapProvider*
- *OpenCycleMapProvider*
- *OpenStreetMapProvider*
- *WikiMapiaMapProvider*
- *YahooMapProvider*
- *BingMapProvider*

Aunque comúnmente se suele presentar *GoogleMaps* como el indiscutible servidor de aplicaciones de mapas, en este caso hemos escogido el servidor de *Bing* (*Microsoft*).

Google presenta una gran ventaja frente a los demás servidores ya que es el servidor más completo y detallado que hay. Pero eso no siempre es una ventaja, ya que nuestra aplicación *software* está destinada a hacer peticiones rápidas cada poco tiempo. De tal forma que la rapidez de respuesta es una prioridad, y el que mejor rapidez de respuesta tiene en este caso es *Bing*.

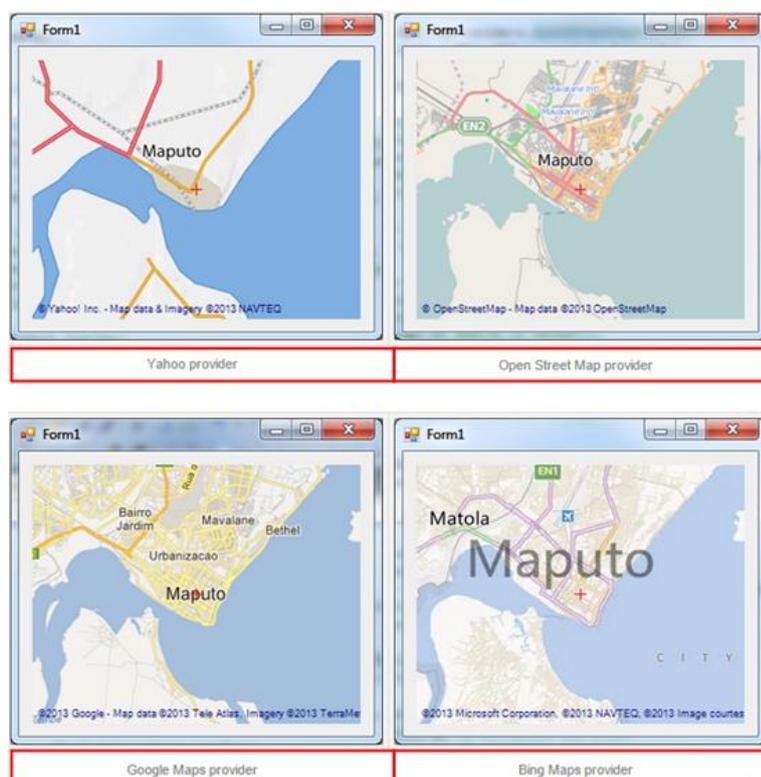


Figura 3.17. Comparación entre cuatro de los mejores servidores de aplicaciones de mapas con los que se puede trabajar en *GMap .NET*

Uno de los problemas que surgieron con esta herramienta es que nuestro objetivo era dibujar la ruta seguida por un tren de alta velocidad. El problema que se planteaba es que *GMap .NET* tiene una función para dibujar rutas. Esta función trabaja uniendo dos puntos dados en forma de línea. Esta línea no es estrictamente una línea recta, ya que dibuja esa línea adaptando a la carretera/camino más cercano. En el caso de un tren de alta velocidad esto es un inconveniente, ya que no trabaja con las vías de tren y dibuja una ruta incorrecta.

Para solucionar este problema se desechó la función que genera rutas y se utilizó una función que genera polígonos, de tal forma que genera un polígono entre dos puntos, o lo que es lo mismo una línea recta.

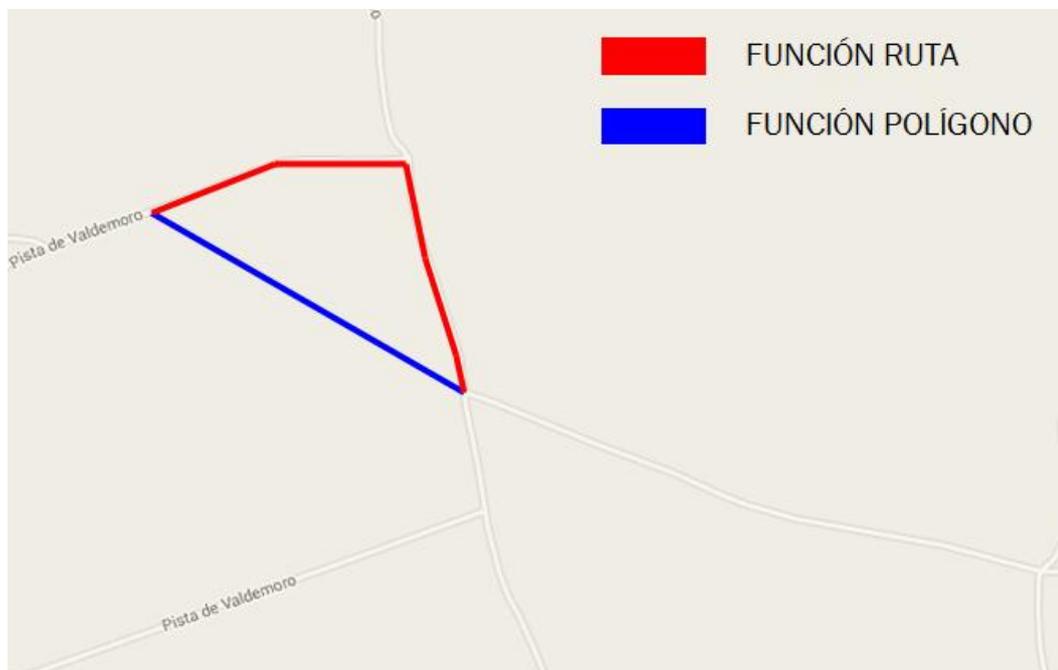


Figura 3.18. Diferencia entre la función ruta (rojo) y la función polígono (azul)

### 3.4.3 Compresión en formato ZIP

Como se indicó anteriormente, uno de los añadidos que se implementaron a nuestra aplicación *software* fue la capacidad de comprimir archivos en formato de compresión *ZIP*. Esto nos mejorará notablemente la función de enviar correos electrónicos que tiene nuestra aplicación, ya que el tamaño de los datos que se adjuntaran al correo será menor y por lo tanto el envío será más rápido.

*ZIP* es un formato de fichero bastante simple, que comprime cada uno de los archivos de forma separada. Comprimir cada archivo independientemente del resto de archivos comprimidos permite recuperar cada uno de los ficheros sin tener que leer el resto, lo que aumenta el rendimiento. Gracias a esta compresión el tamaño final de los archivos comprimidos se reduce considerablemente (depende del factor de compresión usado).



Una de las grandes ventajas del formato *ZIP* frente a otros formatos de compresión es la popularidad del mismo. Gracias a esto el formato *ZIP* es el más usado y ahorrará al usuario final el uso de un *software* externo para realizar una descompresión ya que las versiones más recientes de *Windows* y *Linux* traen de serie un *software* interno de serie.

*Visual Studio* por defecto no proporciona un método eficiente para realizar una compresión en formato *ZIP*. Esto nos obliga a investigar una solución compatible con *C#*. Investigando encontramos una librería libre llamada *SharpZipLib*. *SharpZipLib* es una librería que permite la compresión en formatos *Zip*, *GZip*, *Tar* y *BZip2*. Esta librería está escrita totalmente en *C#* por lo que es totalmente compatible con nuestro entorno y lenguaje de programación.

```
// Función para comprimir archivos en el formato de compresión ZIP
using (ZipOutputStream s = new ZipOutputStream(File.Create(directorioActual
+ "\\datos_de_medidas.zip")))
{
    s.SetLevel(9); // 0-9, 9 being the highest compression
    byte[] buffer = new byte[4096];
    for (int h = 0; h <= 4; h++)
    {
        /*-----*/
        ZipEntry entry = new
        ZipEntry(Path.GetFileName(archivos[h]));
        entry.DateTime = DateTime.Now;
        s.PutNextEntry(entry);
        using (FileStream fs = File.OpenRead(archivos[h]))
        {
            int sourceBytes;
            do
            {
                sourceBytes = fs.Read(buffer, 0,
                buffer.Length);
                s.Write(buffer, 0, sourceBytes);
            } while (sourceBytes > 0);
        }
    }
    s.Finish();
    s.Close();
}
```

Figura 3.19. Fragmento de código para comprimir una carpeta en el formato de compresión *ZIP*

### 3.5 ESTRUCTURA GENERAL DEL PROGRAMA.

Una vez introducidos los recursos necesarios para cumplir los objetivos de nuestra aplicación *software*, se procederá a mostrar la estructura y funcionalidad de cada uno de sus elementos.

Al iniciar nuestra aplicación nos aparecerá un menú general que nos mostrara diferentes botones para acceder a los diferentes submenús. El menú principal tiene internamente un sistema para evitar fallos de conexión. De esta forma si se inicia la aplicación sin estar conectado al sistema el sistema lo detecta y automáticamente produce un error evitando así errores innecesarios.



Figura 3.20. Diseño del menú principal de nuestra aplicación

### 3.5.1 Estado Actual.

El primer botón de nuestro menú principal es el botón principal nos llevará al programa “Estado Actual”. La finalidad de este programa será mostrar los datos medidos por cada uno de los tres *routers*.



La finalidad de este programa es tener un control de nuestro sistema en cualquier instante para revisiones periódicas o problemas puntuales y evitar al usuario el hecho de tener que lidiar con la aplicación web propia del dispositivo.

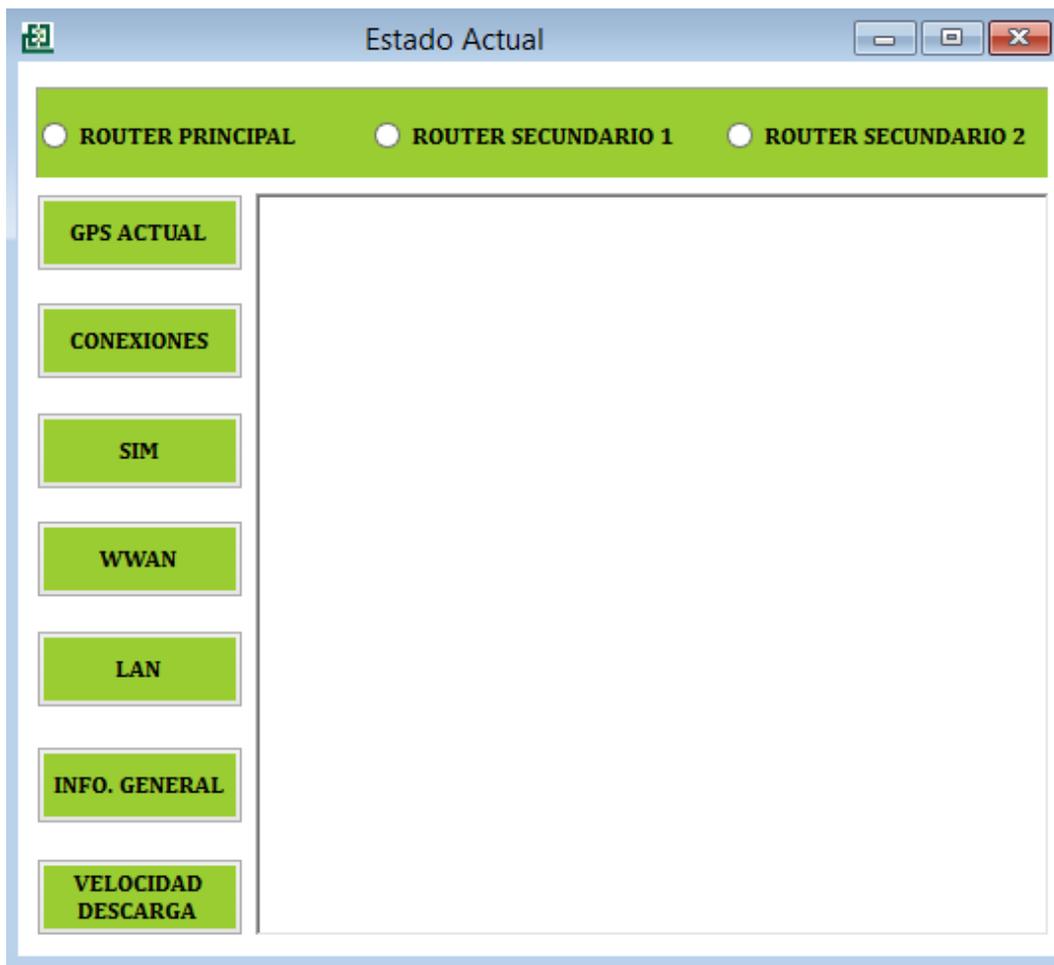


Figura 3.21. Diseño del programa "Estado Actual"

## 3.5.2 Control de trayecto.

El segundo botón nos lleva al submenú “Control de Trayecto”. Este submenú está completamente enfocado a las pruebas de campo. Dentro de este submenú se encuentran una serie de programas que más tarde explicaremos detalladamente.

No obstante, algunas de os programas que se encuentran en este submenú también están destinados a actividades externas de las pruebas de campo, como puede ser el subprograma “Posición Actual”.



Figura 3.22. Diseño del submenú "Control Trayecto"

## 3.5.3 Opciones.



El último botón del menú principal nos lleva al programa “Opciones”. Este programa está destinado configurar determinados aspectos de nuestro sistema. El programa “Opciones” está diseñado como una vía para el desarrollo de futuras líneas de trabajo con este proyecto.



Figura 3.23. Diseño del programa "Opciones"

En los siguientes apartados se procederá a describir de forma detallada cada uno de los programas dentro de nuestra aplicación.

### 3.6 ESTADO ACTUAL.

El programa estado actual se encargará de mostrar por pantalla la información pedida por el usuario. Como se ha indicado con anterior este programa está destinado a realizar controles en momentos puntuales.

El diseño de esta aplicación está enfocado a obtener la información de cada uno de los tres *routers* que forman nuestro sistema. El usuario tiene que indicar dos datos: la información que desea visualizar y de que *router* de los tres la quiere. Por lo tanto este programa presenta tres partes fundamentales

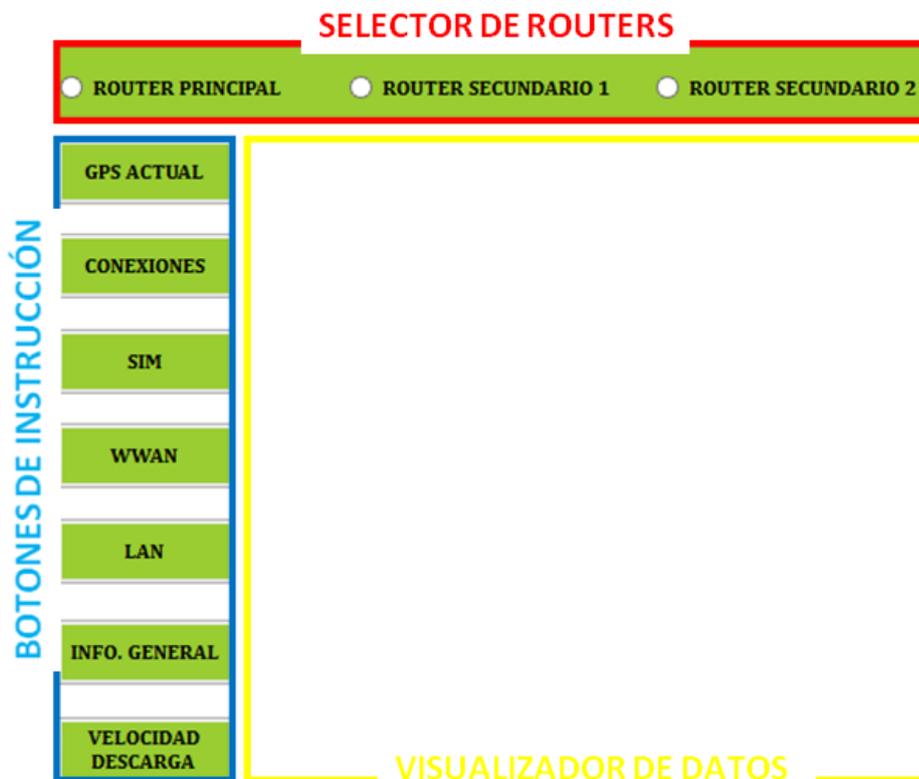


Figura 3.24. Diferentes partes del programa "Estado Actual"

## SELECTOR DE ROUTERS

Lo primero que debe hacer el usuario es seleccionar el dispositivo del que quiera conocer los datos. Por defecto el programa viene preseleccionado con el *router* 1 para evitar errores innecesarios. El programa está diseñado para que el usuario solo pueda visualizar un dato de un *router* a la vez.

Cuando se selecciona un nuevo *router* el programa reacciona ante ese evento, actualizando los comandos *PHP* para ese *router* en concreto.



```
// Cada vez que se selecciona un router se actualizan
// los comandos para ese router en concreto
private void router_1_CheckedChanged(object sender, EventArgs e)
{
    wwan_url = wwan_url_1;
    wan_url = wan_url_1;
    info_url = info_url_1;
    neigh_url = neigh_url_1;
    richTextBox1.Text = "";
}
}
```

Figura 3.25. Ejemplo de actualización de comandos *PHP* ante el evento de cambio de *router*

## BOTONES DE INSTRUCCIÓN

La siguiente acción que debe realizar el usuario es la de pulsar el botón correspondiente a los datos que quiera visualizar. En este caso tampoco se pueden visualizar varios datos simultáneamente.

Cada botón tiene asignado un evento que envía el comando *PHP* concreto y gestiona la respuesta. De este modo cada vez que e pulse un nuevo botón se cambiarán los datos que se estén visualizando por pantalla.

```
/*
 * Ejemplo de comportamiento de un botón:
 *
 *     -> Envía una petición al servidor web.
 *     -> Obtiene la respuesta.
 *     -> Elimina las cabeceras HTML.
 *     -> Muestra los datos por pantalla.
 */
private void info_Click(object sender, EventArgs e)
{
    string url = info_url;
    string responseFromServer = conectar_url(url);
    string respuesta = sacar_datos(responseFromServer);
    richTextBox1.Text = "";
    richTextBox1.Text = respuesta;
}
}
```

Figura 3.26. Código asociado al evento de cada botón

## VISUALIZADOR DE DATOS

Por último cuando el usuario pulsa un botón, aparecerá por pantalla la información solicitada correspondiente al *router* seleccionado. Si hay ya información por pantalla el programa se encarga de borrar todo antes de mostrar más información. En la siguiente imagen se puede ver un ejemplo de la información *GPS* proporcionada del *router* 1 cuando se pulsa el botón *GNSS*.

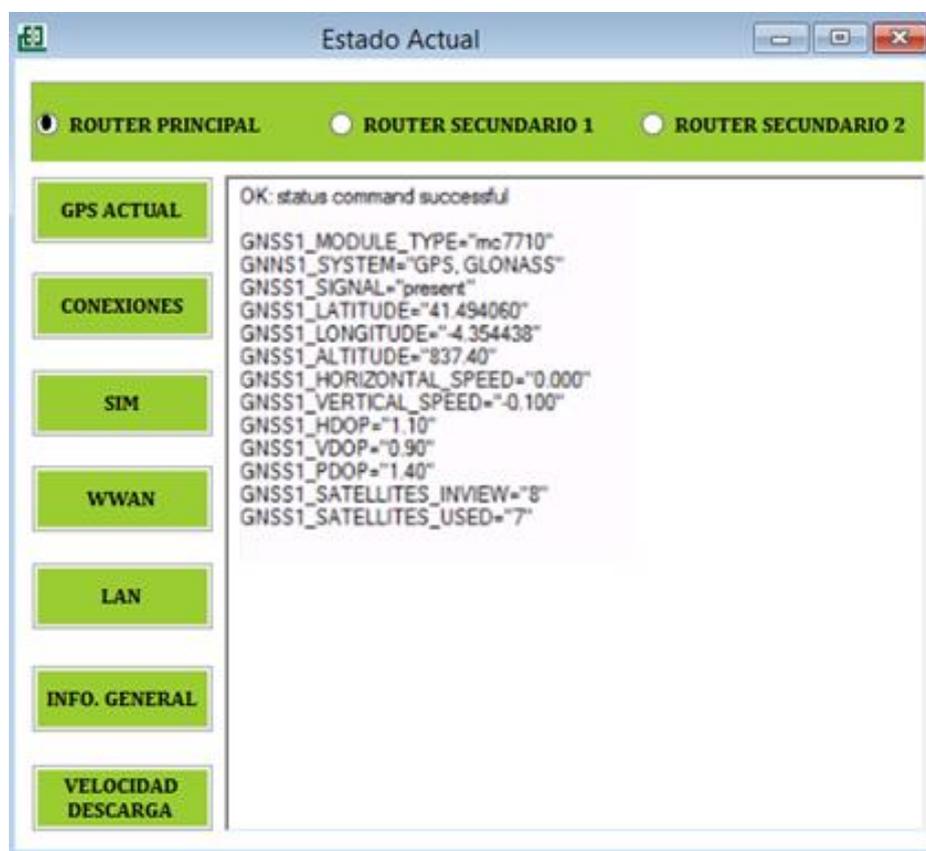


Figura 3.27. Ejemplo de la información mostrada por pantalla del programa "Estado Actual"

## 3.7 CONTROL DE TRAYECTO.

En este punto vamos a explicar detalladamente el objetivo y funcionamiento de cada uno de los programas que se encuentran en el submenú "Control de trayecto". Como se ha comentado anteriormente, esta parte de la aplicación está destinada a la realización de pruebas de campo.

### 3.7.1 Posición actual.

Este es el primer programa que aparece en nuestro submenú es el de "Posición Actual". La finalidad de este programa es mostrar la posición actual en un mapa a partir de las coordenadas GPS proporcionadas por nuestro sistema.

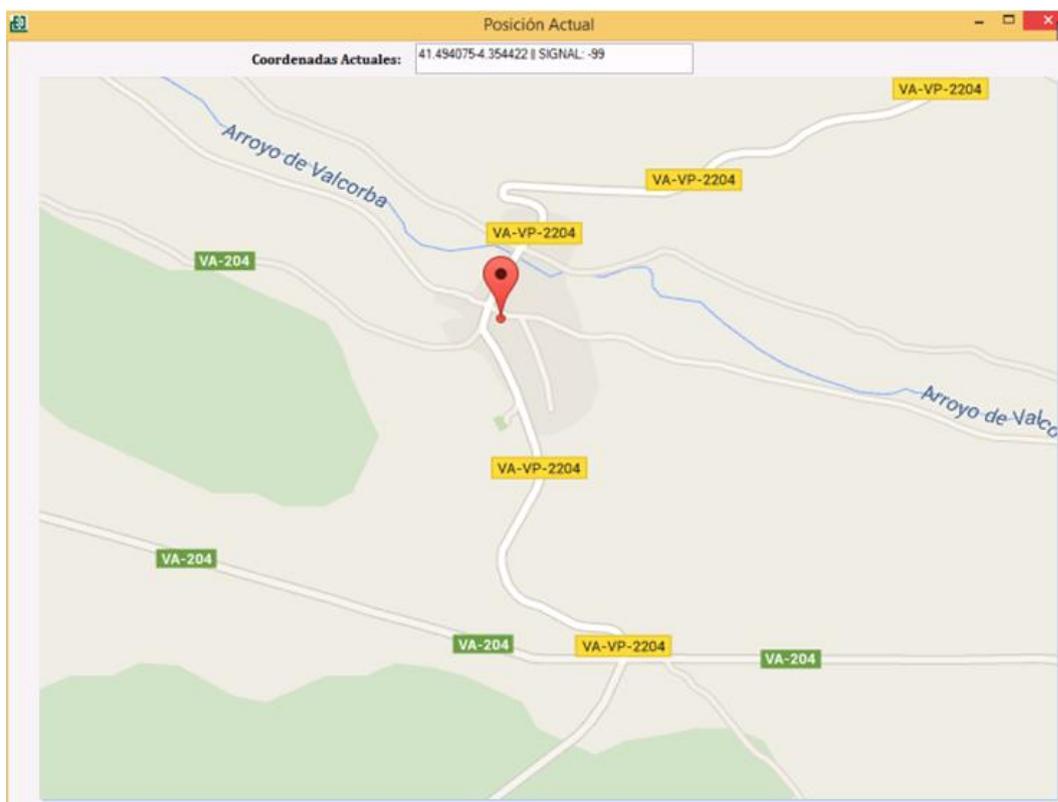


Figura 3.28. Ejemplo del programa "Posición Actual" en funcionamiento

Como se puede observar en la anterior imagen este programa trabaja con dos objetos diferentes.

El primero de ellos se trata de el objeto de la herramienta *GMap .NET* que se encarga de representar la posición *GPS* proporcionada por el *router* en un mapa. En nuestro caso hemos escogido que se ponga una marca en forma de “bombilla” para representar la posición *GPS* actual. El color de dicha marca depende de la cobertura media actual. Mediante la escala de colores definida anteriormente obtenemos diferentes colores acordes con la cobertura.

Con esto se pretende que el usuario pueda comprobar visualmente que el servicio *GPS* y el servicio *SIM* estén operativos. Esto evitará problemas en las pruebas de campo, ya que lo primero que debe hacer el usuario antes de iniciar una prueba de campo es comprobar mediante este programa que el servicio *GPS* y el servicio *SIM* están activos y funcionan correctamente.

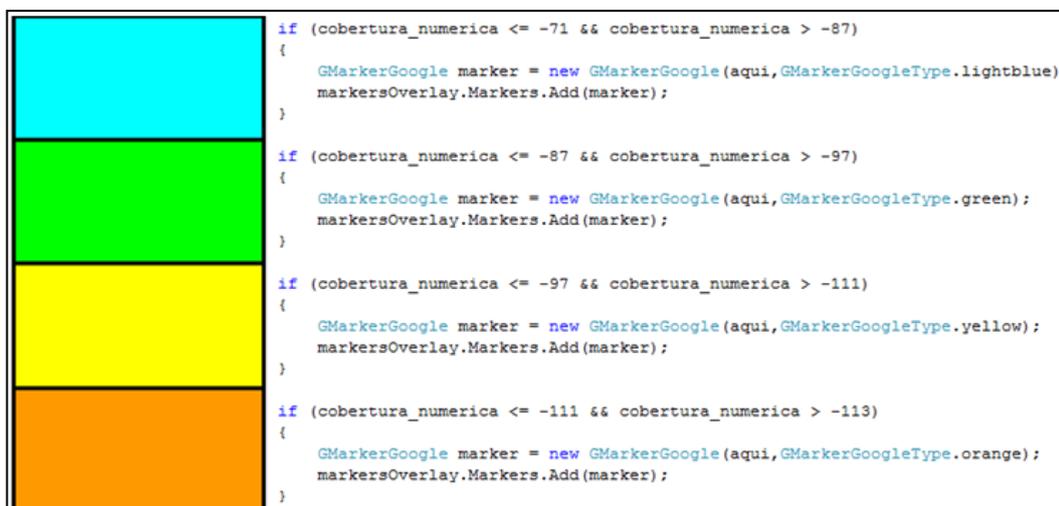


Figura 3.29. Fragmento del código selector de color a partir de la escala de colores definida anteriormente.

El segundo objeto disponible en este programa es un cuadro de texto (de “solo lectura”), que muestra por pantalla las coordenadas *GPS* exactas junto con el nivel de cobertura actual. Esta cuadro de texto mostrara el mensaje “ERROR” si alguno de los sistemas no funcionan. Junto al mensaje de error también aparecerá la causa del mismo, permitiendo al usuario tener un control exhaustivo del estado de los servicios del *router*.

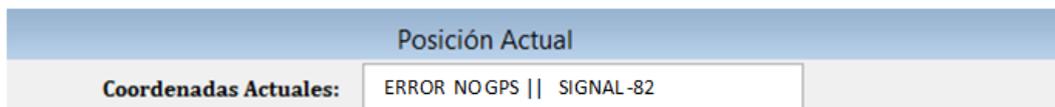


Figura 3.30. Ejemplo del mensaje de error en el cuadro de texto

### 3.7.2 Trayecto en tiempo Real.

“Trayecto en tiempo real” es el programa más importante de cara a las pruebas de campo. Con el podemos mapear un trayecto realizado por nuestro tren de alta velocidad y almacenar periódicamente los datos que nuestro sistema nos va proporcionando. Este programa es el más extenso y complicado de toda nuestra aplicación.

El funcionamiento de este programa es sencillo. Cuando el usuario pulse el botón “START” el programa comenzará a tomar medidas y representar valores. El programa se detendrá solo cuando se pulse el botón “STOP”. Una vez realizado esto se abrirá un dialogo que permitirá almacenar al usuario todos los datos recogidos. El usuario dispondrá de total libertad de elegir el lugar donde se guardaran dichos datos. De esta forma se puede ver que a pesar de ser el programa más complejo de toda la aplicación el funcionamiento de cara al usuario es muy sencillo.

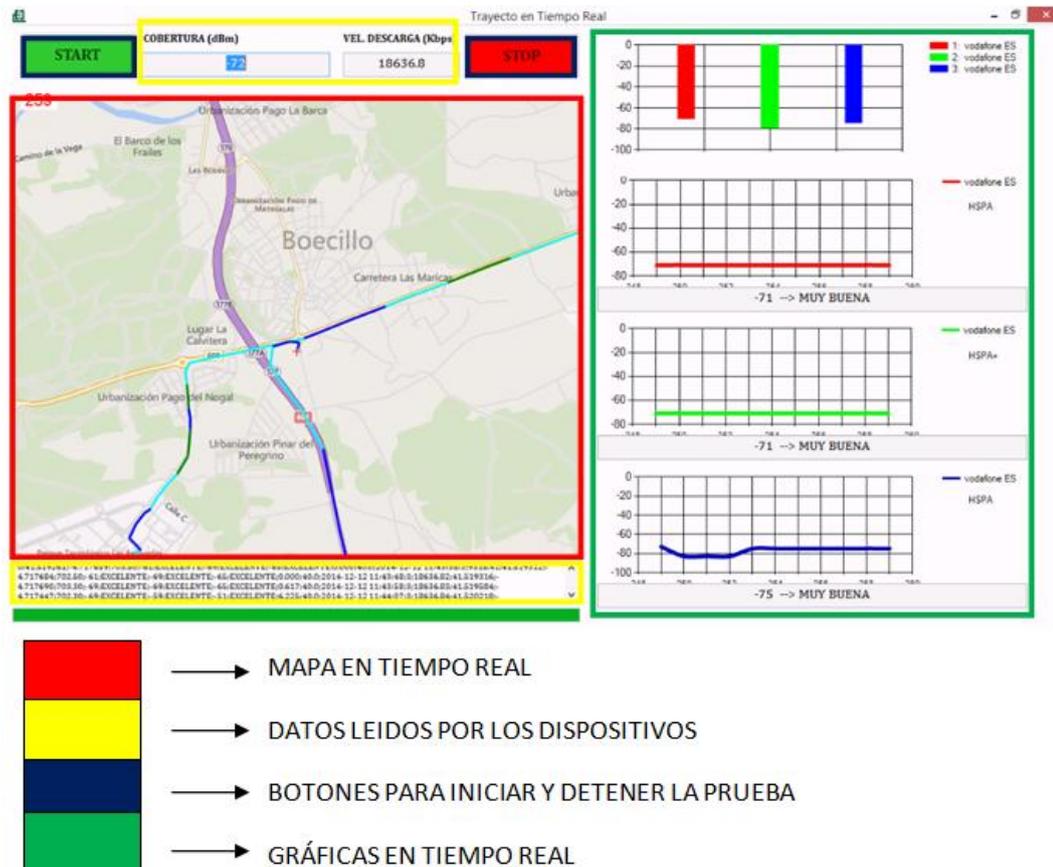


Figura 3.31. Desglose de los componentes del programa "Trayecto En Tiempo Real"

Como se puede apreciar en la imagen anterior, el programa contiene cuatro objetos diferentes. Existe otro objeto que visualmente no se aprecia pero es fundamental para el funcionamiento de este programa: el *BackgroundWorker*.

A continuación se dará una breve explicación de la función en el programa de cada uno de estos objetos.

### BOTONES PARA INICIAR Y DETENER LA PRUEBA

Los botones de "Start" y "Stop" son la única interacción que tiene el usuario con el programa. Se podría decir que el programa consta de dos estados: detenido y tomando medidas. Al iniciar el programa se parte de un



estado detenido. Una vez que se pulsa el botón “Start” se pasa de estado detenido al estado tomando medidas. Solo se volverá al estado detenido cuando el usuario pulse el botón “Stop”.

Cabe destacar que una vez se pulse el botón “Stop” no se podrá volver a pulsar el botón “Start” quedando ambos botones bloqueados. Para poder volver a pulsar el botón “Start” el usuario debe reiniciar el programa cerrándolo y volviendo a abrirlo. De esta forma evitamos que se sobrescriban datos de diferentes trayectorias.

## BACKGROUNDWORKER

Una vez iniciada pulsado el botón “Start”, comienza a funcionar el *BackgroundWorker*. Un *BackgroundWorker* permite ejecutar una operación en un subproceso dedicado e independiente. Es decir, permite realizar operaciones costosas o duraderas en un hilo diferente al del interfaz, de manera que la aplicación sigue respondiendo al usuario mientras este trabajo se procesa en segundo plano. De esta forma nuestro sistema obtendrá datos de los dispositivos cada cinco segundos y representara los mismos en el mapa, las gráficas y los cuadros de texto.

Gracias al *BackgroundWorker* se consigue automatizar el proceso de toma y representación de datos. Este proceso solo se detendrá cuando el usuario pulse el botón “Stop”. En ese momento el *BackgroundWorker* terminara la tarea que esté realizando en ese momento y se detendrá.

## MAPA EN TIEMPO REAL

Como su propio nombre indica este objeto representa un mapa en tiempo real de la trayectoria. Cuando se inicia el proceso, el mapa representa la trayectoria en el mapa. Esta trayectoria consistirá en una progresión de líneas entre dos puntos. El programa tiene dos variables para ello: “localización actual” y “localización anterior”. De esta forma cada vez que se tomen nuevos datos de los dispositivos el programa actualizara las variables de esta forma conservando la localización anterior. De esta forma dos líneas consecutivas en la trayectoria siempre tendrán un punto en común.

```
localización_anterior = localización_actual  
localización_actual   = nuevo_dato_medido
```

Figura 3.32. Actualización de las variables para dibujar una trayectoria continua

Cada línea que se dibuja, se dibuja de un color. Este color viene dado por la escala de colores asociada a los niveles de cobertura. Ya que tenemos tres valores de coberturas diferentes (cada una asociada a un *router*), usaremos un valor promedio entre ellas.

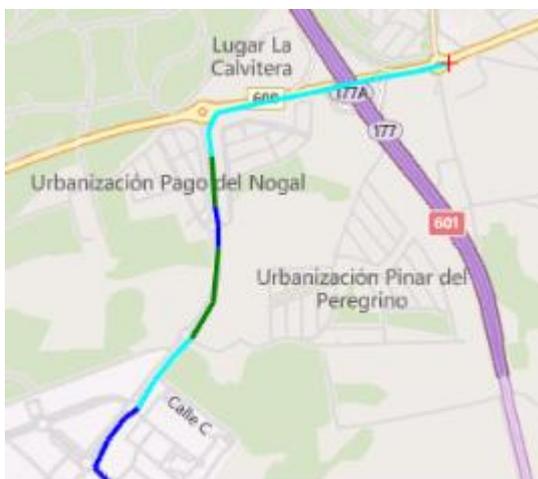


Figura 3.33. Ejemplo de una trayectoria representada en el mapa

## DATOS LEIDOS POR LOS DISPOSITIVOS

En este programa aparecen varios cuadros de texto de solo lectura, que nos indican diversos datos interesantes. Estos cuadros están diseñados para mantener un control del buen funcionamiento del programa durante las pruebas de campo.

En la parte superior del programa nos encontramos con dos cuadros de texto diferentes:

- **COBERTURA (dBm)**: En este cuadro se muestra por pantalla el dato correspondiente a la cobertura media medida. Este dato se calcula



realizando el promedio entre los tres valores de señal de cobertura proporcionada por los tres *routers* de nuestro sistema. Esto ayudara al usuario a analizar diferentes situaciones durante el trayecto de pruebas. Pueden surgir determinadas situaciones difíciles de plasmar en los informes finales de las pruebas de campo y que se podrían apreciar durante el trayecto. Por ejemplo es complicado reflejar determinados tramos de túneles y pasos bajo tierra en el mapa. El usuario puede observar el comportamiento del sistema en estas determinadas situaciones y anotar posibles problemas.

- **VELOCIDAD DESCARGA (kbps):** Con este dato se pretende mostrar por pantalla una aproximación del valor de la velocidad de descarga de un dispositivo (por defecto el *router* central). Este dato va ligado a la cobertura, ya que a menor señal de cobertura menor ancho de banda y menor velocidad de descarga.

En la parte inferior del programa nos encontramos con un cuadro de texto amplio que muestra por pantalla la información de las 4 últimas medidas realizadas por nuestro sistema. En este cuadro se muestra la misma estructura almacenada en nuestro informe final. La finalidad de esta ventana es comprobar que se están tomando los datos correctamente y que no existe ningún problema.

```
0;41.519261;-4.717659;703.50;-61;EXCELENTE;-69;EXCELENTE;-65;EXCELENTE;0.000;40.0;2014-12-12 11:43:38;3;9318.41;41.519312;-  
4.717684;702.50;-61;EXCELENTE;-69;EXCELENTE;-65;EXCELENTE;0.000;40.0;2014-12-12 11:43:48;3;18636.82;41.519316;-  
4.717690;703.30;-69;EXCELENTE;-69;EXCELENTE;-65;EXCELENTE;0.617;40.0;2014-12-12 11:43:58;3;18636.83;41.519584;-  
4.717447;702.30;-69;EXCELENTE;-59;EXCELENTE;-51;EXCELENTE;6.225;40.0;2014-12-12 11:44:07;3;18636.84;41.520218;-
```

Figura 3.34. Ejemplo de los datos presentados en el cuadro de texto inferior

## GRÁFICAS EN TIEMPO REAL

Por último, se ha utilizado un objeto para crear graficas (derivado de MSChart). Estas graficas son una representación de la cobertura en tiempo real. Cada vez que se toma una medida se representa en esta graficas. En total hay cuatro gráficas diferentes:

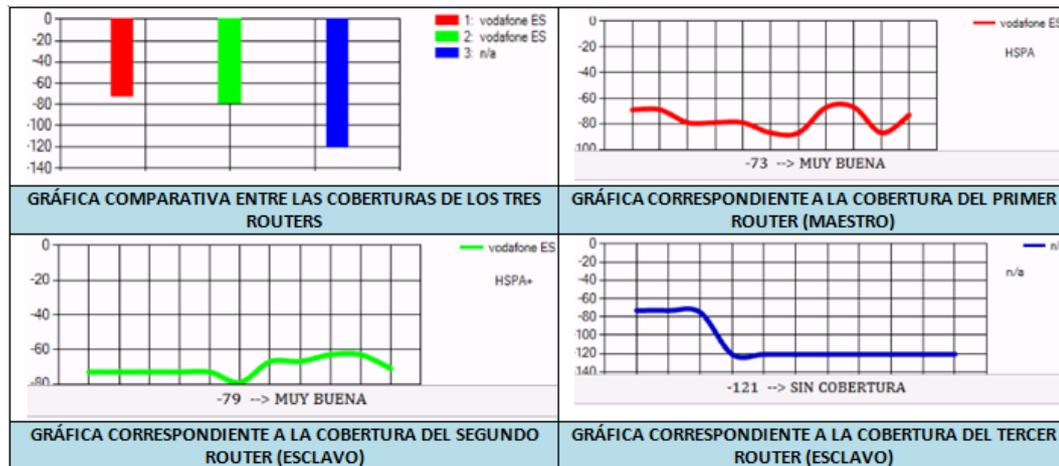


Figura 3.35. Ejemplo de las cuatro gráficas que aparecen en nuestro programa.

Como se puede apreciar en la imagen, la primera gráfica corresponde a una gráfica de barras que compara la cobertura medida por cada *router*. Las otras tres gráficas representan la evolución de la cobertura en cada dispositivo, teniendo en cuenta los últimos diez valores medidos.

De las cuatro gráficas destacamos la primera. Está nos permitirá apreciar la diferencia de la cobertura medida por cada dispositivo en tiempo real. Esto cobrará especial importancia en las pruebas de campo en las que se coloquen diferentes operadores en los diferentes *routers*. De esta forma se realizará una comparación de las diferentes compañías telefónicas en igualdad de condiciones.

## 3.7.3 Generador de Caché.

Desde el principio hemos hablado del objeto de *GMap* encargado de representar puntos en el mapa. Pero este objeto exige una conexión a internet para poder conectarse al servidor de mapas correspondiente y actualizar este objeto con el mapa de la zona. Esto representa un inconveniente, ya que no siempre va a existir una buena conexión a internet para generar mapa. Por eso existe una forma de no depender de una conexión y poder almacenar previamente mapas en memoria. Esta es la finalidad del programa “Generador de Caché”.

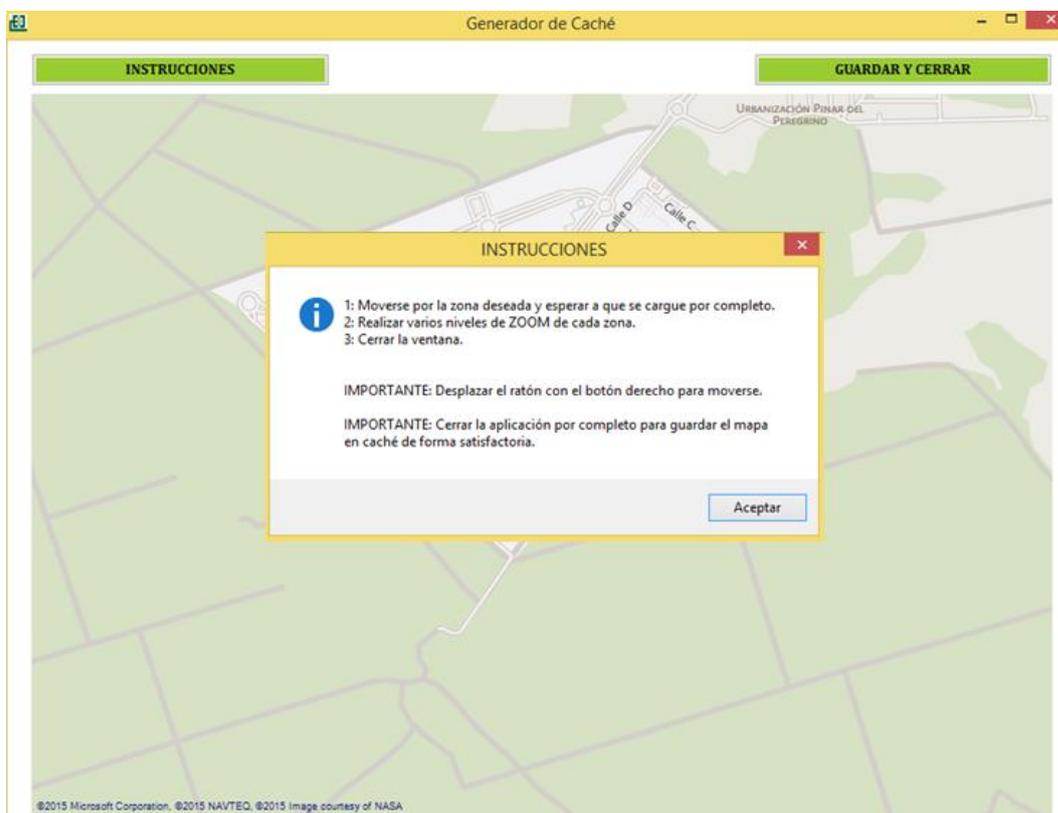


Figura 3.36. Diseño del programa "Generador de Caché"

Nada más iniciar este programa aparece un pequeño aviso en el que se indica el funcionamiento del proceso para generar caché. El funcionamiento es muy simple y lo único que debe hacer el usuario es moverse por el mapa, por la zona donde se vayan a realizar las pruebas de

campo, y aplicar diferentes niveles de zoom en cada zona. De esta forma se generará y almacenará automáticamente caché del mapa.

### 3.7.4 Visualizar Informes.

De cara al que el usuario no necesite ningún *software* externo a nuestra aplicación, se ha implementado un pequeño programa “Visualizador de informes”.

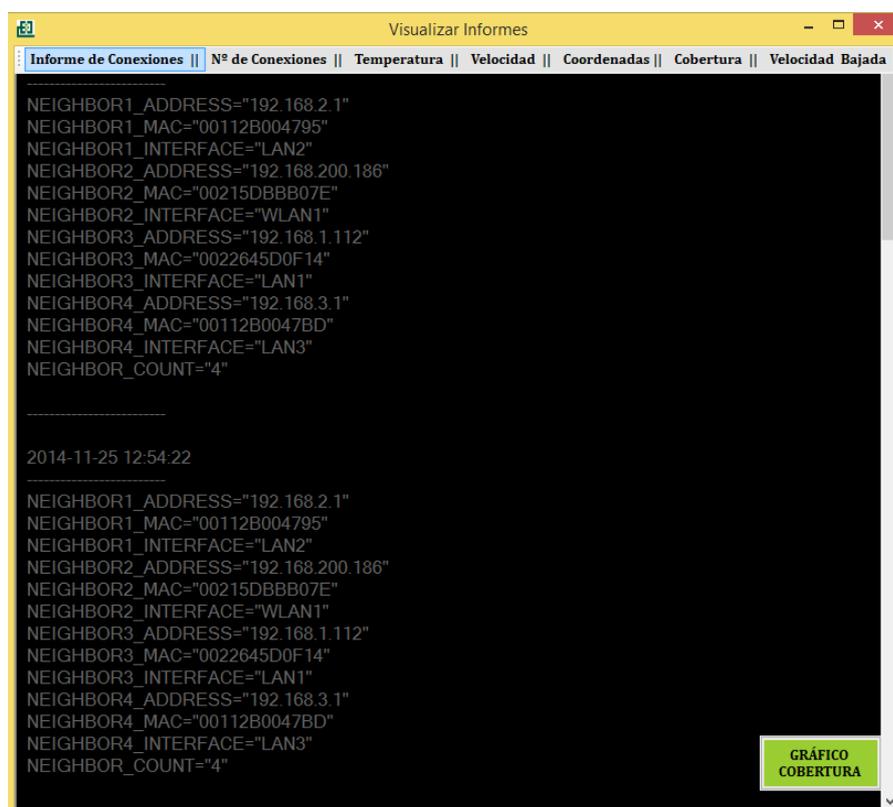


Figura 3.37. Ejemplo de funcionamiento del programa "Visualizador de Informes"

Este programa tiene un funcionamiento muy sencillo. El usuario tiene que seleccionar en la barra superior el informe que desea visualizar. Si se quiere visualizar otro informe solo se tiene que seleccionar otro de la barra superior.

Como se puede apreciar, en la parte inferior del programa se incluye un botón para visualizar el gráfico con la escala de colores asociada a los niveles de cobertura.



Aunque aquí se puedan visualizar algunos datos medidos, los verdaderos resultados se almacenaran en el equipo una vez que el usuario termine el programa “Trayecto Continuo”. Posteriormente se detallaran los archivos que se generaran y su finalidad.



### 3.7.5 Visualizar Gráficas.

Al igual que en el anterior apartado, el programa “Visualizar Gráficas” tiene la finalidad de mostrar unas graficas comparativas entre las diferentes medidas tomadas. En ese apartado indicábamos que el programa “Visualizar Informes” no era tan importante ya que el usuario contaría con un archivo con los resultados finales. Con el programa “Visualizar Gráficas” pasa al contrario ya que ahorrara al usuario generar dichas gráficas con un *software* externo, como puede ser *Microsoft Office Excel*.

Las gráficas que se pueden visualizar están predefinidas y son las siguientes:

- Cobertura frente a Recorrido: En esta gráfica se representa los niveles de cobertura medidos por cada *router* frente al recorrido total realizado en la prueba de campo.
- Velocidad frente a Recorrido: En esta gráfica se representa la velocidad *GPS* frente al recorrido total realizado en la prueba de campo.
- Temperatura frente a Recorrido: En esta gráfica se representa la temperatura media de *router* maestro frente al recorrido total realizado en la prueba de campo.
- Cobertura frente a Velocidad: En esta gráfica se representa los niveles de cobertura medidos por cada *router* frente a la velocidad *GPS*.
- Velocidad de descarga (Bajada) frente a Recorrido: En esta gráfica se representa la velocidad de descarga media frente al recorrido total realizado en la prueba de campo.
- Cobertura frente a Velocidad de descarga (Bajada): En esta gráfica se representa los niveles de cobertura medidos por cada *router* frente a la velocidad de descarga media.



- Nº de conexiones frente a Velocidad de descarga (Bajada): En esta gráfica se representa el número de conexiones frente a la velocidad de descarga media.

### 3.7.6 Enviar Datos por Email.

Este último programa dentro del submenú de “Control de Trayecto” tiene la finalidad de enviar los resultados medidos en las pruebas de campo al destinatario introducido por el usuario. La cuenta de correo desde donde se envían, es una cuenta dedicada específicamente a esta aplicación *software*.

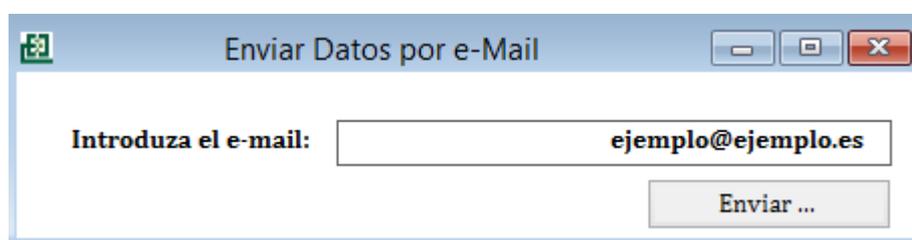


Figura 3.38. Diseño del programa "Enviar Datos por Email"

El programa contiene un sistema por el cual salta un mensaje de error si existe algún fallo en el envío de los datos por email.

## 3.8 OPCIONES

Por último en nuestra aplicación se encuentra el programa opciones. Este programa nos permite interactuar con nuestros *routers*. En este programa nos encontramos con dos opciones diferentes. Al iniciar este programa aparece una ventana de aviso que informa al usuario que el uso de este programa puede alterar y borrar la configuración actual del *router*



Figura 3.39. Diseño del programa "Opciones"

El programa consta de dos botones y un cuadro de texto de solo lectura. Cada botón contiene un comando diferente. Si se pulsa un botón y el sistema lo ejecuta correctamente aparecerá en el cuadro de texto la confirmación de que todo ha funcionado correctamente. Los dos botones contienen las siguientes funciones:

- Reiniciar Router: Este comando reinicia el sistema. Esto es muy útil cuando se quieren reiniciar todos los sistemas.
- Factory Reset: Este comando devuelve la configuración del *router* a sus valores de fábrica. Hay que tener especial cuidado con este



comando ya que borra toda nuestra configuración actual. Este comando está pensado para futuras líneas de trabajo.

### 3.9 FICHEROS GENERADOS.

Como se ha indicado con anterioridad, nuestra aplicación generará una serie de ficheros en los que se reflejarán las medidas tomadas durante las pruebas de campo. Estos ficheros de resultados están orientados a poder reflejar los datos medidos de la forma más intuitiva posible.

Cada conjunto de datos medidos durante el trayecto se incluirá en nuestros ficheros de resultado. Al terminar una prueba de campo el programa dará la posibilidad al usuario de guardar los resultados en la localización del disco duro que se desee. Una vez hecho esto el usuario dispondrá de los datos en cualquier momento sin necesidad de usar la aplicación.

Los ficheros que se generaran son los siguientes:

- Fichero de coordenadas 1 (.kml) que muestra el trayecto realizado en las pruebas de campo, con una escala de colores correspondiente al nivel de cobertura del *router* 1 (maestro).
- Fichero de coordenadas 2 (.kml) que muestra el trayecto realizado en las pruebas de campo, con una escala de colores correspondiente al nivel de cobertura del *router* 2 (esclavo).
- Fichero de coordenadas 3 (.kml) que muestra el trayecto realizado en las pruebas de campo, con una escala de colores correspondiente al nivel de cobertura del *router* 3 (esclavo).
- Fichero de datos (.csv) en el que se recogen de forma ordenada todos los datos medidos a lo largo de la prueba de campo.
- Grafico de coberturas en el que se indica la escala usada para asociar los diferentes niveles de cobertura a una escala de colores.



## Diseño de Sistemas WIFI en Trenes de Alta Velocidad



Nombre	Tipo	Tamaño
 coordenadas_3	Archivo KML	73 KB
 coordenadas_2	Archivo KML	73 KB
 coordenadas	Archivo KML	73 KB
 medidas	Archivo de valores...	55 KB
 grafico_coberturas	Imagen PNG	13 KB

Figura 3.40. Ejemplo de los ficheros generados por nuestra aplicación



## 3.9.1 Fichero de datos .CSV

Los archivos CSV son un tipo de documento en formato abierto sencillo para representar datos en forma de tabla, en las que las columnas se separan por punto y coma y las filas por saltos de línea. El formato CSV es muy sencillo y no indica un juego de caracteres concreto, ni cómo van situados los bytes, ni el formato para el salto de línea. Esto conviértete al formato CSV como una gran solución para generar un fichero con los datos obtenidos de las pruebas de campo.

Para interpretar un archivo CSV y ordenarlo correctamente se pueden emplear muchas soluciones diferentes, las más comunes son las hojas de cálculo. Esto ayuda a que el usuario pueda analizar los datos con la solución *software* más cómoda para él.

	A	B	C	D	E	F	G	H	I	J	K	L
1		LATITUD	LONGITUD	ALTURA	COBERTURA	NIVEL COBE	COBERTURA	NIVEL COBE	COBERTURA	NIVEL COBE	VELOCIDAD	TEMPERATU
2	0	41,519261	-4,717659	703,5	-61	EXCELENTE	-69	EXCELENTE	-65	EXCELENTE	0	40
3	1	41,519312	-4,717684	702,5	-61	EXCELENTE	-69	EXCELENTE	-65	EXCELENTE	0	40
4	2	41,519316	-4,71769	703,3	-69	EXCELENTE	-69	EXCELENTE	-65	EXCELENTE	0,617	40
5	3	41,519584	-4,717447	702,3	-69	EXCELENTE	-59	EXCELENTE	-51	EXCELENTE	6,225	40
6	4	41,520218	-4,716718	704,1	-63	EXCELENTE	-59	EXCELENTE	-51	EXCELENTE	8,643	40
7	5	41,520542	-4,716341	703,6	-75	MUY BUENA	-67	EXCELENTE	-65	EXCELENTE	4,321	40
8	6	41,520651	-4,715822	701,2	-75	MUY BUENA	-67	EXCELENTE	-65	EXCELENTE	6,636	40
9	7	41,521153	-4,716389	700,1	-59	EXCELENTE	-51	EXCELENTE	-59	EXCELENTE	8,746	40
10	8	41,521624	-4,717002	700,9	-59	EXCELENTE	-67	EXCELENTE	-73	MUY BUENA	7,974	40
11	9	41,522737	-4,716308	699,3	-69	EXCELENTE	-67	EXCELENTE	-73	MUY BUENA	15,33	40
12	10	41,523536	-4,715463	701,5	-69	EXCELENTE	-73	MUY BUENA	-73	MUY BUENA	11,163	40
13	11	41,523919	-4,714793	700,2	-69	EXCELENTE	-73	MUY BUENA	-73	MUY BUENA	9,311	40
14	12	41,525041	-4,713893	704,1	-79	MUY BUENA	-73	MUY BUENA	-75	MUY BUENA	14,97	40
15	13	41,525957	-4,712927	703,4	-79	MUY BUENA	-73	MUY BUENA	-123	SIN COBERT	16,411	40,5
16	14	41,526665	-4,712041	702,5	-79	MUY BUENA	-73	MUY BUENA	-123	SIN COBERT	11,421	40
17	15	41,528074	-4,710866	700,2	-87	BUENA	-79	MUY BUENA	-123	SIN COBERT	18,829	40
18	16	41,530155	-4,710632	700,4	-87	BUENA	-67	EXCELENTE	-123	SIN COBERT	17,903	40,5
19	17	41,530977	-4,710604	701,5	-67	EXCELENTE	-67	EXCELENTE	-123	SIN COBERT	9,569	40,5
20	18	41,531777	-4,710778	701,8	-67	EXCELENTE	-63	EXCELENTE	-123	SIN COBERT	13,993	40,5
21	19	41,533688	-4,711029	706,1	-87	BUENA	-63	EXCELENTE	-123	SIN COBERT	17,337	40,5
22	20	41,53467	-4,711198	710,7	-73	MUY BUENA	-71	MUY BUENA	-123	SIN COBERT	11,884	40,5
23	21	41,535392	-4,710689	713,5	-73	MUY BUENA	-79	MUY BUENA	-123	SIN COBERT	14,147	40,5
24	22	41,535825	-4,708095	712	-73	MUY BUENA	-79	MUY BUENA	-123	SIN COBERT	21,658	40,5
25	23	41,536169	-4,705914	711,4	-79	MUY BUENA	-73	MUY BUENA	-123	SIN COBERT	23,767	40,5
26	24	41,536692	-4,702533	709,9	-79	MUY BUENA	-73	MUY BUENA	-123	SIN COBERT	23,664	40,5
27	25	41,536994	-4,700449	708,9	-79	MUY BUENA	-73	MUY BUENA	-123	SIN COBERT	19,086	40,5
28	26	41,537296	-4,698832	709,1	-79	MUY BUENA	-73	MUY BUENA	-123	SIN COBERT	12,81	40,5
29	27	41,537768	-4,697299	709	-67	EXCELENTE	-73	MUY BUENA	-123	SIN COBERT	19,858	40,5
30	28	41,538735	-4,694039	707,7	-67	EXCELENTE	-77	MUY BUENA	-123	SIN COBERT	26,34	40,5
31	29	41,539819	-4,690314	707,6	-67	EXCELENTE	-71	MUY BUENA	-123	SIN COBERT	27,831	40,5

Figura 3.41. Ejemplo de un archivo CSV generado en unas pruebas de campo y abierto con una hoja de cálculo



## 3.9.2 Fichero de coordenadas .KML

*KML* es un lenguaje de etiquetas de marcado basado en *XML* para representar datos geográficos en tres dimensiones. Fue desarrollado para ser manejado con *Keyhole LT*, precursor de *Google Earth*. Los ficheros *KML* a menudo suelen distribuirse comprimidos como ficheros *KMZ*.

Un fichero *KML* especifica una característica (un lugar, una imagen o un polígono) para *Google Earth*. Contiene título, una descripción básica del lugar, sus coordenadas (latitud y longitud) y alguna otra información.

Un documento *KML* de ejemplo podría ser:

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
  <Placemark>
    <name> NOMBRE ASIGNADO AL LUGAR</name>
    <description>DESCRIPCIÓN DEL LUGAR</description>
    <Point>
      <coordinates>-5.843868,43.360758,0</coordinates>
    </Point>
  </Placemark>
</kml>
```

Para poder interpretar un documento *KML* es recomendable usar *Google Earth*, aunque existen diferentes alternativas. En nuestro caso usaremos *Google Earth* ya que es la mejor elección.

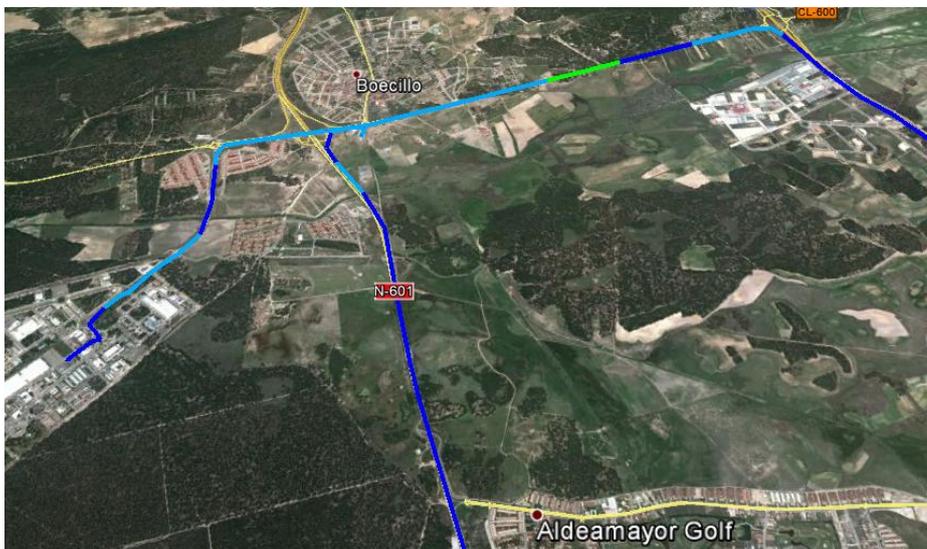


Figura 3.42. Ejemplo de la interpretación de un documento KML en Google Earth

Nuestra aplicación generará tres archivos *KML* distintos. Cada uno corresponderá a los datos aportados por cada uno de los tres *routers* de nuestro sistema.

Los valores que nos anotaremos en nuestros documentos *KML* serán las coordenadas del punto actual, las coordenadas del punto anterior, y el color asociado al nivel de cobertura.

```

<Placemark <styleUrl #s_ylw-pushpin styleUrl> <LineString <coordinates --4.717659,41.519261 -4.717659,41.519261 /></coordinates> </LineString> </Placemark>
<Placemark <styleUrl #s_ylw-pushpin styleUrl> <LineString <coordinates --4.717684,41.519312 -4.717659,41.519261 /></coordinates> </LineString> </Placemark>
<Placemark <styleUrl #s_ylw-pushpin styleUrl> <LineString <coordinates --4.717690,41.519316 -4.717684,41.519312 /></coordinates> </LineString> </Placemark>
<Placemark <styleUrl #s_ylw-pushpin styleUrl> <LineString <coordinates --4.717447,41.519584 -4.717690,41.519316 /></coordinates> </LineString> </Placemark>
<Placemark <styleUrl #s_ylw-pushpin styleUrl> <LineString <coordinates --4.716718,41.520218 -4.717447,41.519584 /></coordinates> </LineString> </Placemark>
<Placemark <styleUrl #s_ylw-pushpin styleUrl> <LineString <coordinates --4.716341,41.520542 -4.716718,41.520218 /></coordinates> </LineString> </Placemark>
<Placemark <styleUrl #s_ylw-pushpin styleUrl> <LineString <coordinates --4.715822,41.520651 -4.716341,41.520542 /></coordinates> </LineString> </Placemark>
<Placemark <styleUrl #s_ylw-pushpin styleUrl> <LineString <coordinates --4.716389,41.521153 -4.715822,41.520651 /></coordinates> </LineString> </Placemark>
<Placemark <styleUrl #s_ylw-pushpin styleUrl> <LineString <coordinates --4.717002,41.521624 -4.716389,41.521153 /></coordinates> </LineString> </Placemark>
<Placemark <styleUrl #s_ylw-pushpin styleUrl> <LineString <coordinates --4.716308,41.522737 -4.717002,41.521624 /></coordinates> </LineString> </Placemark>

```

COLOR DE LA LINEA

DOS PUNTOS GPS QUE FORMAN LA LINEA QUE REPRESENTARA UN TRAMO DE LA RUTA

Figura 3.43. Ejemplo de la estructura de nuestros archivos *KML*

A la hora de escoger los colores se ha intentado que fueran códigos de colores estándar en XML. Los diferentes colores se expresan a continuación:



COLOR	CODIGO KML	
AZUL	#m_ylw-pushpin	
CIAN	#msn_ylw-pushpin2	
VERDE	#msn_ylw-pushpin0	
AMARILLO	#msn_ylw-pushpin1	
NARANJA	#msn_ylw-pushpin3	
ROJO	#msn_ylw-pushpin	

Tabla 3.6. Colores usados en el documento *KML*

## 3.10 ORGANIZACIÓN DE LOS FICHEROS Y MODIFICACIÓN DE LAS VARIABLES.

La aplicación *software* no solo genera ficheros para el usuario, también genera ficheros que usara internamente. La mayoría de estos ficheros servirán para grabar los resultados de las pruebas de campo y poderlos representar de forma gráfica posteriormente.

Todos estos ficheros se almacenaran en una carpeta dentro del directorio donde se ejecuta la aplicación. Esta carpeta llamada “data” contendrá ficheros de texto con la información correspondiente a cada parámetro que se quiera representar gráficamente.

	datos_de_medidas	25/11/2014 12:06	Archivo WinRAR Z...	18 KB
	velocidad	25/11/2014 13:00	Archivo TXT	1 KB
	temperatura	25/11/2014 12:55	Archivo TXT	1 KB
	plantilla_rutas	01/09/2014 12:07	Archivo TXT	6 KB
	informe_conexiones	25/11/2014 12:55	Archivo TXT	4 KB
	coordenadas	25/11/2014 12:55	Archivo TXT	1 KB
	conexiones	25/11/2014 12:55	Archivo TXT	1 KB
	cobertura3	25/11/2014 12:55	Archivo TXT	1 KB
	cobertura2	25/11/2014 12:55	Archivo TXT	1 KB
	cobertura_media	25/11/2014 12:55	Archivo TXT	1 KB
	cobertura	25/11/2014 12:55	Archivo TXT	1 KB
	bajada	25/11/2014 12:55	Archivo TXT	1 KB

Figura 3.44. Archivos auxiliares que usa la aplicación *software*

Por otro lado existe un fichero dentro de la carpeta que contiene una serie de parámetros que pueden ser variables en determinadas circunstancias. El programa obtiene el valor de diferentes variables de este archivo. La finalidad de esto es que se pueda cambiar determinados parámetros de la aplicación sin modificar el código de la aplicación.

El archivo que contiene el valor de determinadas variables se denomina “datos.cfg”. El formato CFG es un formato que se aplica a archivos de configuración que suministran determinados parámetros a los programas. Este archivo contendrá el valor de los comandos *PHP* que nuestra aplicación usara para comunicarse con nuestro *router*.



```
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd= &command=status&arg0=-s&arg1=gns  
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd= &command=status&arg0=-s&arg1=neigh  
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd= &command=status&arg0=-s&arg1=wan  
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd= &command=status&arg0=-s&arg1=wwan  
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd= &command=status&arg0=-s&arg1=lan  
http://192.168.1.1/cli.php?version=2&output=html&usr=admin&pwd= &command=status&arg0=-s&arg1=info  
http://tartarus.org/~simon/putty-snapshots/x86/psftp.exe  
http://192.168.2.1/cli.php?version=2&output=html&usr=admin&pwd= &command=status&arg0=-s&arg1=wwan  
http://192.168.3.1/cli.php?version=2&output=html&usr=admin&pwd= &command=status&arg0=-s&arg1=wwan  
http://192.168.2.1/cli.php?version=2&output=html&usr=admin&pwd= &command=status&arg0=-s&arg1=neigh  
http://192.168.2.1/cli.php?version=2&output=html&usr=admin&pwd= &command=status&arg0=-s&arg1=wan  
http://192.168.2.1/cli.php?version=2&output=html&usr=admin&pwd= &command=status&arg0=-s&arg1=info  
http://192.168.3.1/cli.php?version=2&output=html&usr=admin&pwd= &command=status&arg0=-s&arg1=neigh  
http://192.168.3.1/cli.php?version=2&output=html&usr=admin&pwd= &command=status&arg0=-s&arg1=wan  
http://192.168.3.1/cli.php?version=2&output=html&usr=admin&pwd= &command=status&arg0=-s&arg1=info
```

Figura 3.45. Contenido del archivo "datos.cfg"



## 4 REALIZACIÓN DE PRUEBAS DE CAMPO.

En este capítulo se tratará el significado de las pruebas de campo y su desarrollo en la Centro Tecnológico *CARTIF*.

Para la realización de estas pruebas se ha necesitado el siguiente material:

- Sistema formado por nuestros tres routers, las antenas celulares, la antena *GPS* y un sistema de alimentación para los tres *routers*.
- Un ordenador portátil para ejecutar la aplicación y procesar los datos proporcionados por los tres *routers*.
- Un vehículo de pruebas, en nuestro caso se ha usado una furgoneta especial equipada para realizar pruebas en carretera. Esta furgoneta cuenta con un sistema de alimentación que nos sirve para alimentar nuestro sistema.
- Dispositivos que se puedan conectar por WIFI y consumir ancho de banda, como son por ejemplo los teléfonos móviles.



Figura 4.1. Imagen de la furgoneta especializada usada en las pruebas de campo, en la que se puede apreciar la situación de las difentes antenas



## 4.1 OBJETIVO DE LAS PRUEBAS DE CAMPO.

Uno de los objetivos fundamentales de la aplicación *software* que se ha desarrollado es el de realizar unas pruebas de campo en el cual se realicen las pruebas necesarias para valorar nuestro sistema. La importancia de estas pruebas es un objetivo fundamental en el proyecto planteado por el Centro Tecnológico *CARTIF* y *MAREV ENGINEERING*. Los objetivos fundamentales que deben conseguir estas pruebas de campo se resumen a continuación:

- Analizar el comportamiento del sistema en determinadas situaciones: Mediante estas pruebas se espera evaluar a nuestro sistema en las situaciones adversas y analizar el comportamiento y posibles soluciones ante estas. Un ejemplo de situación adversa puede ser el atravesar una zona de baja cobertura y ver cómo reacciona el sistema si se queda bloqueado, cierra conexiones, etc.
- Comparar diferentes compañías telefónicas en igualdad de condiciones: Gracias a disponer un sistema con tres *routers* diferentes podemos comparar diferentes compañías telefónicas y evaluar las diferencias entre los servicios que nos ofrecen. Al tener tres *routers* podemos insertar tarjetas *SIM* de diferentes compañías en cada uno de ellos. De esta forma los tres *routers* estarán y las mismas condiciones y se podrá apreciar las diferencias entre la cobertura que puede ofrecer cada una de las compañías telefónicas. Gracias a las herramientas de nuestra aplicación *software* podemos apreciar de forma rápida y gráfica los diferentes servicios. Una vez realizado esto se podrán sacar conclusiones y escoger la compañía que mejor cobertura telefónica aporte. El análisis se realizara teniendo en cuenta el nivel de la señal y el tipo de servicio dado.
- Establecer los límites de funcionamiento de nuestro sistema: Mediante estas pruebas podremos probar el funcionamiento completo del sistema y sacar conclusiones. Por ejemplo, podremos establecer el número máximo de usuarios que pueden estar conectados a cada *router* estableciendo un ancho de banda mínimo para cada uno de ellos. De esta forma conseguiremos



establecer las limitaciones de cada dispositivo y ver cuántos dispositivos harían falta por vagón.

- Implementación de mejoras a partir de los resultados de las pruebas de campo: A medida que se vayan desarrollando diferentes pruebas de campo, irán surgiendo diferentes necesidades y mejoras que implementar en nuestro sistema y en nuestra aplicación. Por ejemplo a raíz de la primera prueba de campo, se implementó un sistema que graficaba en tiempo real la evolución del nivel de cobertura. Gracias a esto se consiguió apreciar mejor determinadas zonas y caídas de cobertura a lo largo del trayecto.

## 4.2 PRIMERA PRUEBA.

La primera prueba se realizó a los dos meses de iniciar el proyecto. Con esta prueba se buscaba una primera toma de contacto con el sistema. En ella se usó la versión más sencilla de la aplicación *software* y solo se contó con un *router*. Se realizó un trayecto corto de una hora de duración. Se partió desde el Parque Tecnológico de Boecillo con destino Cuellar. Durante el trayecto se buscaron zonas de baja cobertura y se comprobó que el *router* detecta que no hay cobertura y simplemente espera a que la señal aumente.

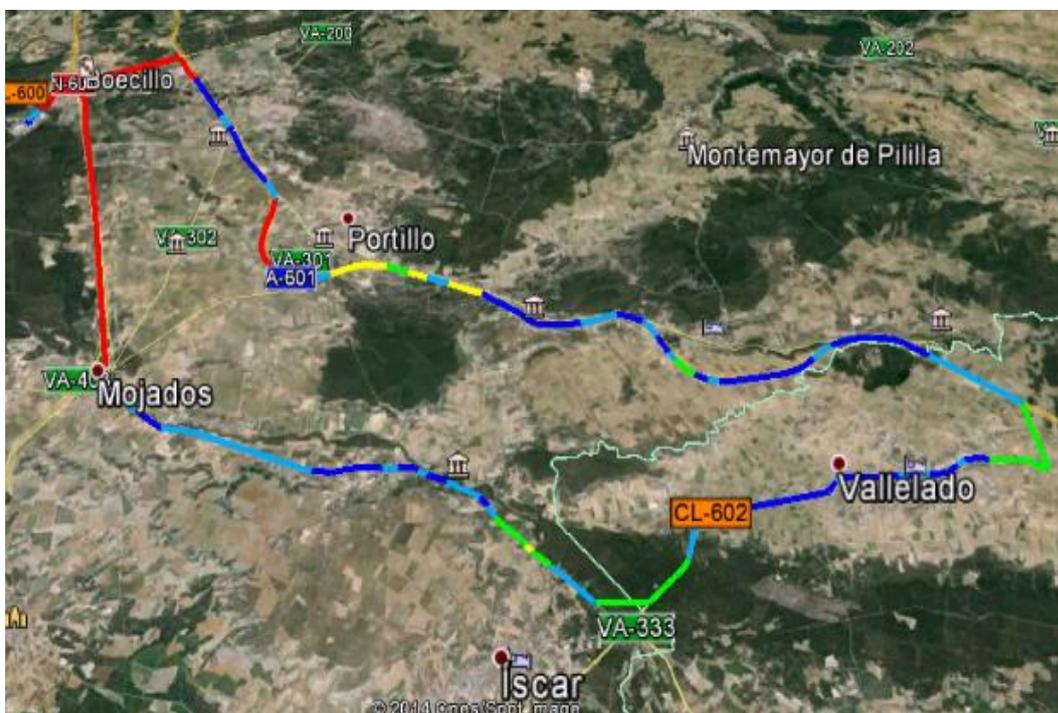


Figura 4.2. Detalle de la ruta realizada en la primera prueba de campo

Como se puede apreciar en la figura anterior, hay determinadas zonas del trayecto que están en rojo, lo que significa que no hay cobertura. En general se puede ver como los niveles de cobertura en son bastante buenos en casi todo el trayecto.

## 4.3 SEGUNDA PRUEBA.

Con la segunda prueba se preparó una simulación con todo el sistema completo y la aplicación en su versión definitiva. Anterior a esta prueba, se realizaron pequeñas pruebas intermedias en las que se comprobaba que las nuevas implementaciones y mejoras en la aplicación funcionaban de forma correcta.

En esta prueba se busca simular el funcionamiento del sistema en un tren de alta velocidad. Para ello se usaron los tres dispositivos con dos tarjetas *SIM* cada una. Durante la prueba se desconectaba de forma aleatoria un *router*. Con esto se pretendía simular el comportamiento del sistema cuando alguno de los *routers* se quedaba sin cobertura.

El trayecto realizado fue similar al de la primera prueba, pasando por zonas sin cobertura y con el vehículo de pruebas a la máxima velocidad permitida para así analizar los niveles de cobertura frente a la velocidad del vehículo.

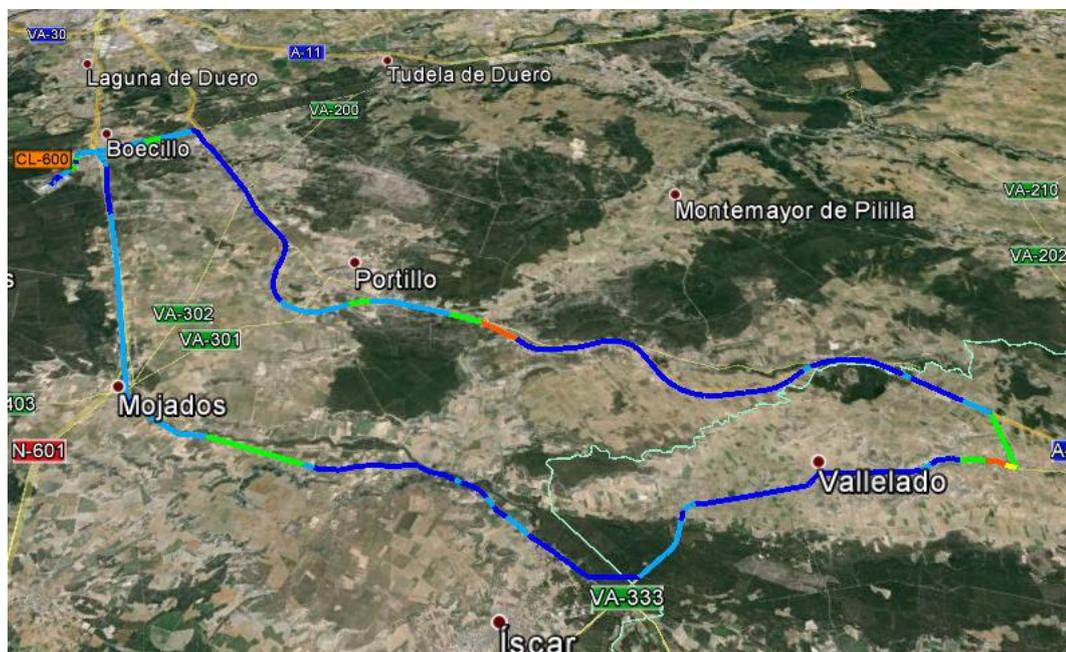


Figura 4.3. Ejemplo de la ruta realizada en la segunda prueba de campo



## 4.4 Resultados y conclusiones.

Una vez realizadas las pruebas de campo y procesado los resultados obtenidos se pueden presentar unas primeras conclusiones acerca del comportamiento de nuestro sistema. A continuación explicaremos los resultados obtenidos en cada prueba de campo en comparación con los objetivos planteados.

### RESULTADOS DE LA PRIMERA PRUEBA DE CAMPO.

En esta primera prueba se contaba con la primera versión de la aplicación *software* y con un solo *router* por lo que algunos objetivos no se pudieron lograr. Al ser una primera toma de contacto con el sistema en una situación real el recorrido planteado para esta primera prueba de campo buscaba dar lugar al máximo número de situaciones diferentes: comportamiento en lugares con cobertura 4G, comportamiento sin cobertura, comportamiento a gran velocidad, etc.

Respecto al primer objetivo, analizar el comportamiento del sistema en determinadas situaciones, destacamos cuatro diferentes casos:

- Zona con cobertura 4G en las que se pudiera medir el máximo ancho de banda que podrían gestionar nuestros dispositivos. Pudimos apreciar una buena respuesta por parte de estos ya que se llegó a medir una velocidad de descarga de 50 Mbps y 21,3 Mbps de velocidad de subida. Comparando con los datos proporcionados por la compañía, comprobamos que estos resultados corresponden a los máximos que se pueden alcanzar.
- Zona sin cobertura para poder comprobar como reacciona nuestro sistema y el tiempo que tarda en reconectarse. Cuando se salió de la zona de cobertura se pudo apreciar que el dispositivo tardaba poco en buscar y recuperar la señal.
- Comportamiento ante alta velocidad (100 km/h) para observar la relación entre la señal de cobertura y la velocidad del vehículo y poder comprobar que el sistema funciona de la misma forma, estando parado que estando en movimiento a gran velocidad.



Al disponer de un solo dispositivo *router*, no se pudo comparar diferentes compañías telefónicas en igualdad de condiciones.

Durante esta prueba pudimos analizar y establecer los límites de funcionamiento de nuestro sistema y así poder dimensionar un sistema completo para un tren de alta velocidad. Se llegó a comprobar que con tres *routers* se podría dar servicio a un vagón completo.

Una vez finalizada la primera prueba de campo, se pudieron sacar una serie de conclusiones para la implementación de mejoras a partir de los resultados obtenidos.

- En primer lugar se implementaron varias mejoras en la aplicación para solucionar diferentes fallos que no permitían a la aplicación trabajar correctamente.
- Una vez visto que la aplicación funcionaba correctamente para un solo dispositivo, se implementó un sistema capaz de trabajar con los tres *routers* diferentes. De esta forma se obtendrá la información correspondiente de cada dispositivo de forma individual. Se generará un fichero *.kml* diferente para cada *router* con la misma ruta pero los niveles de cobertura correspondientes a cada dispositivo.
- Por último se implementó una mejora que permitía al usuario ver la evolución de la cobertura a lo largo del recorrido de forma gráfica de los tres dispositivos. De esta forma se podrían observar diferencias entre las diferentes compañías durante el propio trayecto y analizar situaciones que serían complicadas de analizar a posteriori con los datos obtenidos.

### RESULTADOS DE LA SEGUNDA PRUEBA DE CAMPO.

En esta segunda prueba de campo, se buscaba una prueba definitiva en la cual la aplicación *software* estaba en una etapa muy madura. El trayecto realizado en esta segunda prueba fue notablemente más largo. El objetivo principal era encontrar situaciones en las cuales se pudiera diferenciar bien



entre las diferentes compañías (cada uno de los tres *routers* tenía una compañía telefónica diferente).

En esta prueba se buscaba analizar el comportamiento del sistema en determinadas situaciones en las que se pudiera apreciar el comportamiento de las diferentes compañías. Las principales situaciones fueron las siguientes:

- Zonas de baja cobertura en la cual se apreciara que compañía ofrece mejores servicios y poder analizar estos datos gráficamente.
- Zonas de alta cobertura con servicio 4G y poder analizar gráficamente que compañía ofrece un mejor ancho de banda.

Como se ha dicho con anterioridad esta segunda prueba estaba enfocada principalmente a comparar diferentes compañías telefónicas en igualdad de condiciones, para así poder determinar la compañía que mejor servicio puede ofrecer en un determinado trayecto. Gracias a las cuatro gráficas de cobertura que tiene nuestra aplicación *software* pudimos realizar durante el trayecto un análisis exhaustivo de comportamiento de cada una de las compañías telefónicas.

Una vez analizadas estas dos pruebas de campo de forma individual se pueden obtener las siguientes conclusiones:

- El sistema depende de las instalaciones de telefonía que existan en el trayecto a realizar. Cuantas más antenas repetidoras de telefonía móvil existan en el trayecto mejor servicio se podrá ofrecer.
- El sistema no se ve afectado por la velocidad. Durante las pruebas se ha podido apreciar que el funcionamiento del sistema ha sido el mismo cuando el vehículo de pruebas estaba parado que cuando estaba a máxima velocidad.
- El ancho de banda aportado por tres *routers* es suficiente para cubrir un vagón al completo. El sistema es capaz de asegurar un ancho de banda para cincuenta usuarios aproximadamente.



- Cada compañía telefónica se comporta de una forma diferente dependiendo del servicio que se esté usando (3G, 4G). A la hora de implementar este sistema en un tren de alta velocidad habría que analizar cada ruta y elegir la o las compañías que mejor servicio puedan ofrecer.

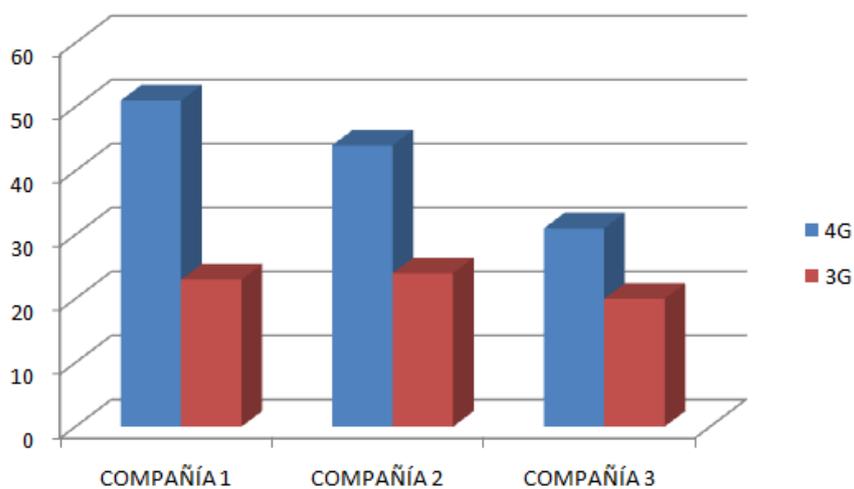


Figura 4.4 Ejemplo de servicios ofrecidos por tres compañías telefónicas diferentes.



Universidad de Valladolid

# Diseño de Sistemas WIFI en Trenes de Alta Velocidad





## 5 ESTUDIO ECONÓMICO

En este capítulo se realiza un estudio económico completo del sistema desarrollado.

### 5.1 COSTES DIRECTOS

Se trata de los costes que son directamente imputables al producto. Se dividen en costes de personal y costes de material.

#### 5.1.1 Costes de personal

Se supone que el proyecto ha sido desarrollado en una empresa de ingeniería, en este caso Centro Tecnológico *CARTIF*. En la realización de proyecto intervienen un ingeniero sénior, que será el director del proyecto y un ingeniero junior o técnico que realizará el proyecto.

- Director del proyecto: Su labor será plantear las directrices del proyecto, dirigir el trabajo, corregir y resolver las dudas.
- Técnico: Analiza los aspectos fundamentales del proyecto planteando las soluciones oportunas.

Para calcular los costes de personal, se multiplica el coste/hora de cada trabajador que interviene en el proyecto por el número total de horas trabajadas. Para ello se divide el proyecto en varias etapas, especificando en cada una, las horas dedicadas por cada miembro del personal.

Podemos dividir el proyecto en las siguientes etapas:

1. Análisis y observación del problema planteado.
2. Búsqueda bibliográfica y documentación.
3. Análisis de sistemas y dispositivos que se adaptan al problema.
4. Diseño detallado de la solución seleccionada (prototipo).
5. Puesta en marcha del prototipo realizado.
6. Diseño y realización de experimentos con el sistema.



### 7. Redacción y escritura de la memoria.

	ETAPA							TOTAL
	1	2	3	4	5	6	7	
<b>DIRECTOR</b>	2	1	6	16	8	3	6	<b>42</b>
<b>TÉCNICO</b>	8	16	60	120	120	120	56	<b>500</b>

Tabla 5.1. Horas imputadas a cada miembro durante el proyecto

Para el cálculo de las horas efectivas anuales vamos a coger la Resolución de 9 de octubre de 2013, de la Dirección General de Trabajo, por la que se registra y publica el XVII Convenio colectivo nacional de empresas de ingeniería y oficinas de estudios técnicos. Esta resolución aparece en el BOE del 25/10/2013 y establece el cómputo máximo de horas de trabajo de 1800 horas. El coste horario y semanal de cada trabajador queda reflejado en la siguiente tabla.

Costes del equipo de trabajo		
Concepto	Director	Técnico
<b>Sueldo Bruto</b>	36.000	22.000
<b>Seguridad Social (35%)</b>	12.600	7.700
<b>Total</b>	48.600	29.700
<b>Costes hora (€/hora)</b>	<b>27</b>	<b>16,5</b>

Tabla 5.2. Coste/hora de cada trabajador del equipo

Por lo tanto, los costes totales de personal serán:



Trabajador	Horas Trabajadas	Coste horario	Coste total (€)
Director	42	27	1.134
Técnico	500	16,5	8.250
<b>Total</b>			<b>9.384</b>

Tabla 5.3. Coste total del personal del proyecto

**COSTES DIRECTOS DE PERSONAL ..... 9.384 €**

### 5.1.2 Costes de material

Dentro de los costes de material se consideran los costes de material por compra de un equipo informático con licencias para el proyecto y dispositivos para la realización y montaje del sistema completo.

<b>HARDWARE</b>			
Concepto	Cantidad	Coste Unit (€)	Coste (€)
Router NB3700-LW-G	1	2.329	2.329
Router NB3700-L-G	2	1.985	3.970
Accesorios kit inicial NetModule	1	350	350
Antena 2G/3G/4G LPMM-7-27	2	145,75	291,50
Cable RF240 Hembra-Macho (10m)	3	42,80	128,40
Adaptador SMA-TNC (Hembra-Macho)	6	5,87	35,22
Portes (Alemania/Nacional)	-	-	130
Fuente de Alimentación	1	57,87	57,87
<b>TOTAL HARDWARE</b>			<b>7.291,99</b>

Tabla 5.4. Hardware para la realización del proyecto



SOFTWARE	
Concepto	Coste
Microsoft Visual Studio 2013 Pro con MSDN	1.064,88
Microsoft® Office 2010 Professional en español	387,01
<b>TOTAL SOFTWARE</b>	<b>1.451,89</b>

Tabla 5.5. Software para la realización del proyecto

**COSTES DIRECTOS DE MATERIAL = ..... 7.291,99 + 1.451,89 = 8.743,88 €**

## 5.2 COSTES INDIRECTOS

Dentro de estos costes se refleja el consumo de energía eléctrica, teléfono, ADSL, gastos de desplazamiento y gastos de alquiler. Se considera que el periodo de ejecución del proyecto es de 5 meses y que en la misma oficina técnica se realizan varios proyectos simultáneamente.

- Consumo de energía eléctrica: 500 €.
- Alquiler: 1000 €.
- Teléfonos e internet: 500 €.
- Material oficina y otros: 200 €.

**TOTAL ..... 2200 €/mes.**

Si consideramos que realizamos un factor de utilización de 0,20 de los recursos para este proyecto, entonces:

- $COSTES = 2.200€ / mes \times 5 meses \times 0,2 = 2.200€$
- Costes imputables solo al proyecto (desplazamientos de pruebas, etc..) : 200 €.

**COSTES INDIRECTOS ..... 2.400 €**



## 5.3 COSTES TOTALES

Los costes totales del proyecto se resumen en la siguiente tabla:

<b>COSTES TOTALES</b>	
COSTES DIRECTOS DE PERSONAL	9.384 €
COSTES DIRECTOS DE MATERIAL	8.743,88 €
COSTES INDIRECTOS	2.400 €
<b>COSTE TOTAL DEL PROYECTO</b>	<b>20.527,88 €</b>

Tabla 5.6. Desglose de los costes del proyecto

A este coste se le aplica un beneficio industrial de un 20%.

- COSTE TOTAL DEL PROYECTO: 20.527,88 €.
- BENEFICIO INDUSTRIAL (20%): 4.105,58 €.

**IMPORTE TOTAL DEL PROYECTO (SIN IVA) ..... 29.806,50 €**



Universidad de Valladolid

# Diseño de Sistemas WIFI en Trenes de Alta Velocidad





## 6 CONCLUSIONES Y LINEAS FUTURAS DE TRABAJO

En este capítulo se quieren plasmar los resultados y conclusiones acerca de este proyecto asociado al trabajo fin de grado. Estas conclusiones representaran un resumen de todo lo conseguido con el proyecto. De la misma forma se plantearan unas líneas futuras de de trabajo y mejora del proyecto que por diversos motivos no se han podido realizar pero si se han planteado a lo largo del tiempo que se ha realizado el proyecto.

### 6.1 CONCLUSIONES

Las conclusiones más importantes que se pueden sacar de este proyecto, se centran en torno a la capacidad y respuesta de los *routers*, los cuales han sido elegidos para encontrar una solución al problema que se proponía inicialmente.

En primer lugar, la iniciativa de elegir dispositivos que pudieran ofrecer el servicio de WIFI (Wireless) a través de la tecnología 3G/4G durante todo el trayecto para un tren de alta velocidad como una tecnología de bajo coste, se cumple, ya que la inversión realizada frente al uso de otro tipo de tecnologías es mucho menor, debido a que los proveedores de 3G y 4G vienen realizando una fuerte apuesta por esta tecnología desde hace años.

Por otro lado, durante el desarrollo del proyecto, surge la duda de que utilizar tres dispositivos para ofrecer WIFI a todos los usuarios del tren de alta velocidad no es viable, ya que de forma general, los trenes tanto de alta velocidad como de media distancia, están formados por vagones separados entre sí y construidos en su totalidad por metal.

Como bien se sabe, el metal es considerado como uno de los materiales que más influencia negativa presenta en cuanto al WIFI se refiere, debido a que propicia grandes interferencias en la señal que transmite el WIFI. Por lo que la utilización de tres dispositivos en un tren completo no cumpliría los objetivos de forma satisfactoria.

Por lo tanto, lo ideal sería utilizar el sistema propuesto (3 *routers*) para cada uno de los vagones, pudiendo así ofrecer un servicio más que óptimo y



además con capacidad suficiente para que en futuros desarrollos sobre el presente proyecto se pudieran realizar mejoras del propio sistema en otros ámbitos, como ofrecer servicios de TV a la carta, estado del tren en tiempo real, etc.

También y cómo bien se sabe, hay puntos de la geografía dónde el servicio 3G/4G no llega con facilidad, por lo que al tener tres dispositivos diferentes se ha tenido en cuenta una solución.

Esta consiste en utilizar un proveedor de 3G/4G diferente para cada dispositivo, garantizando así, que en cualquier momento del recorrido siempre alguno de los proveedores nos proporcione la cobertura suficiente como para mantener el servicio WIFI a todos los usuarios del tren.

Además, otro de los posibles problemas que se presentaban inicialmente en trenes de alta velocidad sobre el WIFI es la influencia de la velocidad sobre el mismo. Pues bien, se han realizado pruebas de campo a velocidades acorde con nuestras capacidades (utilizando una furgoneta como medio de transporte) y se ha demostrado que la cobertura 3G/4G recibida y por lo tanto la capacidad de ofrecer WIFI a los pasajeros del tren no se ve afectada. Como se puede ver en la siguiente imagen:

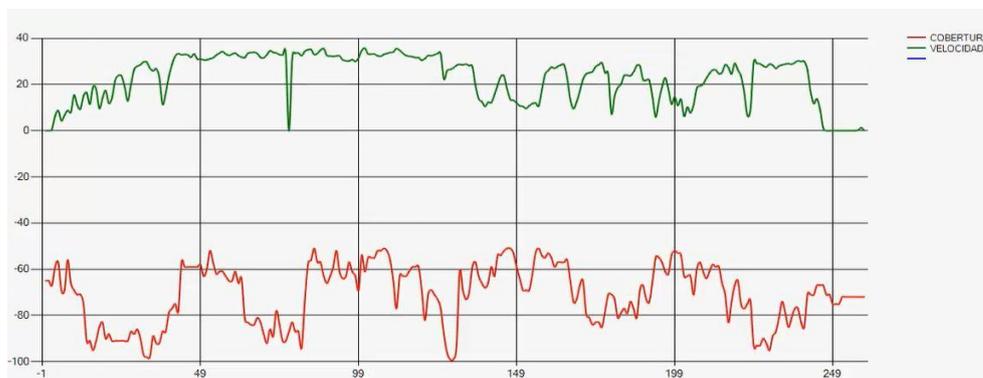


Figura 6.1. Gráfica de la cobertura frente a la velocidad del vehículo

En la parte superior de la gráfica se puede ver la variación de la velocidad en metros por segundo (m/s) durante todo el recorrido y además se puede observar que ha velocidades de 35 m/s (126 km/h aprox.) las coberturas rondan los -50dBm lo que es una cobertura excelente.



Por último y por falta de medios, no se ha podido realizar la prueba en trenes por lo que no se puede saber cómo se comporta el sistema realizado en este tipo de vehículos y con una gran demanda de conexiones al mismo tiempo, aunque se prevé que pueda responder de forma satisfactoria.

Por otro lado se ha constatado que este sistema es totalmente apto para ser implantado en otro tipo de vehículos, como por pueden ser transportes públicos de la índole de autobuses, autocares, etc.

En gran medida, los objetivos que se plasmaban de manera inicial, quedan totalmente cumplidos.

## 6.2 LÍNEAS FUTURAS DE TRABAJO

En cuanto a posibles futuras mejoras en este campo de desarrollo, se puede destacar que es interesante mejorar la aplicación *software* desarrollada para este sistema.

Una de las posibles mejoras que se plantean actualmente es la de configurar los *routers* sin acceder a ellos desde el navegador pidiendo los parámetros necesarios, guardándolos en un archivo de extensión ".*cfg*", la cual determina la configuración de cada dispositivo y subiéndolos posteriormente de forma automática a cada uno de los *router* en formato ".*zip*".

Esta mejorara estaría enfocada a permitir que un usuario configurara el sistema de forma semiautomática sin tener conocimiento de el funcionamiento de este.



```
config.version=1.3
config.product=3700
network.hostname=NB3700
wwan.0.status=1
wwan.0.card=0
wwan.0.sim=0
wwan.0.stype=1
wwan.0.number=*99***1#
wwan.0.username=vodafone
wwan.0.password=[enc]ft1Rmecr1ASlw5zMrJaMA==
wwan.0.apn=ac.vodafone.es
wwan.0.auth=3
autorun.status=0
switch.port.1.interface=1
switch.port.2.interface=2
wlan.0.status=1
wlan.0.mode=access-point
wlan.0.regdom=ES
wlan.0.settings.channel=7
wlan.0.vss.0.status=1
wlan.0.vss.0.ssid=cartif2014
wlan.0.vss.0.secmode=wpa-psk
wlan.0.vss.0.psk=[enc]GS09yG7d++aHsvbHkK4loQ==
wlan.0.vss.1.status=1
wlan.0.vss.1.ssid=cartif2014 2
wlan.0.vss.1.secmode=wpa-psk
wlan.0.vss.1.psk=[enc]GS09yG7d++Y9ZHdtPxeZhw==
network.lan.1.mode=wan
network.lan.2.mode=wan
network.lan.3.mode=wan
network.lan.4.mode=
network.wlan.0.mode=routed
network.wlan.1.mode=bridged
network.wan.0.interface=lan1
network.wan.0.mode=dhcp
network.wan.1.interface=lan2
network.wan.1.mode=dhcp
```

Figura 6.2. Ejemplo de un archivo de configuración del *router*

Junto con esta mejora se pretende desarrollar un programa que permita al usuario escoger los valores de las variables que configurarían el *router*. El programa guardaría esta configuración y la subiría al *router*.

Entre los diferentes parámetros se puede ver el estado del WIFI (*WLAN.0.status=1*), el modo, el canal seleccionado, el nombre de la red que se quiere crear (SSID), el modo de seguridad, los modos y tipos de redes que se desean aplicar a cada *router*, el estado del DHCP y una infinidad más de parámetros que es necesario conocer más en profundidad para implementarlos en la aplicación.

Otra de las posibles aplicaciones en el futuro es la de estudiar aún más en profundidad el dispositivo, permitiendo así el poder realizar otras funciones, debido a que el fabricante de los propios dispositivos considera que con una ingeniería compleja se podría realizar una aplicación en la que por ejemplo, se permitiese controlar el tráfico al antojo del usuario. Es decir, que los usuarios que pretenden realizar "**Streaming**" (visualización de videos,



películas, series, descarga continua, etc.) que precisan un mayor ancho de banda fueran controlados por dos de los *routers* que componen el sistema y que los usuarios que pretenden realizar "**Browsing**" (Navegar por páginas web, mensajería instantánea, etc.) que precisan un menor ancho de banda fueran controlados por un único *router*.

En la figura anterior se puede apreciar un autobús dotado de dispositivos audiovisuales individuales que funcionan a través de WIFI. Esto exige que el sistema WIFI del vehículo proporcione un ancho de banda capaz de cubrir las necesidades de todos los dispositivos a la vez.

En esta línea y dado que los dispositivos que componen este sistema son muy potentes y mediante un correcto estudio de los dispositivos, seguramente se podrían realizar otras muchas aplicaciones que actualmente no están a nuestro alcance.



Universidad de Valladolid

# Diseño de Sistemas WIFI en Trenes de Alta Velocidad





## 7 BIBLIOGRAFÍA

A continuación se describe la bibliografía consultada tanto en formato digital (Páginas web, pdf, etc.), como en formato de papel (libros, *datasheet*, etc.).

### SITIOS WEB

- [1] Ibersystems. *Redes WiMax* [en línea] Consulta: Mayo 2015. Disponible en: <http://www.redeswimax.info/>
- [2] Wikipedia. *WiMax* [en línea] Consulta: Mayo 2015. Disponible en: <http://es.wikipedia.org/wiki/WiMAX>
- [3] Ventajas Desventajas. *Ventajas y desventajas de WiMax* [en línea] Consulta: Mayo 2015. Disponible en: <http://www.ventajasdesventajas.com/wimax/>
- [4] Francisco J. Flores. *Ventajas y desventajas de los sistemas WiMax* [en línea] Consulta: Mayo 2015. Disponible en: <http://www.gartel.net/node/1>
- [5] Wikipedia. *Internet por satélite* [en línea] Consulta: Mayo 2015. Disponible en: [http://es.wikipedia.org/wiki/Internet\\_por\\_sat%C3%A9lite](http://es.wikipedia.org/wiki/Internet_por_sat%C3%A9lite)
- [6] Conocimientos Informáticos. *Ventajas y desventajas de banda ancha satelital* [en línea] Consulta: Mayo 2015. Disponible en: <http://ordenador.wingwit.com/Redes/internet-networking/70472.html>
- [7] Nassat Development Satellite Systems. *Internet vía satélite* [en línea] Consulta: Mayo 2015. Disponible en: <http://nassat.com/productos/trenes/>
- [8] Hispasat. *Acceso a Internet* [en línea] Consulta: Mayo 2015. Disponible en: <http://www.hispasat.com/es/soluciones-y-servicios/mercado-operadores/acceso-a-internet>
- [9] Sergio Uceda "Comparaiso". *Qué es Internet por satélite y cuanto cuesta* [en línea] Consulta: Mayo 2015. Disponible en:



<http://www.comparaiso.es/manuales-guias/que-es-internet-satelite-precios>

[10] Rita Conde. *¿Qué significan 1G, 2G, 3G y 4G?* [en línea] Consulta: Mayo 2015. Disponible en:

[http://celulares.about.com/od/Preguntas\\_frecuentes/a/Que-Significan-1g-2g-3g-Y-4g.htm](http://celulares.about.com/od/Preguntas_frecuentes/a/Que-Significan-1g-2g-3g-Y-4g.htm)

[11] Ángel Gutiérrez. *¿Qué es el 3G y 4G?* [en línea] Consulta: Mayo 2015. Disponible en:

<http://Windowsespanol.about.com/od/RedesYDispositivos/a/Internet-Movil.htm>

[12] Google Sites. *Telefonía 3G*. [en línea] Consulta: Mayo 2015. Disponible en:

<https://sites.google.com/site/telefonía3g/Home/hi/marco-teorico/ventajas/desventajas>

[13] Simulador *NETMODULE* [en línea] Consulta: Mayo 2015. Disponible en:

<http://simulator.netmodule.com/>

[14] *GMap.NET - Great Maps for Windows Forms & Presentation* [en línea] Consulta: Mayo 2015. Disponible en:

<https://greatmaps.codeplex.com/>

[15] *GMap.NET Tutorial: Calculating the area of a polygon on the map* [en línea] Consulta: Mayo 2015. Disponible en:

<http://www.websofia.com/?s=GMap.NET>

## **Datasheet**

[16] *Teldat. Teldat H1-Rail* [Datasheet] Consulta: Diciembre 2014.

[17] *Icomera. Icomera X6* [Datasheet] Consulta: Diciembre 2014.

[18] *NETMODULE. NB3700 Mobile & WLAN Router* [Datasheet] Consulta: Diciembre 2014.



## Libros

[19] Ceballos, Francisco Javier. *Enciclopedia de Microsoft Visual C# Interfaces gráficas y aplicaciones para Internet con Windows forms y ASP.NET*. 4ª Edición. Editorial Ra-Ma.

[20] Firtman, Maximiliano y Natale, Leonardo. *Visual Studio .NET Framework 3.5 para profesionales*. Editorial Marcombo SA

[21] Estrada Robles, Jorge. *Diseño de Sistemas WIFI en Trenes de Alta Velocidad*. Proyecto Final de Carrera, Universidad de Valladolid.



Universidad de Valladolid

# Diseño de Sistemas WIFI en Trenes de Alta Velocidad





# ANEXOS



Universidad de Valladolid

# Diseño de Sistemas WIFI en Trenes de Alta Velocidad



## ANEXO A: MANUAL DE CONFIGURACIÓN DEL SISTEMA

2014

### MANUAL DE PUESTA EN MARCHA DEL ROUTER NB3700



Centro Tecnológico CARTIF  
19/09/2014

A continuación vamos a describir los pasos que hay que seguir para la puesta en marcha del sistema desarrollado, el cual, nos permitirá obtener conexión a Internet mediante WIFI en trenes de alta velocidad, utilizando los equipos comerciales elegidos para llegar a cumplir los objetivos propuestos.

Los equipos comerciales elegidos como hemos visto en apartados anteriores, está formado por tres *routers* de la compañía *NETMODULE*, concretamente un *NB3700-LW-G* y dos *NB3700-L-G*, los cuales, y el propio proveedor nos lo indica están desarrollados para ser utilizados en vehículos ferroviarios.

Inicialmente y para la correcta puesta en marcha del dispositivo, debemos tener en cuenta que se cumplen las condiciones ambientales, las cuales son:

- Evitar la exposición directa a la radiación solar.
- Proteger el dispositivo de humedad, vapores y líquidos agresivos.
- Garantizar la circulación de aire alrededor del dispositivo.
- Garantizar que el dispositivo es utilizado únicamente en interiores.
- Garantizar que la altitud máxima a la que va a funcionar el dispositivo son 4000 metros.



La configuración inicial hay que hacerla por separado para cada dispositivo, para ello, previamente y como ya se ha mencionado anteriormente (ver 5.1.3. Otras Conexiones) ya se habían dejado libres los conectores "Ethernet 1" de cada dispositivo.

En primer lugar se va a configurar el dispositivo principal, es decir, el *router* que actuará como maestro y será el encargado de gestionar el sistema completo.

Para ello hay que realizar en todos los *routers* un acceso inicial o también

denominado primer acceso. Este acceso se consigue conectado el adaptador *M12* a RJ45 del puerto *Ethernet 1* al puerto *Ethernet* del PC.

Las conexiones físicas que debe de tener el dispositivo (*router principal*) en este momento son:

- Alimentación a 24 V<sub>DC</sub>.



- Cable *Ethernet M12* a RJ45 al *Ethernet 1*.



- Insertar dos tarjetas *SIM* (o 1 al menos) que proporcionarán cobertura *3G/4G* en las ranuras de la parte frontal del dispositivo. Y se asegurará que estas quedan perfectamente posicionadas (Ayudándose de un clip o similar para introducirlas).

Se puede observar que estas ranuras están designadas con las etiquetas *SIM1* y *SIM2*, para posteriormente realizar la configuración correcta.

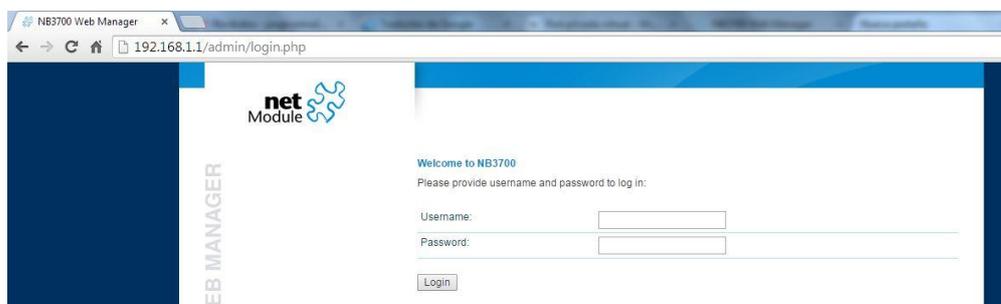


- Instalar la antena *GPS* y la antena celular (solo en Mob1), ya que estos módulos únicamente poseen un módem y solamente usan una de las tarjetas *SIM* en cada instante de tiempo.
- Instalar las dos antenas *WIFI* en los conectores denominados *WLAN1* y *WLAN2*. Teniendo en cuenta que si solamente se utiliza una única antena *WIFI*, se debe conectar al conector *WLAN1*. El sistema funcionará de la misma manera si se tiene una única antena *WIFI* como si se dispone de dos. Pero si se instalan dos antenas *WIFI*, se obtendrá un mejor rendimiento y una mejor cobertura de la señal.



Una vez se ha realizado la conexión, y siempre cerciorándose de que el DHCP está activo (que por defecto lo está), se abre una página de un navegador web (Internet Explorer, Mozilla Firefox, *Google Chrome*, etc.) y se introduce la dirección IP del dispositivo que por defecto es **192.168.1.1**.

De forma instantánea se accede al *Web Manager* de *NETMODULE* para la configuración del dispositivo tal y como se muestra en la siguiente figura:



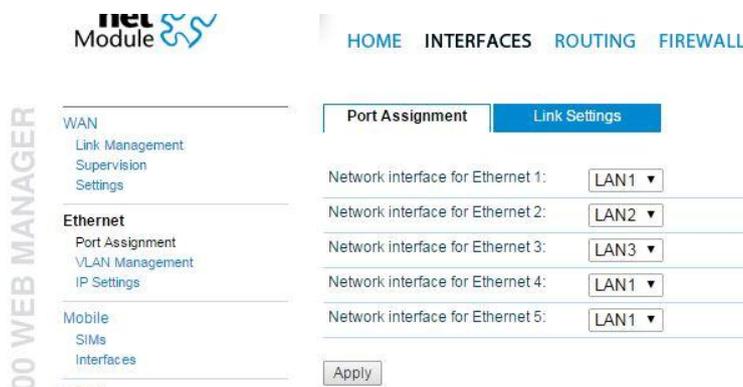
Cuando el *router* está en estado de fábrica, al introducir la dirección anterior, este pedirá un nombre de usuario y una contraseña de administrador. Esta contraseña deberá contener mínimo 6 caracteres, de los cuales al menos 2 deben de ser números y 2 deben ser letras, para garantizar una seguridad intermedia de acceso al dispositivo.

Es importante conocer a partir de ese momento, el usuario y la contraseña elegidos, ya que a partir de ese momento, serán necesarios para acceder al *router* en cualquier otro momento. En el caso que deseemos cambiar este usuario y contraseña de administrador debemos de reiniciar el *router* al modo de fábrica. (\*\*\*) Para realizar el reseteo de fábrica es necesario apretar el botón de reset, durante al menos 5 segundos).

Una vez realizados estos pasos se accede a la configuración del *router* principal y para que este pueda funcionar de la forma esperada habrá que realizar la siguiente configuración:

- *Interfaces / Ethernet / Port Assignment:*

En este submenú se prepara al dispositivo para las futuras conexiones con los otros *routers* y dado que este es el *router* principal habrá que configurarlo como se muestra en la imagen siguiente:



De esta manera conseguimos que por el puerto *Ethernet 2* se crea una conexión *LAN 2*, la cual permitirá conectar en un futuro al *router* principal con el *router 2* o uno de los *routers* secundarios y del mismo modo conseguimos que por el puerto *Ethernet 3* se crea una conexión *LAN 3*, la cual permitirá conectar al dispositivo principal con el otro *router* secundario o también denominado *router 3*. Es decir, se crea una conexión en paralelo de los tres dispositivos.

- *Interfaces / Ethernet / IP Settings:*

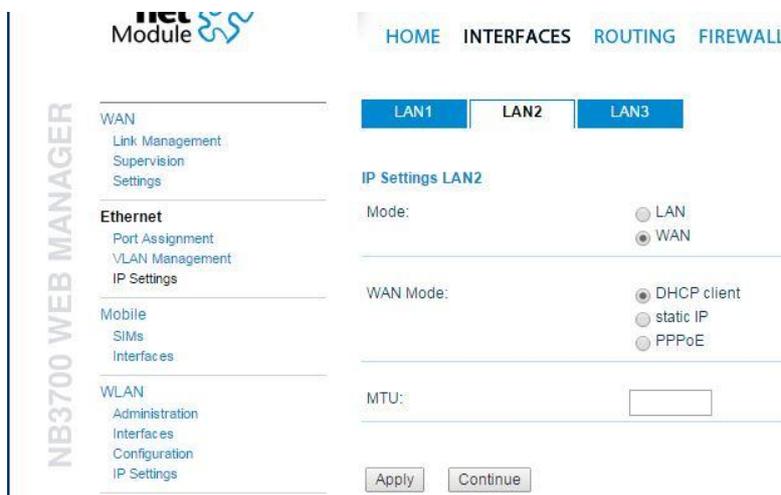


Se configura la red creada anteriormente *LAN 1* en modo *LAN* de la siguiente manera:



The screenshot shows the 'net Module' web interface. The left sidebar is labeled 'NB3700 WEB MANAGER' and contains categories: WAN, Ethernet, Mobile, WLAN, and USB. The main content area has tabs for HOME, INTERFACES, ROUTING, FIREWALL, and VPN. Under 'INTERFACES', there are sub-tabs for LAN1, LAN2, and LAN3. The 'IP Settings LAN1' section shows 'Mode' set to LAN. The 'Static Configuration' section has input fields for IP address (192.168.1.1) and Subnet mask (255.255.255.0). There are also fields for Alias IP address and Alias subnet mask.

Y las redes *LAN 2* y *LAN 3* también creadas anteriormente en modo *WAN* y en modo **DHCP client**.



The screenshot shows the 'net Module' web interface. The left sidebar is labeled 'NB3700 WEB MANAGER'. The main content area has tabs for HOME, INTERFACES, ROUTING, and FIREWALL. Under 'INTERFACES', there are sub-tabs for LAN1, LAN2, and LAN3. The 'IP Settings LAN2' section shows 'Mode' set to WAN. The 'WAN Mode' section has radio buttons for DHCP client (selected), static IP, and PPPoE. There is an input field for MTU. At the bottom, there are 'Apply' and 'Continue' buttons.

De esta manera los dispositivos que se conecten a los puertos *Ethernet 2* y *Ethernet 3* mediante las redes *LAN 2* y *LAN 3* respectivamente serán clientes del dispositivo principal mediante DHCP.

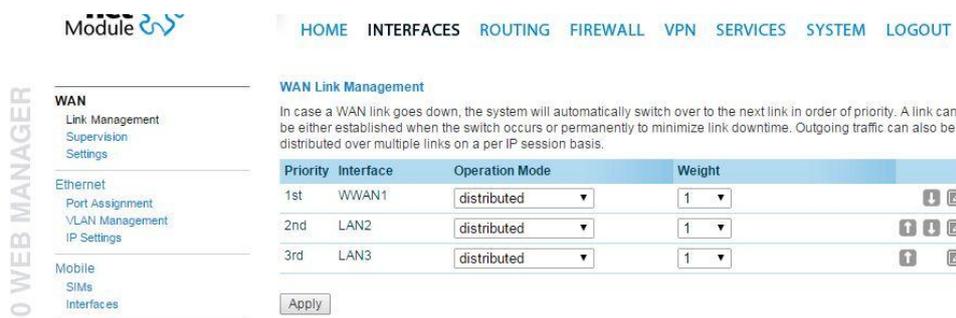
Una vez se han realizado los pasos anterior, el dispositivo ha creado tres redes diferentes. *WAN1* que es la propia red creada para el propio *router* principal, y las redes *LAN 2* y *LAN 3* las cuales serán las



encargadas de conectarse a los *routers* secundarios 2 y 3 respectivamente.

- *Interfaces / WAN / Link Management:*

Aquí, se pueden ver las tres redes que se acaban de describir en el párrafo anterior y se debe determinar el nivel de prioridad de cada red y el modo de operación de cada una. Como se quiere que las tres redes funcionen de igual manera y distribuir el tráfico de datos entre los tres *routers* de forma uniforme de manera inicial, *NETMODULE* recomienda usar el modo "**distributed**" para las tres redes, dotándolas de un peso igual para tratar de repartir el flujo de datos de manera uniforme.



The screenshot shows the 'WAN Link Management' configuration page in the NETMODULE web interface. The page has a navigation menu at the top with options: HOME, INTERFACES, ROUTING, FIREWALL, VPN, SERVICES, SYSTEM, and LOGOUT. On the left, there is a sidebar menu for '10 WEB MANAGER' with categories: WAN (Link Management, Supervision, Settings), Ethernet (Port Assignment, VLAN Management, IP Settings), and Mobile (SIMs, Interfaces). The main content area is titled 'WAN Link Management' and contains a descriptive paragraph: 'In case a WAN link goes down, the system will automatically switch over to the next link in order of priority. A link can be either established when the switch occurs or permanently to minimize link downtime. Outgoing traffic can also be distributed over multiple links on a per IP session basis.' Below this is a table with the following data:

Priority	Interface	Operation Mode	Weight
1st	WWAN1	distributed	1
2nd	LAN2	distributed	1
3rd	LAN3	distributed	1

At the bottom of the table, there is an 'Apply' button.

Ahora se procede a configurar correctamente las tarjetas *SIM*, las cuales normalmente se configuran de forma automática.

- *Interfaces / Mobile / SIMs:*

Una vez se introducen la/las tarjetas *SIM* en las ranuras correspondientes, el sistema, de manera automática las detecta y se puede ver la siguiente pantalla:



Module 

HOME INTERFACES ROUTING FIREWALL VPN SERVICES SYSTEM LOGOUT

**0 WEB MANAGER**

- WAN
  - Link Management
  - Supervision
  - Settings
- Ethernet
  - Port Assignment
  - VLAN Management
  - IP Settings
- Mobile
  - SIMs
  - Interfaces

**SIM Cards**

This menu can be used to assign a default modem to each SIM which will also be used by SMS and GSM voice services. A SIM card can get switched in case of multiple WWAN interfaces sharing the same modem.

SIM	Default	Current	State	PIN Protection	Registered	
SIM1	Mobile1	Mobile1	ready	disabled	yes	
SIM2	none	none	unassigned	unknown	no	

La tarjeta *SIM* introducida en la ranura correspondiente a *SIM1* está en estado "**ready**", lo que significa que está operativa y disponible para ser usada.

Por otro lado en el caso de tener insertada una tarjeta adicional en la ranura correspondiente a *SIM2*, está también estaría asociada al módem denominado *Mobile1* (se recuerda que solo dispone de un módem) pero el estado sería de "**unassigned**", esto es debido a que solamente una de las tarjetas es utilizada en un instante de tiempo dado.

En el caso de que no se esté de acuerdo con la configuración que el dispositivo ha realizado de forma automática, esta se puede cambiar pulsando en el icono en forma de lápiz de la derecha.



En este caso, tanto para la tarjeta *SIM1*, como para la tarjeta *SIM2* se configurarían a voluntad del usuario.

Una vez se han aplicado todos los cambios anteriores seleccionando siempre el botón "Apply", y accediendo al submenú *Mobile / Interfaces* se podrán ver las redes *WWAN* asociadas a cada tarjeta.

WWAN Interfaces

Interface	Modem	SIM	Number	Service	APN / User		
WWAN1	Mobile1	SIM1	*99***1#	3G-only	ac.vodafone.es / vodafone		
WWAN2	Mobile1	SIM2	*99***1#	Automatic	ac.vodafone.es / vodafone		

Siempre se recomienda usar el tipo de servicio "**Automatic**", para que cada tarjeta pueda ofrecernos siempre la mejor tecnología disponible. No obstante, se podrá cambiar haciendo click en el icono en forma de lápiz de la derecha.

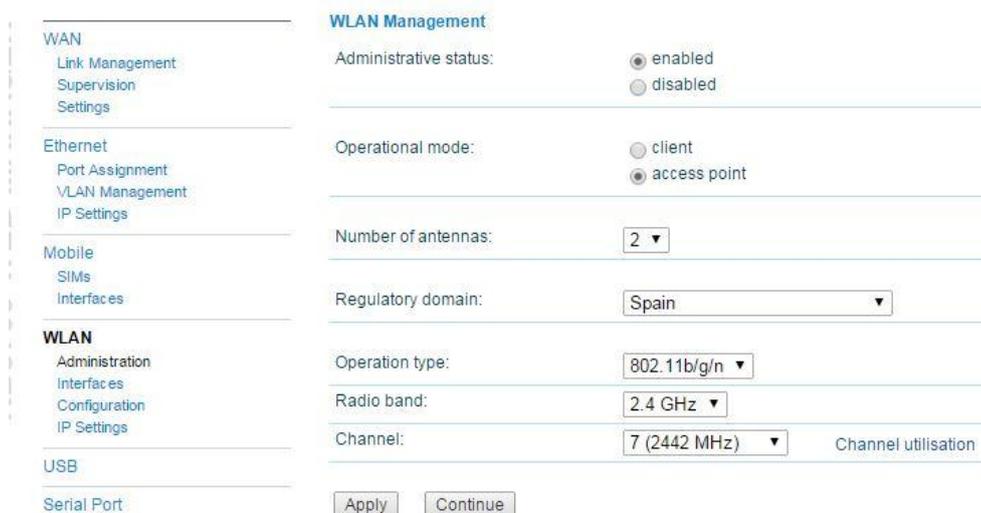



The screenshot shows the 'IO WEB MANAGER' interface. On the left is a vertical navigation menu with categories: WAN (Link Management, Supervision, Settings), Ethernet (Port Assignment, VLAN Management, IP Settings), Mobile (SIMs, Interfaces), WLAN (Administration, Interfaces, Configuration, IP Settings), USB, and Serial Port. The main content area is titled 'Edit WWAN Interface WWAN1' and has three tabs: 'Mobile', 'Connection', and 'Advanced'. The 'Mobile' tab is active. It contains the following fields: 'Modem:' with a dropdown menu set to 'Mobile1'; 'SIM:' with a dropdown menu set to 'SIM1'; and 'Service type:' with a dropdown menu set to 'Automatic'. There is an 'Apply' button at the bottom.

Realizando todos los pasos anteriores, se tiene un dispositivo preparado para administrar los *routers* secundarios que se conectarán a él y las tarjetas *SIM* disponibles para ofrecer servicio el servicio de Internet. Únicamente queda habilitar la propiedad que tiene este *router* para proporcionar WIFI a los usuarios.

- *Interfaces / WLAN / Administration:*

En este submenú se habilitará el servicio de proporcionar WIFI a través de las tarjetas *SIM* configuradas anteriormente. Y se realiza de la siguiente manera:



The screenshot shows the 'IO WEB MANAGER' interface with the 'WLAN Management' configuration page. The left navigation menu is the same as in the previous screenshot. The main content area is titled 'WLAN Management' and contains the following fields: 'Administrative status:' with radio buttons for 'enabled' (selected) and 'disabled'; 'Operational mode:' with radio buttons for 'client' and 'access point' (selected); 'Number of antennas:' with a dropdown menu set to '2'; 'Regulatory domain:' with a dropdown menu set to 'Spain'; 'Operation type:' with a dropdown menu set to '802.11b/g/n'; 'Radio band:' with a dropdown menu set to '2.4 GHz'; and 'Channel:' with a dropdown menu set to '7 (2442 MHz)' and a 'Channel utilisation' label. There are 'Apply' and 'Continue' buttons at the bottom.

Se habilita la opción de proporcionar WIFI, marcando "enabled" en el estado del sistema, utilizando como modo de operación "access point",



de esta forma el sistema será el punto de acceso a Internet de los dispositivos que se conectan a él.

Es conveniente seleccionar el número de antenas que se conectan al *router*, en este caso se utilizan 2, porque se ha querido un rendimiento máximo.

El estándar que se utiliza es el 802.11n el cual proporciona flujos de datos (150 Mbit/s) y rangos de distancia mayores (70 m en interiores y 250 m en exteriores) que el resto.

- *Interfaces / WLAN / Interfaces:*

En este submenú se da un nombre a la red WIFI, es decir, se proporciona el SSID (Service Set Identifier), el cual queda asociado a la red creada con anterioridad.



- *Interfaces / WLAN / Configuration:*

Se configura el modo de seguridad para la red WIFI creada anteriormente, utilizando el método de seguridad WPA-PSK, el cual permite asociar una contraseña para tener acceso a la red. Esta última se inserta en Passphrase:



**WAN**

Link Management

Supervision

Settings

---

**Ethernet**

Port Assignment

VLAN Management

IP Settings

---

**Mobile**

SIMs

Interfaces

---

**WLAN**

Administration

Interfaces

Configuration

IP Settings

---

**USB**

### WLAN1 Configuration

SSID:

---

Security mode:

---

WPA/WPA2 mixed mode:

---

WPA cipher:

---

Passphrase:

---

Security features:

hide SSID

isolate clients

---

- *Interfaces / WLAN / IP Settings:*

Se selecciona el modo "**routed**", ya que se desea proporcionar una nueva puerta de enlace para el WIFI, en este caso 192.168.200.1:

**WAN**

Link Management

Supervision

Settings

---

**Ethernet**

Port Assignment

VLAN Management

IP Settings

---

**Mobile**

SIMs

Interfaces

---

**WLAN**

Administration

Interfaces

Configuration

IP Settings

### WLAN1 IP Settings

Network mode:

bridged

routed

---

IP address:

---

Subnet mask:

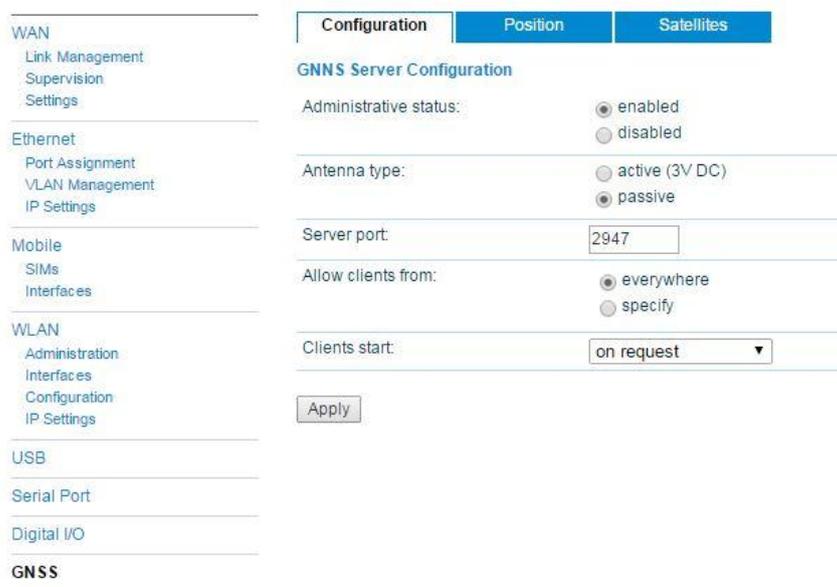
---

El *router* principal ya es capaz de proporcionar WIFI a los usuarios que se conecten a él, únicamente queda la opción de activar el *GPS* para que en un futuro se puedan realizar pruebas de monitoreo de posición del *router* y la respectiva posición del tren.



- *Interfaces / GNSS:*

Lo activamos de la siguiente manera:

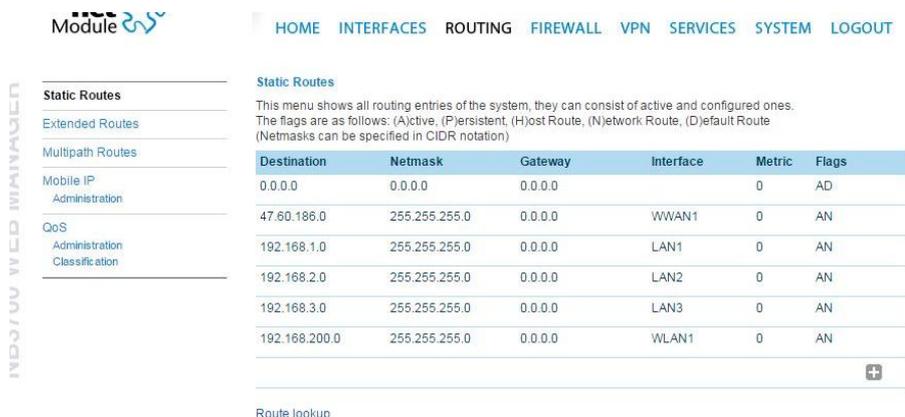


Configuration	Position	Satellites
<b>GNSS Server Configuration</b>		
Administrative status:	<input checked="" type="radio"/> enabled	<input type="radio"/> disabled
Antenna type:	<input type="radio"/> active (3V DC)	<input checked="" type="radio"/> passive
Server port:	<input type="text" value="2947"/>	
Allow clients from:	<input checked="" type="radio"/> everywhere	<input type="radio"/> specify
Clients start:	<input type="text" value="on request"/>	
<input type="button" value="Apply"/>		

El *router* en este instante ya proporciona WIFI por sí mismo a través de la tecnología *3G/4G* y únicamente es necesario configurar unos pequeños detalles para que las redes se puedan distribuir de forma correcta entre todos los *routers*. Para ello habrá que realizar los siguientes pasos:

- *Routing / Static Routes:*

Se deberá comprobar que el sistema quede de forma similar a la mostrada en la siguiente imagen:

Module 

HOME INTERFACES ROUTING FIREWALL VPN SERVICES SYSTEM LOGOUT

**Static Routes**

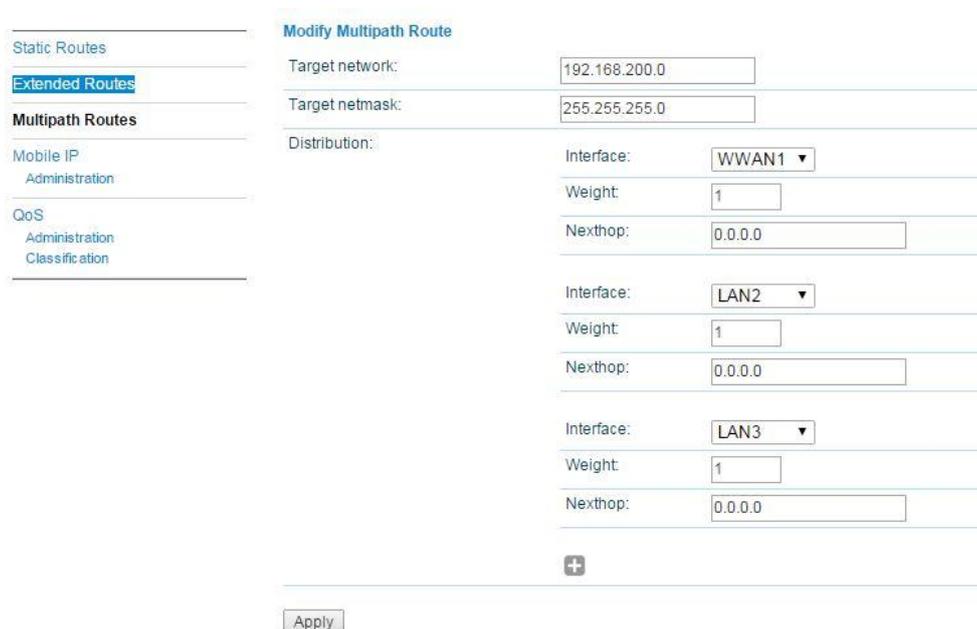
This menu shows all routing entries of the system, they can consist of active and configured ones. The flags are as follows: (A)ctive, (P)ersistent, (H)ost Route, (N)etwork Route, (D)efault Route (Netmasks can be specified in CIDR notation)

Destination	Netmask	Gateway	Interface	Metric	Flags
0.0.0.0	0.0.0.0	0.0.0.0		0	AD
47.60.186.0	255.255.255.0	0.0.0.0	WWAN1	0	AN
192.168.1.0	255.255.255.0	0.0.0.0	LAN1	0	AN
192.168.2.0	255.255.255.0	0.0.0.0	LAN2	0	AN
192.168.3.0	255.255.255.0	0.0.0.0	LAN3	0	AN
192.168.200.0	255.255.255.0	0.0.0.0	WLAN1	0	AN

Route lookup

- **Routing / Multipath Routes:**

Aparecerá una ventana en blanco y a la derecha un icono "+", haciendo click en él se abrirá la siguiente ventana que debemos de configurar de la misma forma:



**Modify Multipath Route**

Target network:

Target netmask:

Distribution:

Interface:

Weight:

Nextthop:

Interface:

Weight:

Nextthop:

Interface:

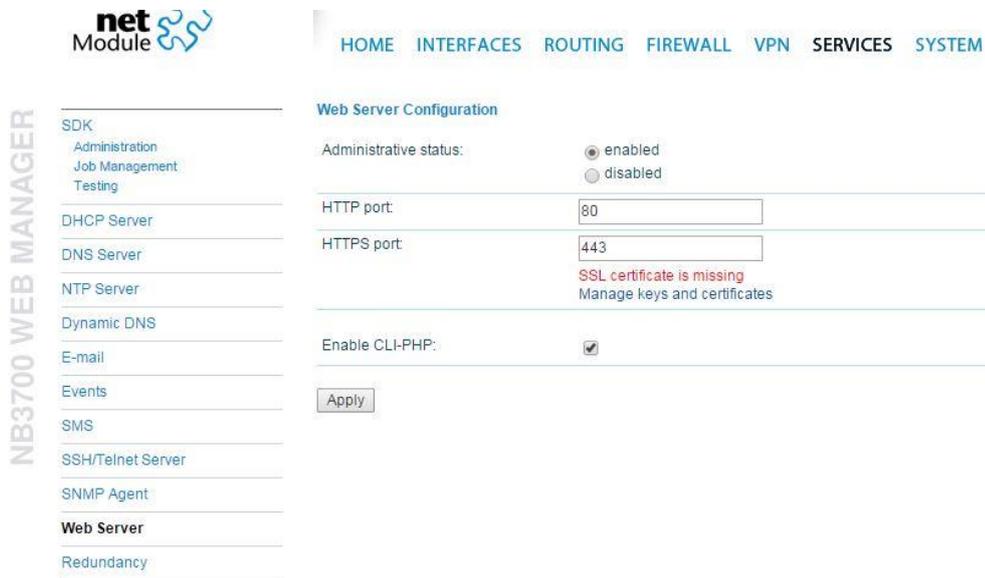
Weight:

Nextthop:

Apply

Por último debemos de habilitar una opción para que podamos acceder al *router* a través de peticiones *PHP*, lo cual es necesario ya que como veremos posteriormente se ha elaborado un *software* de monitorización de posición, coberturas y extracción de datos durante un trayecto del vehículo en el que se instalarán los *routers*.

Para ello, activamos la pestaña *Services / Web Server / Enable CLI-PHP*.



Aplicamos los cambios realizados y el *router* principal ya está preparado para cumplir con los objetivos del proyecto.

De forma similar a como se realizó la configuración del *router* principal hay que configurar el *router* secundario 2.

Las conexiones físicas que debe de tener el dispositivo (*router* secundario 2) en este momento son:

- Alimentación a 24 V<sub>DC</sub>.



- Cable *Ethernet M12* a RJ45 al *Ethernet 1*.



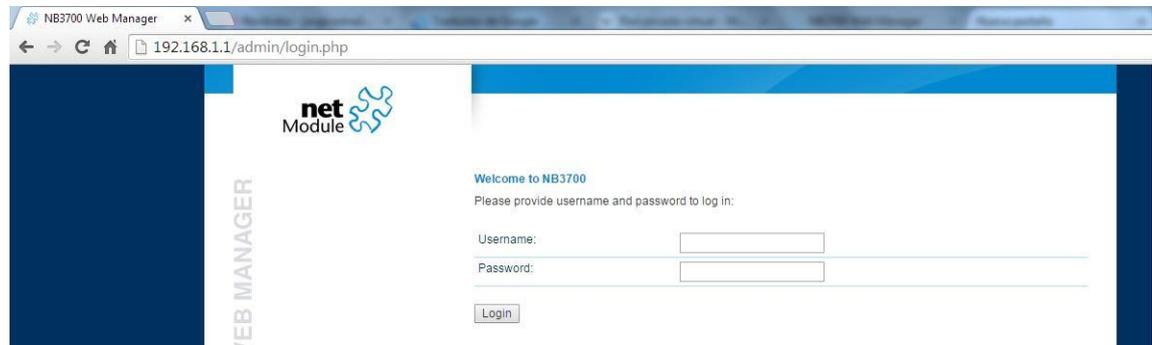
- Insertar dos tarjetas *SIM* (o 1 al menos) que proporcionarán cobertura *3G/4G* en las ranuras de la parte frontal del dispositivo.



- Instalar la antena celular (solo en Mob1), ya que estos módulos únicamente poseen un módem y solamente usan una de las tarjetas *SIM* en cada instante de tiempo.
- Este módulo secundario no dispone de la capacidad de ofrecer WIFI por lo que los conectores *WLAN 1* y *WLAN 2* presentes en *router* principal en este caso están "*limitados*".
- Por último, el conector del *GPS* le dejamos al aire, ya que los tres dispositivos estarán situados en el mismo tren, por lo que la posición será la misma que para el *router* principal.

Una vez se ha realizado la conexión, y de nuevo, cerciorándose de que el DHCP está activo (que por defecto lo está), se abre una página de un navegador web (Internet Explorer, Mozilla Firefox, *Google Chrome*, etc.) y se introduce la dirección IP del dispositivo que por defecto es **192.168.1.1**.

De forma instantánea y de igual manera que en el caso del *router* principal, se accede al *Web Manager* de *NETMODULE* para la configuración del dispositivo tal y como se muestra en la siguiente figura:



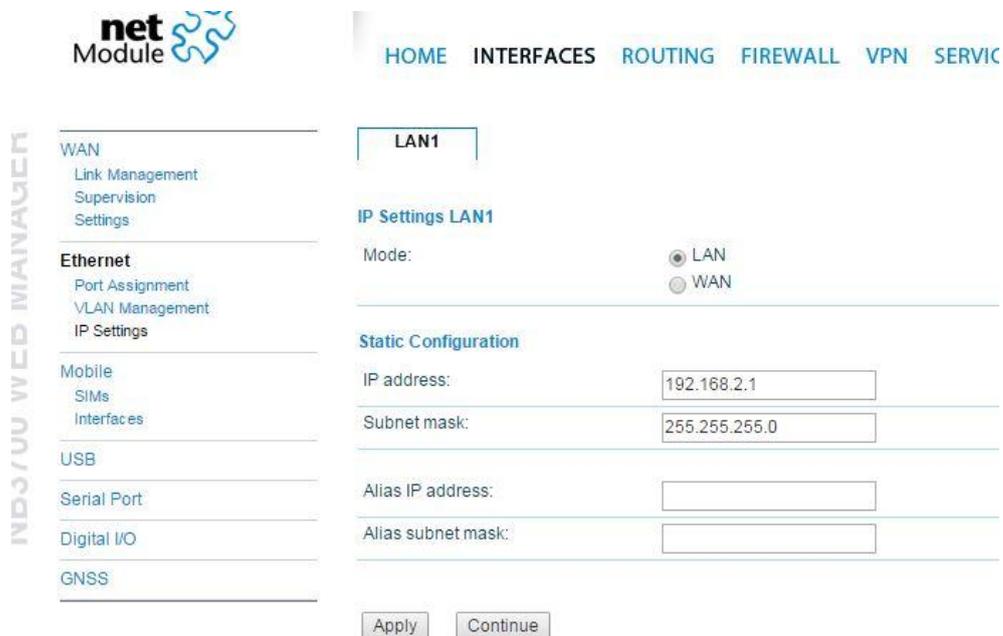
Cuando el *router* está en estado de fábrica, al introducir la dirección anterior, este pedirá un nombre de usuario y una contraseña de administrador. Esta contraseña deberá contener mínimo 6 caracteres, de los cuales al menos 2 deben de ser números y 2 deben ser letras, para garantizar una seguridad intermedia de acceso al dispositivo.

Es importante conocer a partir de ese momento, el usuario y la contraseña elegidos, ya que a partir de ese momento, serán necesarios para acceder al *router* en cualquier otro momento. En el caso que deseemos cambiar este usuario y contraseña de administrador debemos de reiniciar el *router* al modo de fábrica. (\*\*\*) Para realizar el reseteo de fábrica es necesario apretar el botón de reset, durante al menos 5 segundos).

Una vez realizados estos pasos se accede a la configuración del *router* secundario 2 y para que este pueda funcionar de la forma esperada habrá que realizar la siguiente configuración:

- *Interfaces / Ethernet / IP Settings:*

Para que el *router* secundario 2 quede enlazado de forma correcta al *router* principal es necesario que su dirección IP sea 192.168.2.1, por lo tanto se debe cambiar de la siguiente manera:



The screenshot shows the netModule web management interface. On the left is a vertical sidebar with the text 'ND3/00 WEB MANAGER' and a list of menu items: WAN, Ethernet, Mobile, USB, Serial Port, Digital I/O, and GNSS. The main content area has a top navigation bar with 'HOME', 'INTERFACES', 'ROUTING', 'FIREWALL', 'VPN', and 'SERVIC'. Below this, the 'LAN1' configuration page is displayed. It includes a 'Mode' section with radio buttons for 'LAN' (selected) and 'WAN'. A 'Static Configuration' section contains input fields for 'IP address' (192.168.2.1), 'Subnet mask' (255.255.255.0), 'Alias IP address', and 'Alias subnet mask'. At the bottom are 'Apply' and 'Continue' buttons.

Por otro lado, la configuración de las tarjetas *SIM* se hará de forma similar a la realizada para el *router* principal desde el menú *Interfaces / Mobile*.

Por último debemos de habilitar la opción para que podamos acceder al *router* a través de peticiones *PHP*, igual que se ha realizado para el caso del *router* principal y para los mismos fines.

Para ello, activamos la pestaña *Services / Web Server / Enable CLI-PHP*.



Aplicamos los cambios realizados y el *router* secundario 2 también está preparado para cumplir con los objetivos del proyecto.

De forma similar a como se realizó la configuración del *router* principal y en el *router* secundario 2, hay que configurar el *router* secundario 3.

Las conexiones físicas que debe de tener el dispositivo (*router* secundario 3) en este momento son:

- Alimentación a 24 V<sub>DC</sub>.



- Cable *Ethernet M12* a RJ45 al *Ethernet 1*.



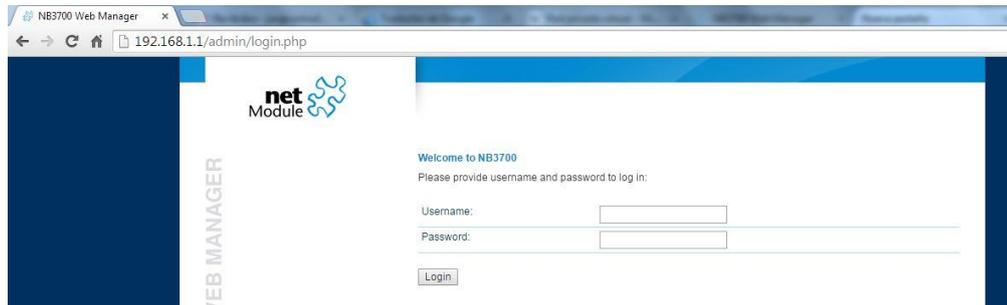
- Insertar dos tarjetas *SIM* (o 1 al menos) que proporcionarán cobertura *3G/4G* en las ranuras de la parte frontal del dispositivo.



- Instalar la antena celular (solo en Mob1), ya que estos módulos únicamente poseen un módem y solamente usan una de las tarjetas *SIM* en cada instante de tiempo.
- Este módulo secundario no dispone de la capacidad de ofrecer WIFI por lo que los conectores *WLAN 1* y *WLAN 2* presentes en *router* principal en este caso están "*limitados*".
- Por último, el conector del *GPS* le dejamos al aire, ya que los tres dispositivos estarán situados en el mismo tren, por lo que la posición será la misma que para el *router* principal.

Una vez se ha realizado la conexión, y de nuevo, cerciorándose de que el DHCP está activo (que por defecto lo está), se abre una página de un navegador web (Internet Explorer, Mozilla Firefox, *Google Chrome*, etc.) y se introduce la dirección IP del dispositivo que por defecto es **192.168.1.1**.

De forma instantánea y de igual manera que en el caso del *router* principal y el *router* secundario 2., se accede al *Web Manager* de *NETMODULE* para la configuración del dispositivo tal y como se muestra en la siguiente figura:



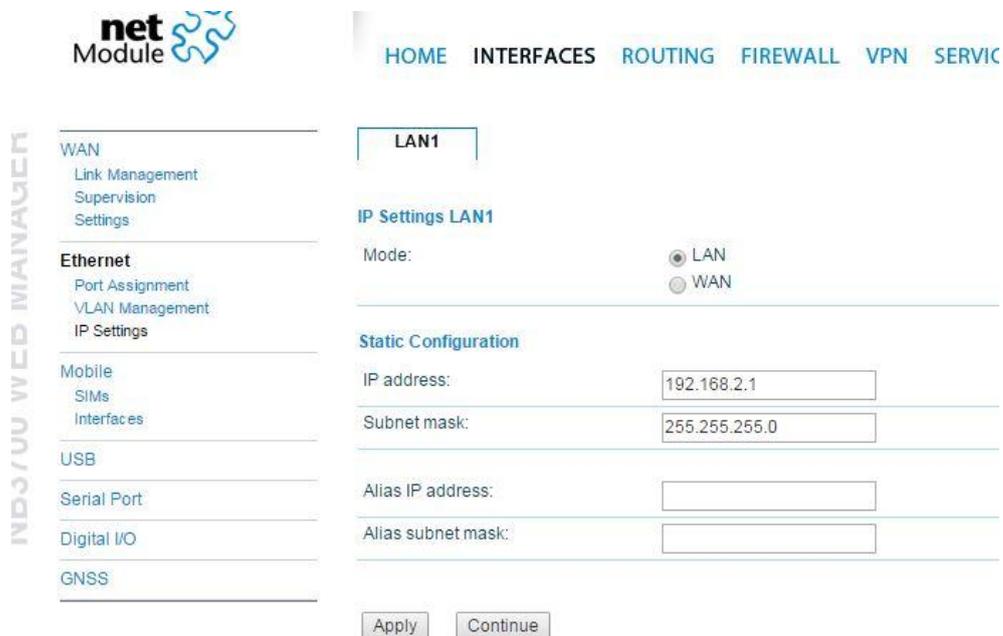
Cuando el *router* está en estado de fábrica, al introducir la dirección anterior, este pedirá un nombre de usuario y una contraseña de administrador. Esta contraseña deberá contener mínimo 6 caracteres, de los cuales al menos 2 deben de ser números y 2 deben ser letras, para garantizar una seguridad intermedia de acceso al dispositivo.

Es importante conocer a partir de ese momento, el usuario y la contraseña elegidos, ya que a partir de ese momento, serán necesarios para acceder al *router* en cualquier otro momento. En el caso que deseemos cambiar este usuario y contraseña de administrador debemos de reiniciar el *router* al modo de fábrica. (\*\*\*) Para realizar el reseteo de fábrica es necesario apretar el botón de reset, durante al menos 5 segundos).

Una vez realizados estos pasos se accede a la configuración del *router* secundario 2 y para que este pueda funcionar de la forma esperada habrá que realizar la siguiente configuración:

- *Interfaces / Ethernet / IP Settings:*

Para que el *router* secundario 3 quede enlazado de forma correcta al *router* principal es necesario que su dirección IP sea 192.168.3.1, por lo tanto se debe cambiar de la siguiente manera:



The screenshot shows the netModule web interface. On the left is a vertical sidebar with the text 'IND3/00 WEB MANAGER' and a list of menu items: WAN, Ethernet, Mobile, USB, Serial Port, Digital I/O, and GNSS. The main content area has a top navigation bar with 'HOME', 'INTERFACES', 'ROUTING', 'FIREWALL', 'VPN', and 'SERVIC'. Below this, the 'LAN1' configuration page is displayed. It includes 'IP Settings LAN1' with a 'Mode' section where 'LAN' is selected. Under 'Static Configuration', there are input fields for 'IP address' (192.168.2.1), 'Subnet mask' (255.255.255.0), 'Alias IP address', and 'Alias subnet mask'. At the bottom are 'Apply' and 'Continue' buttons.

Por otro lado, la configuración de las tarjetas *SIM* se hará de forma similar a la realizada para el *router* principal desde el menú *Interfaces / Mobile*.

Por último debemos de habilitar la opción para que podamos acceder al *router* a través de peticiones *PHP*, igual que se ha realizado para el caso del *router* principal y para los mismos fines.

Para ello, activamos la pestaña *Services / Web Server / Enable CLI-PHP*.



net Module

HOME INTERFACES ROUTING FIREWALL VPN SERVICES SYSTEM

NB3700 WEB MANAGER

SDK  
Administration  
Job Management  
Testing

DHCP Server

DNS Server

NTP Server

Dynamic DNS

E-mail

Events

SMS

SSH/Telnet Server

SNMP Agent

**Web Server**

Redundancy

Web Server Configuration

Administrative status:  enabled  disabled

HTTP port:

HTTPS port:   
SSL certificate is missing  
Manage keys and certificates

Enable CLI-PHP:

Apply

Aplicamos los cambios realizados y el *router* secundario 3 también está preparado para cumplir con los objetivos del proyecto.

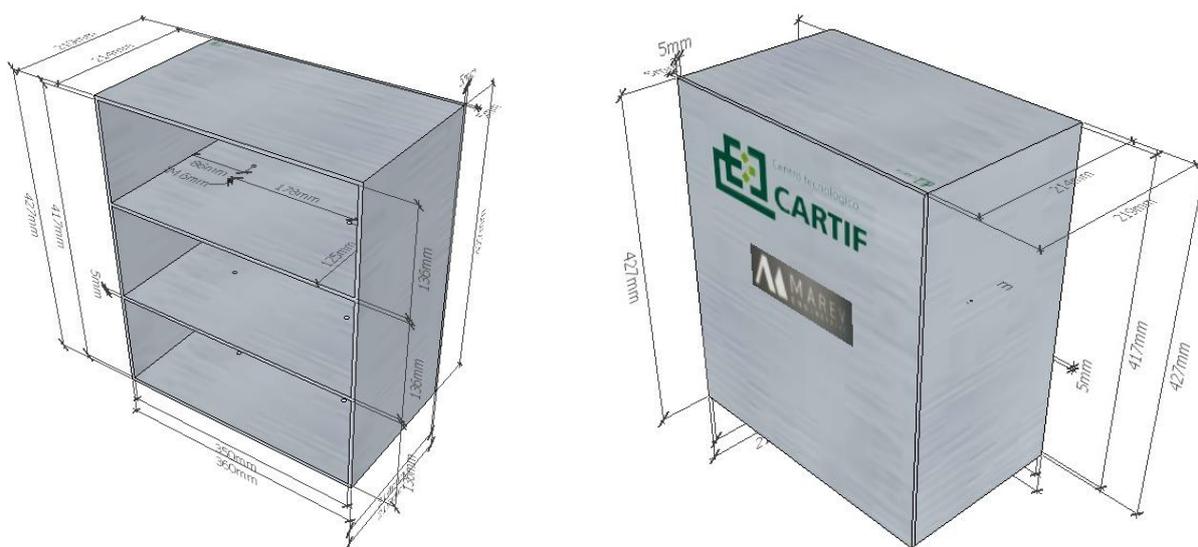
*El sistema ya ha quedado completamente configurado, únicamente queda conectar los puertos Ethernet de cada dispositivo de la forma que se indicó en el capítulo 5.1. CONEXIONADO HARDWARE, quedando la conexión de los puertos Ethernet como la siguiente imagen:*



## ANEXO B: DISEÑO DE LA CAJA SOPORTE PARA EL SISTEMA

Diseño de la caja soporte realizada para instalar los tres *routers* que componen el sistema, con la finalidad de que los dispositivos estén anclados de forma segura durante la duración de todo el trayecto.

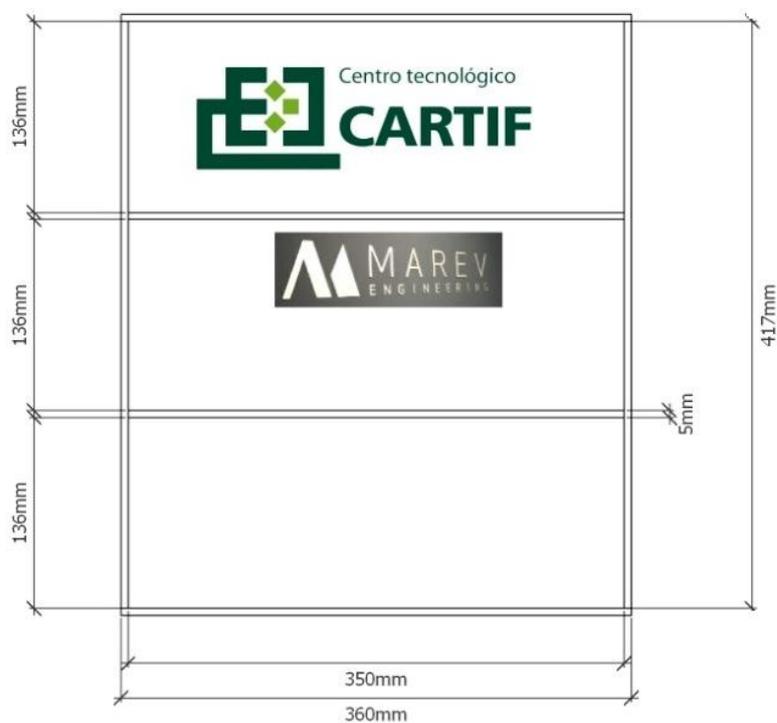
El *software* para el desarrollo del diseño ha sido "SketchUp 2015" y el material utilizado para la fabricación de la caja ha sido aluminio de 5mm.



ANEXO B.1. Vista general de la caja soporte.

## VISTAS MÁS SIGNIFICATIVAS

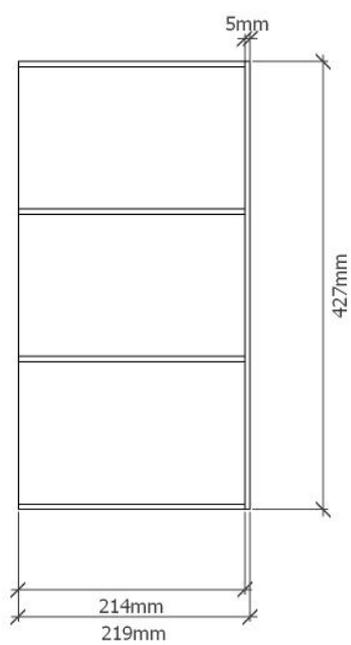
A continuación se detallan las vistas más significativas para la fabricación de la caja soporte del sistema. Las vistas que podemos ver son, el alzado posterior, la planta, la vista lateral izquierda y por último una sección de las baldas donde se colocaran los dispositivos. Esta última consta con las medidas precisas para que los *routers* puedan ser instalados con facilidad.



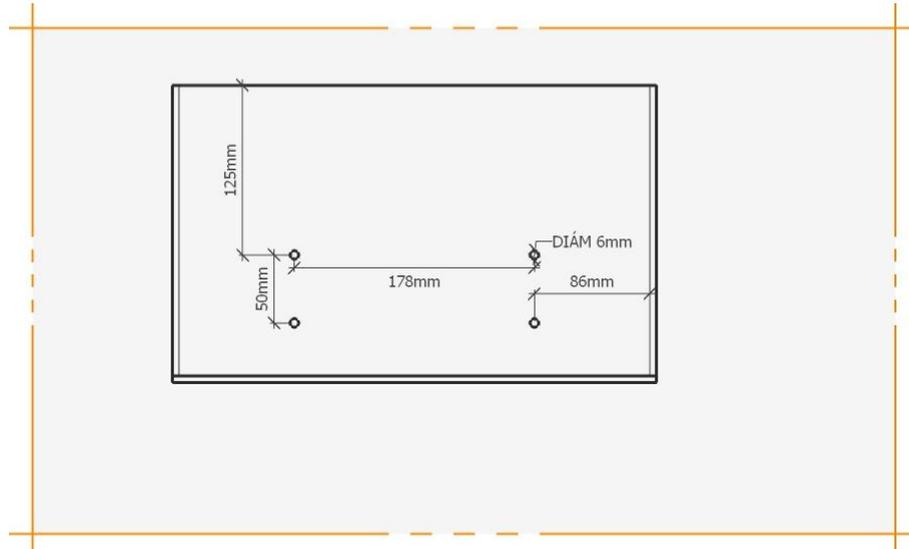
ANEXO B.2. Vista Posterior.



ANEXO B.3. Planta.



ANEXO B.4.Vista lateral izquierda.



ANEXO B.5. Sección de las baldas.