



Trabajo Final de Carrera

**Ingeniería Técnica de Telecomunicación
Especialidad en Sistemas Electrónicos**

Transmisión de Datos PDA-Servidor Web

Jesús Iglesias García

Director: Alex García Quinchia

Departamento de microelectrónica y Sistemas electrónicos

**Escuela Técnica Superior de Ingeniería (ETSE)
Universidad Autónoma de Barcelona (UAB)**

Junio 2010



La persona asignada, Alex García Quinchia, Profesor de la Escuela Técnica Superior de Ingeniería de la Universidad Autónoma de Barcelona (UAB),

CERTIFICA:

Que el trabajo presentado en esta memoria de Trabajo Final de Carrera ha sido realizado bajo su dirección por el alumno Jesús Iglesias García

Y, para que conste a todos los efectos, firman el presente certificado.

Bellaterra, _____

Signatura: *Alex García Quinchia*

ÍNDICE

| | |
|-------------------------------------------------------------|-----------|
| 1. Introducción..... | 8 |
| 2. Descripción de la situación..... | 8 |
| 3. Objetivos..... | 11 |
| 3.1 Objetivo general..... | 11 |
| 3.2 Objetivos específicos..... | 11 |
| 4. Fundamentos teóricos..... | 11 |
| 4.1 La PDA..... | 11 |
| 4.1.1 Conectividad inalámbrica..... | 12 |
| 4.1.1.1 Conexión Wifi..... | 12 |
| 4.1.1.2 Conexión Bluetooth..... | 12 |
| 4.1.2 Sistema Operativo..... | 12 |
| 4.1.3 Plataforma de programación para la PDA..... | 14 |
| 4.2 Servidores web..... | 15 |
| 4.2.1 FTP..... | 15 |
| 4.2.1.1 Servidor FTP..... | 16 |
| 4.2.1.2 Cliente FTP..... | 16 |
| 4.2.1.3 CoreFTP..... | 16 |
| 4.2.2 Entorno de programación de la interfaz web..... | 17 |
| 4.3 Posicionamiento básico: Latitud y Longitud..... | 19 |
| 4.4 Google Maps..... | 19 |
| 4.4.1 API de Google Maps..... | 20 |
| 4.4.1.1 GMap2..... | 20 |
| 4.4.1.2 GLatLng..... | 20 |
| 4.4.1.3 GPolyline..... | 21 |
| 4.4.1.4 GMarker..... | 22 |
| 5. Desarrollo del proyecto..... | 23 |
| 5.1 Desarrollo de la interfaz de usuario para la PDA..... | 23 |
| 5.1.1 Software Transmisión de datos PDA - Servidor FTP..... | 24 |
| 5.1.1.1 Transmisión PDA-FTP Server vía Wifi..... | 24 |
| 5.1.1.2 Implementación del Cliente FTP..... | 25 |
| 5.1.1.3 Configuración de la conexión..... | 26 |
| 5.1.2 Formato de los archivos de ruta..... | 26 |
| 5.1.3 Recepción de datos..... | 28 |
| 5.2 La interfaz web..... | 29 |
| 5.2.1 Estructura del sitio web..... | 30 |
| 5.2.1.1 Árbol de contenidos..... | 32 |
| 5.2.2 Base de datos..... | 33 |
| 5.3 Monitorización de trayectorias..... | 34 |
| 5.3.1 Visualización de rutas en Google Maps..... | 35 |
| 5.3.2 Tipos de rutas..... | 36 |
| 5.3.2.1 Rutas simples..... | 38 |
| 5.3.2.2 Rutas con información de velocidad..... | 38 |
| 5.3.2.3 Rutas de comparación de sistemas..... | 39 |
| 5.3.2.4 Rutas en tiempo real..... | 40 |
| 5.4 Gestión de rutas..... | 41 |
| 5.4.1 Alta de rutas..... | 42 |
| 5.4.2 Modificación de rutas..... | 42 |
| 5.4.3 Baja de rutas..... | 43 |

| | |
|-------------------------------------------------------------------------------------------------------------------|-----------|
| 5.5 Comunicación interfaz web – PDA | 44 |
| 5.6 Testeo y resultados finales..... | 45 |
| 5.6.1 Transmisión de archivos de ruta PDA-Servidor Web | 45 |
| 5.6.2 Visualización de trayectorias en <i>Google Maps</i> | 45 |
| 5.6.3 Visualización de trayectorias en tiempo real | 46 |
| 5.6.4 Transmisión de datos Servidor Web-PDA..... | 47 |
| 5.7 Problemas y soluciones en el desarrollo del proyecto | 47 |
| 6. Planificación del proyecto | 51 |
| 6.1 Diagrama de Gantt..... | 51 |
| 6.2 Fases del proyecto..... | 51 |
| 6.3 Tiempos de realización | 52 |
| 6.4 Requerimientos..... | 53 |
| 6.5 Herramientas de desarrollo | 53 |
| 6.5.1 Microsoft Visual Studio 2005 | 54 |
| 6.5.2 Plataforma de desarrollo web | 54 |
| 7. Conclusiones..... | 55 |
| 8. Futuras ampliaciones | 55 |
| 9. Bibliografía y/o recursos electrónicos | 57 |
| 10. Anexo | 59 |
| 10.1 Código fuente de la aplicación de la PDA | 59 |
| 10.2 Código fuente de la interfaz de usuario / página web | 66 |
| 10.2.1 [newroute.php] Página recepción de rutas y alta en la base de datos... 66 | |
| 10.2.2 [e_trans.php] Alerta sobre el estado de la transmisión en tiempo real .. 68 | |
| 10.2.3 Panel de últimas rutas cargadas..... | 68 |
| 10.2.4 [ruta_simple.php] Visualización de una ruta simple..... | 69 |
| 10.2.5 [ruta_velocidad.php] Visualización de una ruta con información de velocidad | 71 |
| 10.2.6 [distancia_a_ruta.js] Distancia del puntero del ratón a la trayectoria | 76 |
| 10.2.7 [ruta_error.php] Visualización de ruta de comparación de sistemas o error | 80 |
| 10.2.8 [visualización_tiempo_real.php] Visualización de ruta en tiempo real ... 85 | |
| 10.2.9 [alta_ruta.php / alta.php] Gestión de rutas - Altas | 89 |
| 10.2.9.1 alta.php | 91 |
| 10.2.10 [eliminar_ruta.php /eliminar.php] Gestión de rutas - Bajas | 93 |
| 10.2.10.1 eliminar.php..... | 95 |
| 10.2.11 [modificar_ruta.php/modificar.php] Gestión de rutas - Modificaciones. 96 | |
| 10.2.11.1 modificar.php..... | 98 |
| 10.2.12 [busqueda.php] Búsqueda de rutas | 102 |
| 10.2.13 [control.php / s_pda.php] Comunicación con la PDA | 105 |
| 10.2.13.1 s_pda.php | 107 |
| 10.2.14 [acceso_gestion.php/acceso_correcto.php/desconexión.php] Control de acceso a la gestión de rutas | 108 |
| 10.2.14.1 acceso_correcto.php | 108 |
| 10.2.14.2 desconexión.php | 109 |

ÍNDICE DE FIGURAS

| | |
|---------------------------------------------------------------------------------------|----|
| Figura 1: Sistema GPS/INS – PDA | 9 |
| Figura 2: Sistema de transmisión PDA – Servidor web | 10 |
| Figura 3: Entorno de desarrollo Visual Studio 2005 | 14 |
| Figura 4: Cuadro de herramientas y ventana de propiedades de Visual Studio | 15 |
| Figura 5: Cliente FTP CoreFTP | 17 |
| Figura 6: Dreamweaver, vista de código | 18 |
| Figura 7: Aplicación web phpMyAdmin [11] | 18 |
| Figura 8: Representación Latitud, Longitud en el globo terráqueo | 19 |
| Figura 9: Mapa de Google Maps con una superposición de Polyline o ruta | 21 |
| Figura 10: Ejemplo de marcador en un mapa de Google Maps | 22 |
| Figura 11: Esquema general del sistema a implementar | 23 |
| Figura 12: Pantalla inicial y de salida de la interfaz de usuario de la PDA | 24 |
| Figura 13: Formulario o ventana de envío de rutas | 24 |
| Figura 14: Formulario o ventana configuración | 26 |
| Figura 15: Archivo de ruta con información de velocidad | 28 |
| Figura 16: Formulario o ventana de control y recepción de datos | 28 |
| Figura 17: Visualización de ruta en Skyfire | 29 |
| Figura 18: Página web principal del proyecto | 30 |
| Figura 19: Esquema del árbol de contenidos de la página web | 32 |
| Figura 20: Visualización de una trayectoria simple | 35 |
| Figura 21: Marco de visualización de ruta (ruta simple) | 36 |
| Figura 22: Apartado de Rutas de la interfaz web | 37 |
| Figura 23: Mapa con ruta simple | 38 |
| Figura 24: Mapa de ruta con información de la velocidad del vehículo | 39 |
| Figura 25: Mapa con comparación de sistemas de posicionamiento | 40 |
| Figura 26: Visualización de trayectorias en tiempo real | 41 |
| Figura 27: Página de acceso a la gestión de rutas | 41 |
| Figura 28: Página de gestión para cargar nuevas rutas | 42 |
| Figura 29: Página de gestión para modificar rutas | 43 |
| Figura 30: Página de gestión para eliminar rutas | 43 |
| Figura 31: Página de comunicación con la PDA | 44 |
| Figura 32: Comparación visualización de ruta en wikiloc vs. Interfaz web creada | 45 |
| Figura 33: Mapa con comparación de trayectorias | 46 |
| Figura 34: Pestaña de Control de la interfaz de usuario de la PDA | 47 |
| Figura 35: Error del marcador que recorre la trayectoria | 48 |
| Figura 36: Diagrama de Gantt | 51 |
| Figura 37: Aplicación web phpMyAdmin [11] | 54 |

ÍNDICE DE TABLAS

| | |
|--------------------------------------------------|----|
| Tabla 1: Tabla de tipos de ruta | 27 |
| Tabla 2: Tabla pfc de la base de datos | 33 |
| Tabla 3: Tabla admindb de la base de datos | 34 |
| Tabla 4: Tipos de visualización de ruta | 37 |
| Tabla 5: Relación de tareas y tiempos | 52 |

1. Introducción

En este proyecto se visualiza la trayectoria de un vehículo (aéreo o terrestre) en una página web. Para ello se dispone de una PDA (*Personal Digital Assistant*), en la cual se tiene información actualizada de la posición y la velocidad de dicho vehículo. Estos datos son obtenidos de un sistema que combina la navegación inercial y el GPS (*Global Position System*), los cuales estiman de manera precisa la trayectoria del vehículo.

Con el fin de ofrecer una visualización en tiempo real, versátil, accesible y amigable al usuario de la trayectoria del vehículo, se ha desarrollado un sistema de visualización on-line que proporciona un mejor rendimiento en comparación con la que se venía haciendo en la PDA. Para llevarlo a cabo se implementa una interfaz de usuario en la PDA que nos permite transmitir esta información vía WIFI (*Wireless-Fidelity*) a la página web, de igual forma en el servidor web se crea una interfaz que interpreta y gestiona estos datos para posteriormente ser graficados en *Google Maps*.

A lo largo de este documento se detallan todos los aspectos teóricos relacionados con el trabajo, que ayudan a comprender el proceso llevado a cabo.

En la primera parte se sitúa el proyecto en su marco teórico, luego se procede a explicar punto por punto el desarrollo y funcionamiento del sistema a implementar para lograr los objetivos propuestos.

A continuación se presenta el estudio de planificación, donde se detallan todas las fases del proyecto, los recursos necesarios para llevarlo a cabo y el tiempo empleado para cada una de las tareas.

Finalmente se realiza un análisis de los resultados obtenidos y las posibles mejoras que se podrían realizar en un futuro.

2. Descripción de la situación

Este trabajo forma parte de un proyecto de investigación que se desarrolla en el *Departamento de Microelectrónica y Sistemas Electrónicos*, y que tiene como fin el mejorar la navegación que proporciona el tradicional sistema de posicionamiento GPS [1].

En muchas ocasiones la información que visualizamos en el GPS de un vehículo no es del todo exacta, ya que éste proporciona la posición con una precisión entre 15m y 100m. Además, este instrumento presenta algunos otros limitantes como son la lenta frecuencia de actualización, bloqueos de la señal proveniente de los satélites y errores debido al multicamino¹, estos y otros inconvenientes del GPS comprometen la integridad de la navegación.

Con el fin de evitar este tipo de error de precisión, se implementa un sistema de navegación inercial ubicado en el propio vehículo, que es capaz de calcular la velocidad, la posición y la actitud de éste. De esta forma, en los momentos que el GPS no puede proporcionar información precisa, lo hará este nuevo dispositivo, y con el

¹ El multicamino es causado por la reflexión de las señales en superficies próximas al receptor y pueden interferir o producir errores en las señales que llegan directamente desde los satélites al receptor. El error por multicamino es muy difícil de detectar y en ocasiones es imposible de evitar.

conjunto GPS/INS (*Inertial Navigation System*) obtenemos un sistema de posicionamiento más preciso que el basado únicamente en GPS.

El nuevo sistema de navegación que se propone integra los sistemas GPS/INS mediante una FPGA. La integración de estos dos sistemas ofrece una mayor precisión en la navegación que cuando actual independientemente.

Tal y como se muestra en la *Figura 1*, se combinan los datos obtenidos del GPS con los generados por la IMU (*Inertial Measurement Unit*) mediante una serie de algoritmos programados en la FPGA, de tal forma que se acaba generando un paquete de datos producto de la combinación de la información proporcionada por los dos instrumentos de navegación.

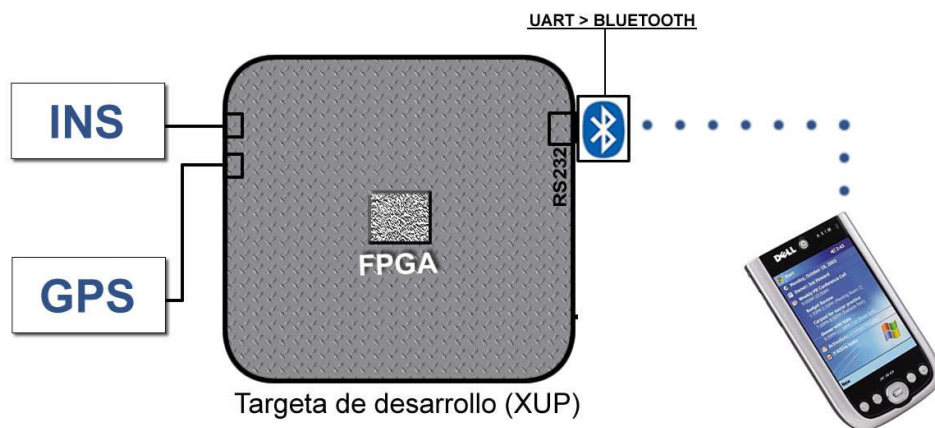


Figura 1: Sistema GPS/INS – PDA

Puesto que se pretende facilitar el seguimiento de la trayectoria del vehículo de manera precisa a partir de los datos obtenidos de esta combinación de instrumentos de navegación, se hace necesaria la visualización de estos datos a través de una interfaz dedicada a ello.

En primer lugar se pensó en crear una interfaz que permitiera la visualización de trayectorias en la propia PDA, realizando la conversión de coordenadas a UTM (*Universal Transversal Mercator*) para graficar sobre el plano, lo que en este caso estaría formado por el conjunto de imágenes correspondientes al mapa en cuestión.

Debido a la complejidad que comporta almacenar y trabajar con todas las imágenes que conforman el mapa sobre el cual se vaya a graficar la trayectoria del vehículo, y a que las limitadas dimensiones de una PDA dificultan la visualización con detalle, se optó por recurrir a otras opciones.

En la actualidad existen multitud de sistemas de información geográfica (SIG) que podríamos emplear con nuestro objetivo, sin ir más lejos, *Google Earth*, un programa informático desarrollado por la empresa *Keyhole Inc.* que permite visualizar imágenes en 3D del planeta, combinando imágenes de satélite, mapas y el motor de búsqueda de *Google* que permite ver imágenes a escala de un lugar específico del planeta. Empleando este software podemos recoger los datos de posicionamiento transmitidos al servidor web y graficar las trayectorias realizadas por un vehículo en dicha aplicación informática.

Ante el gran abanico de posibilidades que se abren con la integración de *Google Earth* en este proyecto, se obvia un detalle no menos importante, pues se está limitando la visualización de las trayectorias a dispositivos informáticos que dispongan de este software previamente instalado. ¿Existe un sistema similar a nivel online? ¿Se podría integrar un sistema de estas características con una aplicación web propia?, y lo que es más importante, ¿Se está ofreciendo un sistema de visualización de trayectorias versátil?

En octubre de 2005 *Google* lanzó una versión web muy similar a *Google Earth* llamada *Google Maps*. Este servidor de aplicaciones de mapas en web permite la integración de mapas y de todas sus características y detalles en páginas web. Además de tratarse de una herramienta potente y en continuo desarrollo, se cuenta gran ventaja de que es una aplicación extendida mundialmente y conocida por la mayoría usuarios de internet. Este sistema cuenta también con el respaldo que proporciona una gran compañía como es *Google*.

Aprovechando el fácil acceso que existe hoy en día a internet, así como su gran versatilidad, se optó por emplear este sistema de mapas para la visualización de la trayectoria del vehículo.

Así pues, este trabajo se centra en el proceso de transmisión de los datos del vehículo (velocidad y posición) a un servidor y en su visualización mediante *Google Maps*. La información relativa al posicionamiento del vehículo se obtiene en una PDA, la cual permite enviar esta información hacia el servidor, donde es interpretada y graficada por una interfaz web. Dicha transmisión se realiza mediante una conexión WIFI, que está disponible en la PDA, aprovechando que es un tipo de conexión cada vez más común en todos los dispositivos informáticos (ordenadores portátiles, *netbooks*, teléfonos móviles...).

Por consiguiente, se propone un sistema que nos permita visualizar la posición y/o trayectoria del vehículo en tiempo real, desde cualquier sitio (con conexión a internet) y en cualquier momento. El esquema del sistema planteado se muestra en la *Figura 2*.

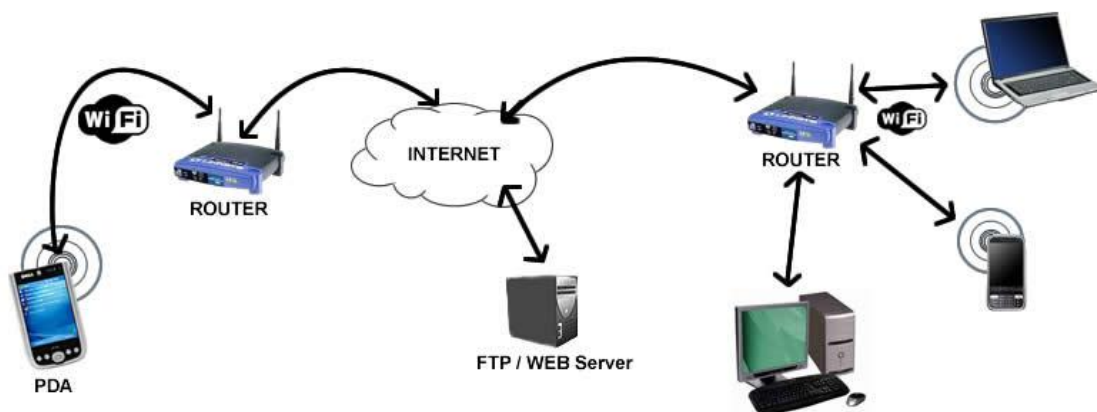


Figura 2: Sistema de transmisión PDA – Servidor web

3. Objetivos

3.1 Objetivo general

- Visualizar la trayectoria de un vehículo en una interfaz web de forma clara y precisa, utilizando la información de posicionamiento que es enviada desde una PDA vía WIFI.

3.2 Objetivos específicos

- Establecer la conexión entre la PDA y el servidor web que permita acceder al servidor FTP, cargar, modificar y eliminar archivos en el servidor.
 - Implementar una interfaz en la PDA que permita gestionar y transmitir datos por WIFI.
 - Establecer una conexión FTP para transmitir y recibir datos del servidor.
 - Almacenar una referencia del archivo en una base de datos, de forma que la aplicación web se entere de la existencia de dicho archivo y se muestre al usuario como una ruta recibida y lista para graficar.
- Al otro lado del sistema se dispone de la interfaz web, la cual debe permitir la lectura y el tratamiento de los archivos transmitidos desde la PDA, y la visualización de las trayectorias que se definen dentro de dichos archivos.
 - La interfaz web debe disponer de los mecanismos necesarios que permitan gestionar los archivos de ruta que se van cargando en el servidor, leerlos e interpretarlos, así como permitir su modificación o eliminación.
 - Graficar las trayectorias que contienen los archivos de ruta en un mapa de *Google Maps*.

Una vez presentados los objetivos de este proyecto se procede a realizar una descripción detallada de todo el proceso así como de los conocimientos teóricos y prácticos empleados.

4. Fundamentos teóricos

A continuación introducimos todos los aspectos teóricos relacionados con el proyecto que nos ayudaran a entender todo su funcionamiento y el proceso realizado por el sistema implementado en este.

4.1 La PDA

Una PDA es un ordenador de mano originalmente diseñado como agenda electrónica (calendario, lista de contactos, bloc de notas y recordatorios) con un sistema de reconocimiento de escritura.

Actualmente estos dispositivos, pueden realizar muchas de las funciones que hace un ordenador de sobremesa (ver películas, crear documentos, juegos, acceder al correo electrónico, navegar por Internet, reproducir archivos de audio, acceder y manipular bases de datos, etc.) pero con la ventaja de ser portátil, y siendo mucho más pequeño que un *notebook* o que los recientemente aparecidos *neetbooks*.

4.1.1 Conectividad inalámbrica

En la actualidad, prácticamente la totalidad de los dispositivos de este tipo que existen en el mercado disponen de conexión Wifi, Bluetooth e infrarrojos, aunque este último empieza a estar en desuso.

4.1.1.1 Conexión Wifi

Wifi (*Wireless Fidelity*) es un conjunto de estándares para redes inalámbricas basado en las especificaciones IEEE 802.11. Wi-Fi no es un acrónimo de "Wireless Fidelity"; Wi-Fi se creó para ser utilizada en redes locales inalámbricas, pero es frecuente que en la actualidad también se utilice para acceder a Internet.

Wi-Fi es una marca de la Wi-Fi Alliance (anteriormente la Wireless Ethernet Compatibility Alliance), la organización comercial que prueba y certifica que los equipos cumplen los estándares IEEE 802.11x.

Este tipo de conexión nos permite acceder a internet o conectarnos con otros dispositivos en cualquier lugar donde exista una red Wifi con velocidades de transmisión comprendidas entre los 11 y 52 Mb/s, y un alcance de entre los 100 y 150 metros.

4.1.1.2 Conexión Bluetooth

La tecnología inalámbrica Bluetooth es una tecnología de ondas de radio de corto alcance (2.4 gigahertzios de frecuencia) cuyo objetivo es el de simplificar las comunicaciones entre dispositivos informáticos, como ordenadores móviles, teléfonos móviles, otros dispositivos de mano y entre estos dispositivos e Internet. También pretende simplificar la sincronización de datos entre los dispositivos y otros ordenadores.

Permite comunicaciones, incluso a través de obstáculos, a distancias de hasta unos 10 metros. Esto significa que, por ejemplo, podemos oír nuestros mp3 desde nuestro comedor, cocina, cuarto de baño, etc. También sirve para crear una conexión a Internet inalámbrica desde un portátil usando un teléfono móvil. Un caso aún más práctico es el poder sincronizar libretas de direcciones, calendarios, etc en una PDA, teléfono móvil, ordenador de sobremesa y portátil automáticamente y al mismo tiempo.

4.1.2 Sistema Operativo

Windows Mobile es un sistema operativo móvil compacto desarrollado por *Microsoft*, y diseñado para su uso en teléfonos inteligentes (*Smartphones*) y otros dispositivos móviles.

Se basa en el núcleo del sistema operativo Windows CE y cuenta con un conjunto de aplicaciones básicas utilizando las API de Microsoft Windows. Está diseñado para ser similar a las versiones de escritorio de Windows estéticamente. Además, existe una

gran oferta de software de terceros disponible para Windows Mobile, la cual se puede adquirir a través de Windows Marketplace for Mobile.

Originalmente apareció como bajo el nombre de *Pocket PC*, como una ramificación de desarrollo de Windows CE para equipos móviles con capacidades limitadas. En la actualidad, la mayoría de los teléfonos con Windows Mobile vienen con un estilete digital, que se utiliza para introducir comandos pulsando en la pantalla. Windows Mobile ha evolucionado y cambiado de nombre varias veces durante su desarrollo, siendo la última versión la llamada Windows Phone 7, anunciada el 15 de febrero del 2010 y sujeta a disponibilidad a finales de 2010.

Para el caso de la PDA que utilizada en este proyecto, se dispone de Windows Mobile 5.0. Esta versión del sistema operativo para dispositivos móviles de Microsoft salió al mercado el 9 de mayo del 2005. Utiliza Windows CE 5.0 y .NET² Compact Framework 1.0 SP2, una plataforma de desarrollo .NET para los programas basados en .NET que utiliza.

Características:

- Una nueva versión de Office llamada "*Office Mobile*".
- Se agregará una versión de Powerpoint denominada "*Powerpoint Mobile*".
- Excel Mobile añade la capacidad de ver representaciones gráficas.
- Word Mobile incluirá la capacidad de insertar tablas y gráficos.
- Reproductor "*Windows Media 10 Mobile*".
- Identificador de llamadas con fotos.
- Un paquete multimedia que facilitará la administración de vídeos y fotos.
- Ayuda mejorada de Bluetooth.
- Interfaz de administración GPS para los programas de navegación instalados.
- Mejoras de la funcionalidad de "*Microsoft Exchange Server*" las mejoras funcionan solamente con Exchange 2003 SP2 instalado.
- Soporte para teclados QWERTY incluido por defecto.
- Simplificación del sistema de informe de errores, como las versiones de Windows de sobremesa y servidores.
- ActiveSync 4.2, prometiendo 10-15% de aumento de la velocidad en la sincronización de datos.
- Cliente para PPTP y L2TP/IPsec VPNs.
- La memoria no volátil (ROM) está disponible en Pocket PC permitiendo un aumento de la batería. Anteriormente más del 50% (suficiente para 72 horas de almacenaje) de energía de la batería se reservaba para mantener datos en la memoria

²El Microsoft .NET Framework, es un componente de software que puede ser o es incluido en los sistemas operativos Microsoft Windows. Provee soluciones pre-codificadas para requerimientos comunes de los programas y gestiona la ejecución de programas escritos específicamente para este framework. Microsoft desea que todas las aplicaciones creadas para la plataforma Windows, sean basadas en el .NET Framework. Su objetivo es crear un marco de desarrollo de software sencillo, reduciendo las vulnerabilidades y aumentando la seguridad de los programas desarrollados.

RAM (volátil). Los dispositivos basados en Windows usa la memoria RAM como su medio de almacenaje primario al uso de memoria flash.

4.1.3 Plataforma de programación para la PDA

Microsoft Visual Studio 2005 es el entorno de diseño y programación que se utiliza para desarrollar la aplicación para la PDA que permite comunicarse con el servidor FTP para transmitir las rutas. Este entorno de trabajo es muy útil porque permite utilizar un emulador de PDA (*Figura 3*) con el mismo sistema operativo que hay actualmente en el lugar de trabajo (*Windows Mobile*) permitiendo hacer múltiples pruebas sin necesidad de conectar la PDA al ordenador.

Visual Studio es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y **Visual Basic .NET**, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET (a partir de la versión net 2002). Así se pueden crear aplicaciones que se intercomunican entre estaciones de trabajo, páginas web y **dispositivos móviles**.

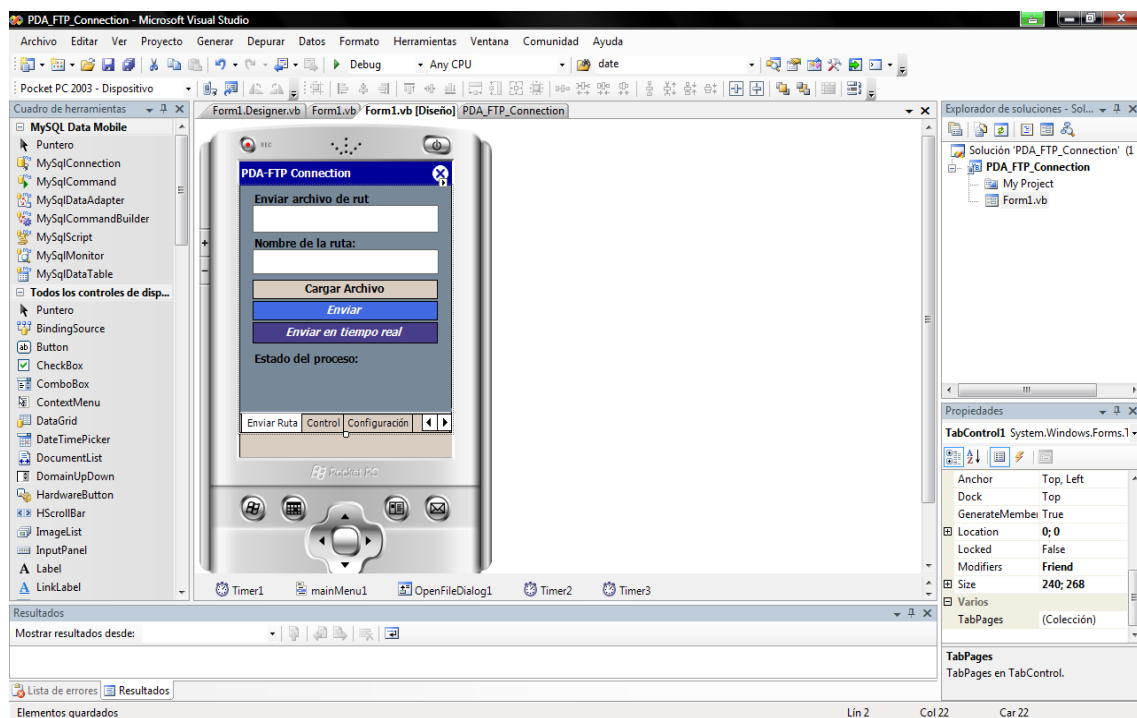


Figura 3: Entorno de desarrollo Visual Studio 2005

El emulador de PDA facilita la programación y las diferentes fases de prueba y depuración de código, al permitir emular el funcionamiento de la PDA y el sistema operativo de ésta.

El lenguaje de programación utilizado es Visual Basic .NET. Es un lenguaje de programación orientado a objetos que se puede considerar una evolución de Visual Basic implementada sobre el *framework .NET*³.

Mediante el cuadro de herramientas (*Figura 4*) podemos ir incrustando los diferentes objetos necesarios para nuestra aplicación, permitiéndonos modificar sus propiedades básicas de forma visual, ya sea modificando el tamaño de éstos directamente sobre ellos u otras propiedades a través de la ventana de propiedades. Y una vez diseñada la interfaz gráfica, es decir, la parte visual de nuestra aplicación procedemos a la programación de cada uno de estos objetos.

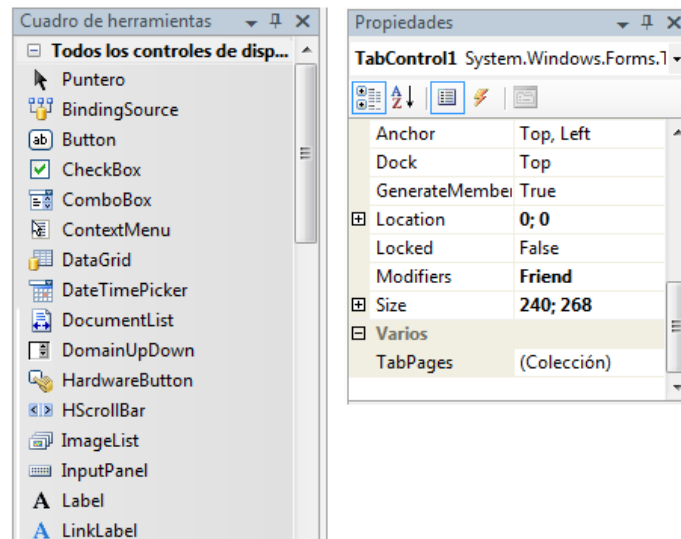


Figura 4: Cuadro de herramientas y ventana de propiedades de Visual Studio

4.2 Servidores web

En Internet, un servidor es un ordenador remoto que provee los datos solicitados por parte de los navegadores de otras computadoras.

Los servidores almacenan información en forma de páginas web y a través del protocolo HTTP lo entregan a petición de los clientes (navegadores web) en formato HTML.

4.2.1 FTP

FTP (*File Transfer Protocol*) es uno de los servicios más antiguos de Internet, permite mover uno o más archivos con seguridad entre distintos ordenadores proporcionando seguridad y organización de los archivos así como control de la transferencia.

Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

³ La palabra inglesa **framework** define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

En el desarrollo de software, un **framework** es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, con base en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

4.2.1.1 Servidor FTP

Un servidor FTP es un programa especial que se ejecuta en un equipo servidor normalmente conectado a Internet (aunque puede estar conectado a otros tipos de redes, LAN, MAN, etc.). Su función es permitir el intercambio de datos entre diferentes servidores/ordenadores.

Por lo general, los programas servidores FTP no suelen encontrarse en los ordenadores personales, por lo que un usuario normalmente utilizará el FTP para conectarse remotamente a uno y así intercambiar información con él.

Las aplicaciones más comunes de los servidores FTP suelen ser el alojamiento web, en el que sus clientes utilizan el servicio para subir sus páginas web y sus archivos correspondientes; o como servidor de backup (copia de seguridad) de los archivos importantes que pueda tener una empresa. Para ello, existen protocolos de comunicación FTP para que los datos se transmitan cifrados, como el SFTP (*Secure File Transfer Protocol*).

4.2.1.2 Cliente FTP

Cuando un navegador no está equipado con la función FTP, o si se quiere cargar archivos en un ordenador remoto, se necesitará utilizar un programa cliente FTP. Un cliente FTP es un programa que se instala en el ordenador del usuario, y que emplea el protocolo FTP para conectarse a un servidor FTP y transferir archivos, ya sea para descargarlos o para subirlos.

Para utilizar un cliente FTP, se necesita conocer el nombre del archivo, el ordenador en que reside (servidor, en el caso de descarga de archivos), el ordenador al que se quiere transferir el archivo (en caso de querer subirlo nosotros al servidor), y la carpeta en la que se encuentra.

Algunos clientes de FTP básicos en modo consola vienen integrados en los sistemas operativos, incluyendo Microsoft Windows, DOS, GNU/Linux y Unix. Sin embargo, hay disponibles clientes con opciones añadidas e interfaz gráfica. Aunque muchos navegadores tienen ya integrado FTP, es más confiable a la hora de conectarse con servidores FTP no anónimos utilizar un programa cliente.

4.2.1.3 CoreFTP

CoreFTP es un cliente FTP que cuenta con una interfaz de fácil manejo y muy intuitiva, cuya configuración de ventanas (local, remoto, log de mensajes entre cliente y server) podemos diseñar a nuestro gusto. Además, tiene soporte para la función *drag-and-drop* y posibilidad de conectarse a dos servidores simultáneamente.

CoreFTP tiene soporte para SSL/TLS y conexiones a través de firewall, y además te permite controlar el consumo de ancho de banda y tener tus sitios FTP de uso más frecuente perfectamente organizados en su gestor interno.

Otras funciones de gran utilidad son la posibilidad de previsualizar los ficheros en el servidor remoto, renombrarlos, editarlos, continuar descargas interrumpidas en el punto justo donde se quedaron, etc.

En la *Figura 5* se puede observar el programa en funcionamiento. Una vez realizada la conexión con el servidor FTP se carga todo el directorio de contenidos en la ventana inferior derecha, manteniendo a la izquierda un navegador de directorios local.

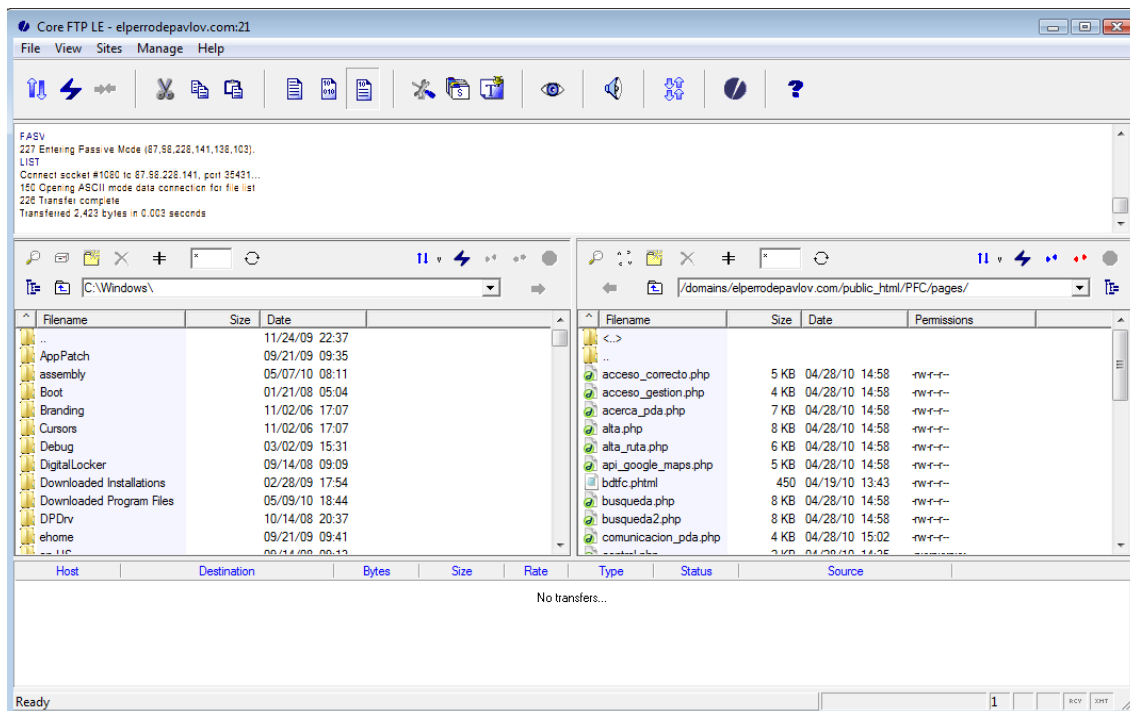


Figura 5: Cliente FTP CoreFTP

4.2.2 Entorno de programación de la interfaz web

Dreamweaver

Es una aplicación en forma de estudio enfocada a la construcción y edición de sitios y aplicaciones web basadas en estándares. Es el programa de este tipo más utilizado en el sector del diseño y la programación web, por sus funcionalidades, su integración con otras herramientas como *Adobe Flash* y, recientemente, por su soporte de los estándares del W3C (*World Wide Web Consortium*).

Cumple perfectamente el objetivo de diseñar páginas con aspecto profesional, y soporta gran cantidad de tecnologías, además muy fáciles de usar:

- Hojas de estilo y capas
- Javascript para crear efectos e interactividades
- Inserción de archivos multimedia...

En la *Figura 6* se puede observar el programa con la opción de visualización de código. Este programa permite también visualizar el código de la aplicación que se está programando y simultáneamente la vista de diseño mediante la selección de la pestaña "Dividir". De esta forma se puede ir viendo el resultado final a medida que se va programando, igual que se puede observar el código que genera cada objeto o elemento que se añade al diseño de la página.

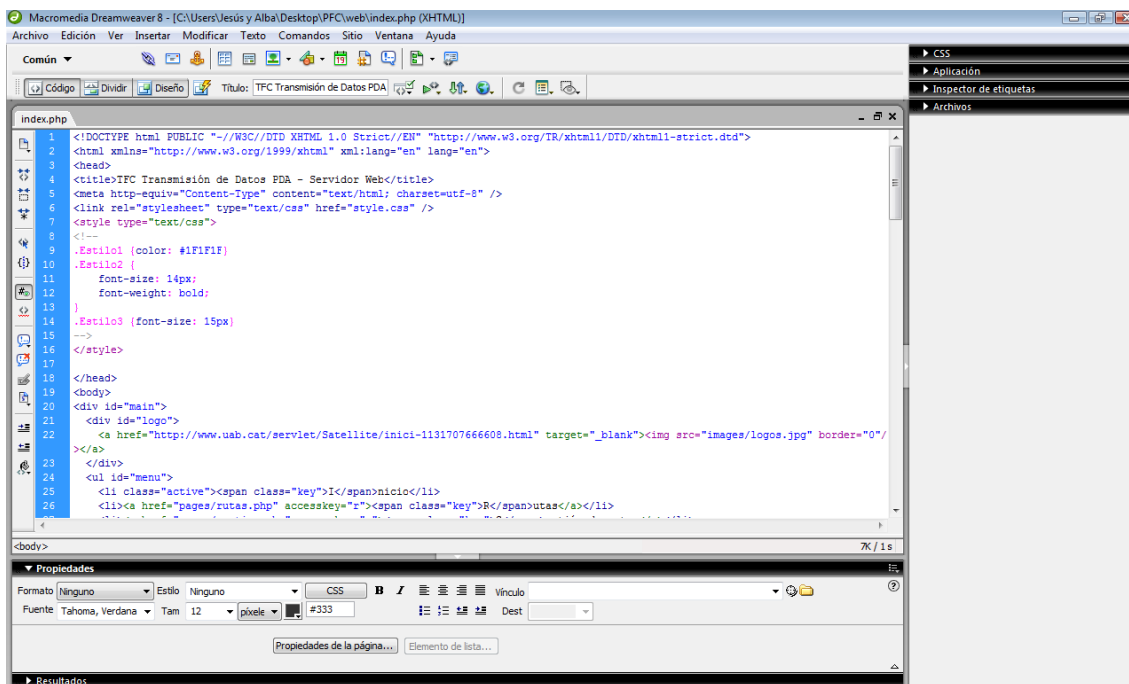


Figura 6: Dreamweaver, vista de código

PhpMyAdmin

Es una herramienta escrita en PHP (*HyperText Preprocessor*) con la intención de manejar la administración de MySQL[10] a través de páginas web, utilizando Internet. Actualmente permite crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL (*Structured Query Language*), administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 50 idiomas. En la Figura 7 se puede observar el contenido de una de las tablas de una base de datos.

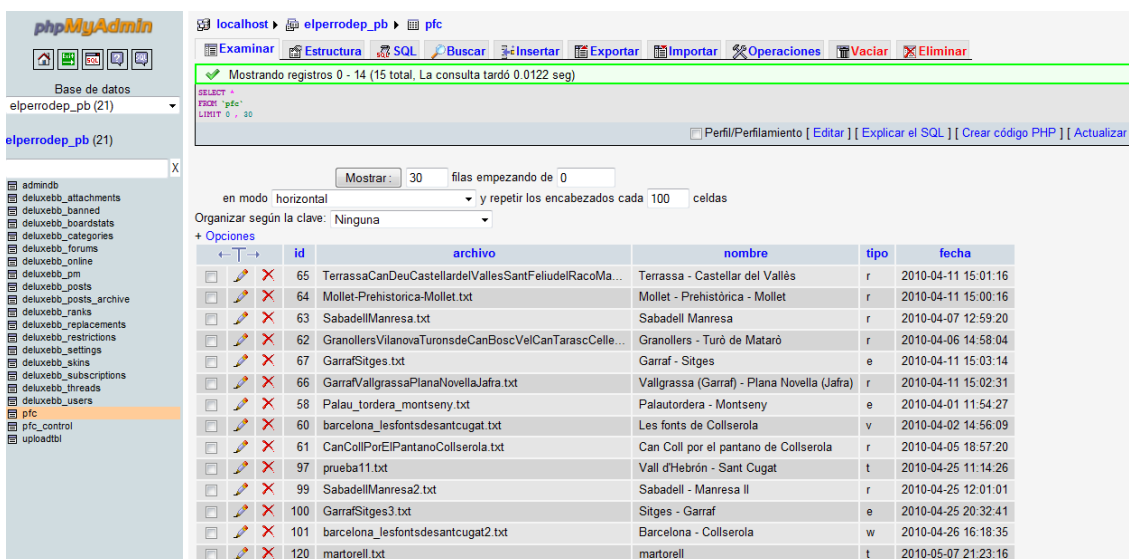


Figura 7: Aplicación web phpMyAdmin [11]

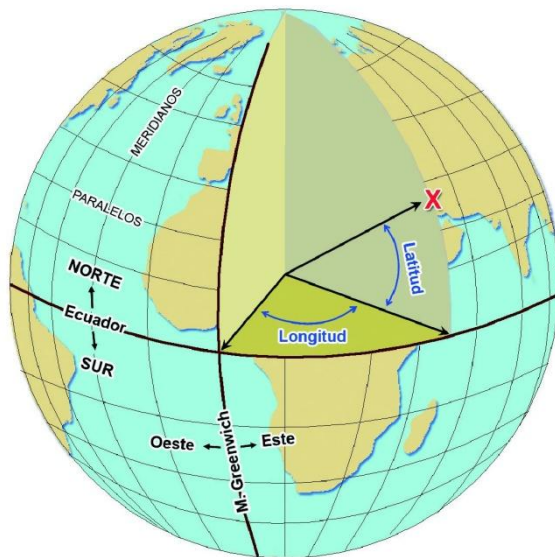
La aplicación en si no es más que un conjunto de archivos escritos en PHP que se puede copiar directamente a un directorio del servidor web en el que se aloja la página web, de modo que, cuando se accede a estos archivos, se muestran unas páginas

donde es posible encontrar las bases de datos a las que se tienen acceso en el servidor de bases de datos y todas sus tablas.

4.3 Posicionamiento básico: Latitud y Longitud

El sistema de coordenadas geográficas es un sistema de referencia que utiliza las dos coordenadas angulares latitud (norte o sur) y longitud (este u oeste) para determinar las posiciones de la superficie terrestre *Figura 8*, o en general de una esfera o un esferoide. Estas dos coordenadas angulares medidas desde el centro de

la Tierra son de un sistema de coordenadas esféricas que está alineado con su eje de rotación. La definición de un sistema de coordenadas geográficas incluye un *datum*⁴, meridiano principal y unidad angular.



Latitud: es la distancia angular entre el ecuador y un punto determinado del planeta medida a lo largo del meridiano que pasa por ese punto.

Longitud: expresa la distancia angular entre un punto dado de la superficie terrestre y el meridiano que se tome como 0° medida a lo largo del ecuador; habitualmente en la actualidad el meridiano de *Greenwich*.

Figura 8: Representación Latitud, Longitud en el globo terráqueo

4.4 Google Maps

A principios de 2005 *Google* lanzó el servicio de *Google Maps* (<http://maps.google.com>), que consiste en una página web en la que se nos muestra un mapa, con mayor o menor detalle, de cualquier parte del mundo.

Pero los mapas de *Google* son mucho más: sin duda lo que más sorprende de ellos es el nivel de interactividad que permiten; desde moverse por el mapa, hacer zoom, mostrar e introducir lugares y fotos de interés, hasta servicios más recientes, como mostrar el aspecto real de las calles a través de capturas de satélites, mostrar el tráfico (por ahora, sólo disponible en EEUU), o el más reciente en España *Google Street View*, que nos permite movernos literalmente por las calles de Madrid y Barcelona (aunque previsiblemente en los próximos meses veremos aumentar el número de ciudades disponibles).

⁴ Un datum es un conjunto de puntos de referencia en la superficie terrestre en base a los cuales las medidas de la posición son tomadas y un modelo asociado de la forma de la tierra (elipsoide de referencia) para definir el sistema de coordenadas geográfico.

4.4.1 API de Google Maps

Una **API** (*Application Programming Interface*)[6] es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Sin duda, lo que realmente supone una enorme diferencia respecto a los callejeros tradicionales es tener a nuestro alcance todo el potencial de *Google Maps* para ponerlo a nuestro servicio y crear nuestros propios mapas, tan sencillos o complejos como queramos, a través de la API pública y gratuita de este servicio. Esto nos permite crear mapas personalizados de todo tipo, desde mostrando la mejor ruta a nuestro negocio, marcando los puntos de interés de nuestro pueblo o superponiendo otro mapa para mostrar cómo quedaría un lugar con una nueva carretera. Las posibilidades son infinitas, y la flexibilidad inmensa.

Mediante un sencillo registro se proporciona una clave de API de Maps válida para un único "directorio" o dominio. Esta clave se ha de añadir al principio del documento *html* que va a contener el mapa mediante un pequeño *script*.

```
<script  
src="http://maps.google.com/maps?file=api&v=2&key=CLAVE  
PROPORCIONADA A PARTIR DEL REGISTRO" type="text/javascript">
```

4.4.1.1 GMap2

Crea una instancia de la clase *GMap2* para poder crear un mapa. Ésta es la clase central del API. Todo lo demás es auxiliar. Se crea un mapa nuevo dentro del contenedor HTML en cuestión, que generalmente suele ser una capa (elemento *DIV*).

```
var map = new GMap2(document.getElementById("capa_mapa"));  
  
map.addControl(new GSmallMapControl()); // Control de  
ampliación/disminución del mapa  
  
map.addControl(new GMapTypeControl()); // Control de selección  
de tipo de mapa  
  
map.setMapType(G_NORMAL_MAP); // Se establece la vista de mapa  
normal por defecto
```

4.4.1.2 GLatLng

GLatLng es un punto de acuerdo a una latitud y una longitud de coordenadas geográficas.

Hay que tener en cuenta que, aunque las proyecciones de mapas suelen asociar la longitud con la coordenada x del mapa y la latitud con la coordenada y, la coordenada de latitud siempre se escribe en primer lugar, seguida por la longitud, según es costumbre en cartografía.

```
GLatLng(lat:Number, lng:Number, unbounded?:Boolean)
```

4.4.1.3 GPolyline

GPolyline es una clase de la API de *Google Maps* que permite crear una superposición en el mapa que dibuja una *polilínea* utilizando para ello las funciones de dibujo vectorial del navegador si las hay o, si no, una superposición de imagen desde los servidores *Google*. Se puede observar un ejemplo en la *Figura 9*.

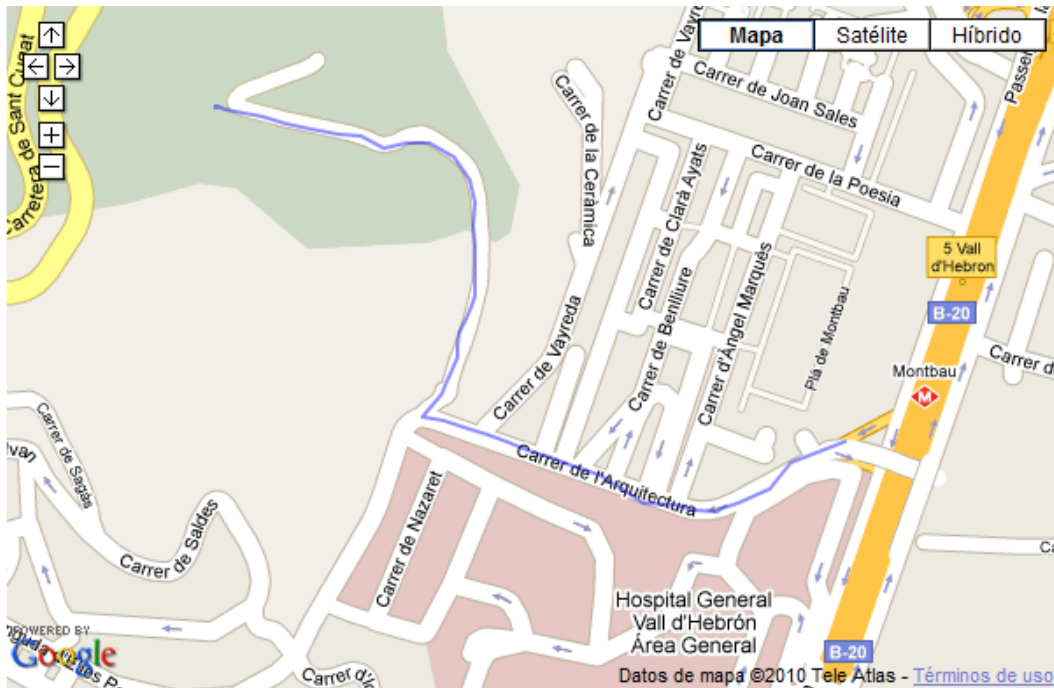


Figura 9: Mapa de Google Maps con una superposición de Polyline o ruta

```
GPolyline(latlngs:GLatLng[], color?:String, weight?:Number,
opacity?:Number, opts?:GPolylineOptions)
```

Crea una polilínea a partir de una matriz de vértices. El valor de color se indica como una cadena que contiene el color en estilo HTML numérico hexadecimal, por ejemplo, #RRGGBB. El valor de weight corresponde a la anchura de la línea en píxeles. El nivel de opacity se representa como un número entre 0 y 1. La línea será una semitransparente y suavizada.

Ejemplo:

```
map.setCenter(new GLatLng(41.430074,2.144615), 14);

var polyline = new GPolyline([new
GLatLng(41.429730,2.143810),new
GLatLng(41.429530,2.143160),...], "#0000ff", 2);

map.addOverlay(polyline);
```

4.4.1.4 GMarker

GMarker es la clase que permite colocar una marca en una posición (Latitud,Longitud) determinada del mapa.

Los objetos de marcador tienen elementos *latlng* correspondientes a la posición geográfica en la que los marcadores se anclan en el mapa, así como un elemento icon. Si en el constructor no se define la clase icon, se utiliza el icono predeterminado G_DEFAULT_ICON.

Tras agregarse a un mapa, la ventana de información se podrá abrir a través del marcador. El objeto marcador activará eventos de ratón y eventos de ventana de información. En la *Figura 10* se observa uno de estos marcadores sobre la trayectoria graficada.

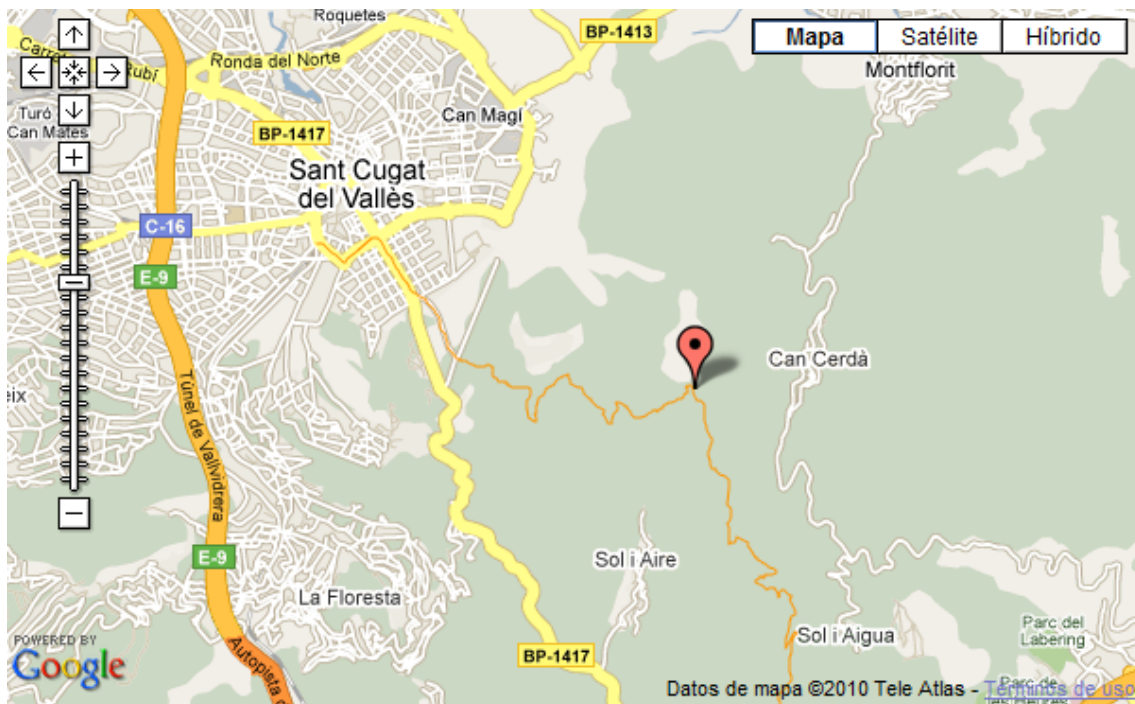


Figura 10: Ejemplo de marcador en un mapa de Google Maps

5. Desarrollo del proyecto

Si se observa la *Figura 11*, se pueden ver claramente las tres grandes fases que conforman el sistema implementado: la obtención de los datos de posicionamiento (FPGA), la transmisión de estos datos de la PDA hacia el servidor web, y la visualización en una interfaz web.

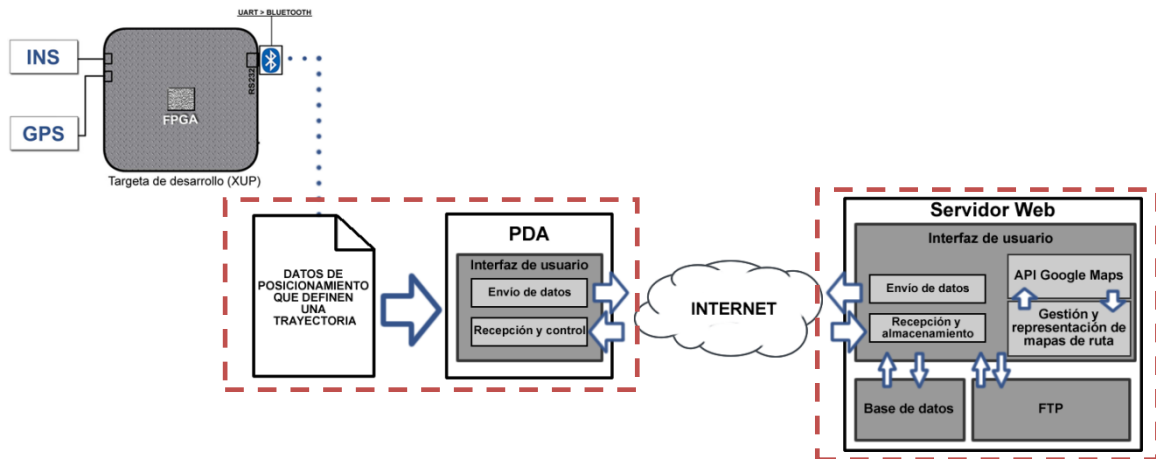


Figura 11: Esquema general del sistema a implementar

La FPGA instalada en el vehículo recoge los datos de posicionamiento a partir de los sensores dispuestos para ello. Estos datos son tratados y transmitidos mediante *Bluetooth* hacia la PDA, donde serán almacenados. El desarrollo de este proyecto comienza en este punto, separándolo en las dos grandes fases delimitadas por los dos recuadros remarcados en rojo de la *Figura 11*, pues una vez se dispone de todos los datos de posicionamiento en la PDA surge la necesidad de graficarlos, con el fin de facilitar su análisis. El sistema propuesto pretende mostrar estos datos sobre un mapa de *Google Maps* en una interfaz web, y por ello se precisa trasladar los datos de posicionamiento a un servidor web. Para ello se implementa una interfaz de usuario en la PDA que permite manipular y transmitir los archivos que contienen la trayectoria del vehículo.

Para poder mostrar visualmente los datos de posicionamiento del vehículo sobre un Mapa de *Google Maps*, se crea un entorno web que integra la tecnología de *Google Maps* y la base de datos que contiene todos los archivos transmitidos por la PDA.

A continuación se explican detalladamente todos los procesos realizados para llegar a completar las diferentes fases del proyecto citadas y conformar la solución tecnológica que cumple con los objetivos propuestos.

5.1 Desarrollo de la interfaz de usuario para la PDA

La tarea de transmisión de datos de la PDA hacia el servidor web comienza creando un nuevo proyecto *Smart Device*, haciendo uso de la plataforma de programación *Visual Studio 2005*, comentada anteriormente. La aplicación resultante (*Figura 12*) consta de un formulario con un conjunto de 5 pestañas que nos permiten visualizar las diferentes sub-funciones del programa: Inicio/Presentación, Envío de datos, Control, Configuración y Salida del programa.



Figura 12: Pantalla inicial y de salida de la interfaz de usuario de la PDA

5.1.1 Software Transmisión de datos PDA - Servidor FTP

Como ya se ha comentado en apartados anteriores, la principal función que debe cumplir la aplicación desarrollada para la PDA es la comunicación y la transmisión de datos con el servidor FTP donde está alojada la página o interfaz web. Para ello se ha de establecer la conexión con el servidor FTP vía WIFI cada vez que se necesite transmitir datos hacia el servidor.

5.1.1.1 Transmisión PDA-FTP Server vía Wifi

La segunda pestaña de nuestra aplicación (Enviar Ruta) muestra una pantalla con dos campos de texto: el primero para indicar la ruta local del archivo de ruta a enviar, y el segundo para asociar o establecer un nombre a dicho archivo.

Tal y como se muestra en la *Figura 13*, se dispone un botón que permite cargar el archivo de ruta local, y otros dos que nos permiten transmitirlo. El botón de carga de archivo abre un dialogo/ventana donde se muestra el conjunto de archivos locales disponibles en la PDA, filtrando el tipo para, únicamente, mostrar los .txt, que son los que se permiten transmitir. Una vez escogido el archivo, aparece su ruta en el primer campo de texto.



Figura 13: Formulario o ventana de envío de rutas

Los dos botones de envío proporcionan dos formas diferentes de transmitir el archivo de ruta. El primero envía el paquete completo, y el segundo simula lo que denominamos transmisión de ruta en tiempo real. Para ambos casos se establece una conexión segura con el servidor FTP. Para ello se utiliza una librería que ejerce la función de cliente FTP[3].

Para llevar a cabo la simulación de la transmisión de una ruta en tiempo real se divide el conjunto de puntos geográficos que contiene el archivo de ruta .txt en pequeños paquetes de 20 puntos cada uno y cada 50ms se añade un paquete de 20 puntos al paquete a transmitir, de tal forma que en la interfaz web se irá recibiendo la información de la ruta progresivamente, creando el efecto de movimiento del vehículo con el correspondiente muestreo de la trayectoria en tiempo real.

En la parte inferior de la pantalla se puede observar en todo momento el estado del proceso de envío, visualizando el porcentaje enviado y la velocidad a la cual se está transmitiendo (puntos por segundo; teniendo en cuenta que un punto está formado por una latitud y una longitud).

5.1.1.2 Implementación del Cliente FTP

Como ya se ha comentado en el apartado anterior, la conexión con el servidor FTP o la implementación de lo que sería el cliente FTP, se realiza mediante una librería que ya proporciona una serie de funciones que permiten conectar con el servidor FTP, enviar datos, consultar datos, desconectarnos,...

A continuación se detallan algunas de estas funciones, propiedades y/o eventos, aunque se puede obtener toda la información relativa a esta librería en su página web [3].

Login(): Función que inicia una sesión con el servidor FTP, pasando el nombre de usuario y contraseña especificados durante la declaración de la variable *tipo ClienteFTP* o por sus propiedades. Devolviendo *True* si se ha logrado la conexión o *False* si ha fallado. También devuelve eventos que fueron ocasionados por un error, ya sea por el servidor o por el sistema.

Logout(): Función que cierra la sesión con el servidor FTP.

UploadFile(sFileName[,sBinario]): Función que carga un archivo al directorio actual en el servidor FTP. *sBinario* es opcional. Si es *False* se indica que se va a cargar un archivo de texto plano y si es *True* que se va a cargar un archivo binario (zip,rar,avi,jpg,gif,etcétera).

OcurrioError(sError): Evento que ocurre cuando ha existido un error del sistema, como por ejemplo error en los socket⁵, variables o índices fuera del límite.

NombreServidor(): Propiedad que asigna o recupera el nombre del Servidor FTP.

CarpetaRemoto(): Propiedad que asigna o recupera el directorio con el que se trabaja al iniciar la sesión con el Servidor FTP.

⁵ Socket designa un concepto abstracto por el cual dos programas (posiblemente situados en ordenadores distintos) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada. Un *socket* queda definido por una dirección IP, un protocolo de transporte y un número de puerto.

PuertoServidor(): Propiedad que asigna o recupera el puerto por el que se establece la conexión con el Servidor FTP.

NombreUsuario(): Propiedad que asigna o recupera el *Nombre de Usuario* con el que se puede acceder al Servidor FTP.

PasswordUsuario(): Propiedad que asigna o recupera la *Contraseña de Usuario* con el que se puede acceder al Servidor FTP.

TipoServidor(): Propiedad que asigna o recupera el tipo de soporte de la conexión del Servidor. *True*=Pasivo, *False*=Activo.

5.1.1.3 Configuración de la conexión

La pestaña *Configuración* de la aplicación da paso a la ventana que permite modificar la configuración de nuestro cliente FTP. En la *Figura 14* se puede observar que se permite modificar el servidor, el usuario, el *password* y la ruta.



Figura 14: Formulario o ventana configuración

De esta forma se puede cambiar el servidor con el que se desea trabajar, de forma sencilla y sin tener que acceder al código de la aplicación, modificando los datos de acceso básicos, que son el nombre del servidor con el que se quiere conectar, el nombre de usuario y su contraseña de acceso correspondiente, y por último la ruta/directorio dentro del servidor donde se quieren cargar los archivos que se enviarán.

5.1.2 Formato de los archivos de ruta

Ya se han comentado algunos aspectos sobre los archivos a transmitir en apartados anteriores, como por ejemplo, que se trata de archivos de texto plano .txt (*Figura 15*), o que están compuestos del conjunto de puntos geográficos (latitud, longitud) que

conforman la trayectoria del vehículo. No obstante, su estructura o formato no se queda ahí.

Como se verá en la interfaz web, se distinguen varios tipos y formatos de archivos de ruta, donde cada uno de estos aportará un valor diferente para el usuario. Tal y como se puede observar en la *Tabla 1*, se distinguen 6 tipos diferentes.

| Nombre del tipo de ruta | Codificación |
|-----------------------------------------------------------|--------------|
| Ruta simple | r |
| Ruta con información de velocidad | v |
| Ruta comparativa de sistemas (error GPS) | e |
| Ruta comparativa de sistemas con información de velocidad | w |
| Ruta simple que se transmitió en tiempo real | t |
| Ruta simple que se está transmitiendo en tiempo real | a |

Tabla 1: Tabla de tipos de ruta

- Para el caso de las **rutas simples**, la estructura del archivo es [r,latitud,longitud,latitud,longitud,...], donde el parámetro **r** define el tipo de ruta. A continuación, aparecen los diferentes puntos de la trayectoria separados por comas, teniendo en cuenta que cada punto está formado por una latitud y una longitud.
- En las **rutas con información de velocidad** la estructura del archivo es [v,latitud,longitud,velocidad,latitud,longitud,veocidad,...]. Como se puede ver, **v** es el valor que toma el parámetro que define este tipo de rutas.
- Para las **rutas con comparación de sistemas** se utilizan dos archivos con la siguiente estructura [e,latitud,longitud,latitud,longitud,...]. El valor **e** en el parámetro de tipo de ruta indicará que se trata de una ruta con comparación y se graficarán los dos archivos de ruta simultáneamente.
- Para el caso de las **rutas en tiempo real**, el archivo de ruta de éstas es idéntico al de las rutas simples, sólo que el valor que toma el parámetro de tipo de ruta es **a** mientras se está transmitiendo y **t** cuando ya se ha finalizado la transmisión.
- Por último, se puede dar el caso de tener una **ruta con información de velocidad y comparación de sistemas**. En este caso el formato del archivo es el mismo que para el caso de rutas con comparación de sistemas, con la única diferencia de que el valor que toma el parámetro de tipo de ruta es **w**.

La sencillez de la estructura de estos archivos permite identificar el tipo de ruta y graficarla en el menor tiempo posible y con el mayor detalle.

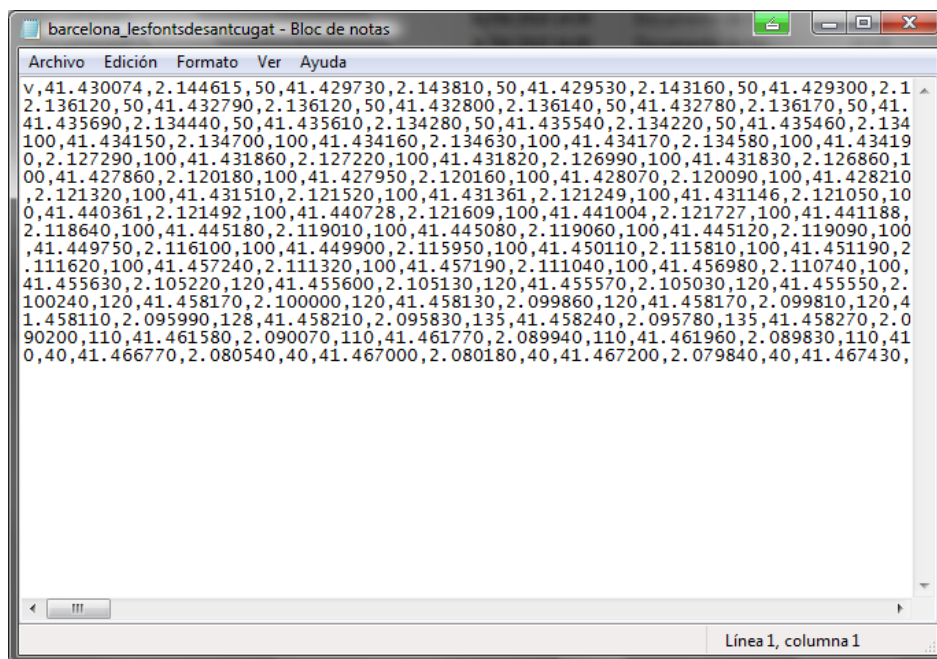


Figura 15: Archivo de ruta con información de velocidad

5.1.3 Recepción de datos

Una función adicional de la aplicación de la PDA es la que ofrece la pestaña “Control”. A través de esta opción se accede a una ventana en la que se puede comprobar el estado (activo/inactivo) de dos dispositivos del vehículo, siendo su modificación posible únicamente desde la página web. Con esto se puede hablar de recepción de datos, ya que el estado de los dispositivos se modifica desde la página web, y en la interfaz de la PDA se reciben los datos referentes al estado de los mismos.

Mediante los botones “Iniciar control” y “Detener control” se puede iniciar o detener la comprobación del estado de ambos dispositivos, o lo que es lo mismo, la recepción de datos.

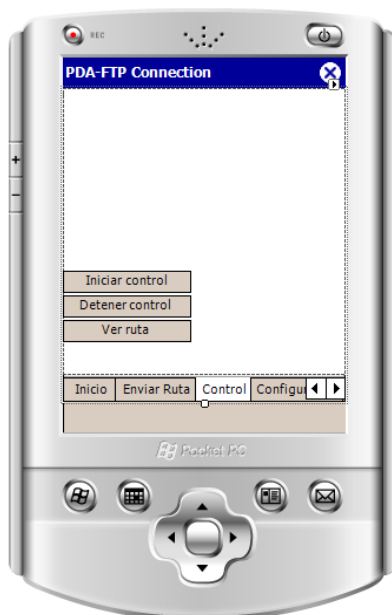


Figura 16: Formulario o ventana de control y recepción de datos

Como se puede observar en la *Figura 16*, existe un tercer botón que permite visualizar en la misma PDA el mapa de la última ruta cargada en el servidor. Esta opción ejecuta una ventana del navegador de internet para dispositivos móviles *Skyfire* [13] que permite la visualización de mapas de Google Maps (*Figura 17*), y lo que es más importante, permite interactuar con el mapa (ampliarlo, reducirlo, desplazarlo...).

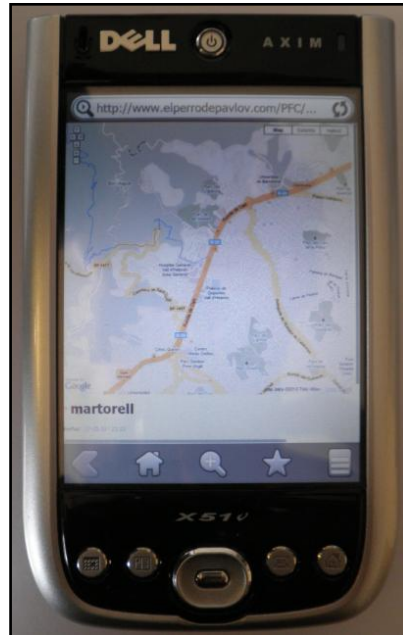


Figura 17: Visualización de ruta en Skyfire

Una vez se ha completado la creación de la interfaz de usuario para la PDA se precisa de la aplicación web capaz de interpretar los datos enviados por la PDA. En el siguiente capítulo se detalla el desarrollo del entorno web.

5.2 La interfaz web

La segunda gran fase del proyecto se encuentra al otro lado de la transmisión de los archivos de ruta, donde nos encontramos con la aplicación web que procesa dichos archivos. Básicamente, esta aplicación web muestra al usuario un mapa de Google Maps con la trayectoria que contiene el archivo en cuestión y con toda la información adicional que corresponda. Una interfaz accesible, ágil y de sencilla navegación.

Esta interfaz web está creada combinando varios lenguajes de programación: *HTML*, *CSS* (páginas de estilo), *PHP* y *MySQL* para realizar las conexiones pertinentes con la base de datos y establecer las sesiones (contenido bajo autenticación de usuario) para la gestión de las rutas, y *JavaScript* para trabajar con la API de *Google Maps*.

Una vez que la PDA carga el archivo de ruta en el FTP, se conecta con una página en el servidor web, la cual ejecuta el código que da de alta la ruta en la base de datos asociándola al nombre indicado, de forma que posteriormente se pueda acceder a dicho archivo fácilmente. A partir de aquí, la ruta y los datos relacionados quedan almacenados, y el archivo entra ya a formar parte del conjunto de información visible en la página web.

etse UAB
Universitat Autònoma de Barcelona

1 Inicio Rutas Gestión de rutas Tiempo real Comunicación PDA

Depto. de Microelectrónica y Sistemas Electrónicos

El Proyecto

API Google Maps 2

Enlaces de interés

Nombre de la ruta: 3

Buscar

Últimas rutas cargadas 4

[07-05-10] · 21:23 martorell

[02-05-10] · 01:05 Ruta con 50 puntos

[26-04-10] · 16:18 Barcelona - Collserola

Información acerca de la PDA y de la transmisión de archivos al Servidor Web.

Trasmisión de Datos PDA-Servidor Web

Monitorización de trayectorias

Visualización de la trayectoria de un vehículo (aéreo o terrestre) online, a partir de los datos de posicionamiento recibidos desde una PDA a través de una conexión WIFI

Trayectorias en tiempo real

Esta página web nos permite visualizar la trayectoria de un vehículo en tiempo real. Los datos son enviados desde una PDA que esta acoplada a un sistema de navegación mediante bluetooth. Es decir, si se están transmitiendo datos en el mismo instante en el que nos encontramos navegando por la página web, podemos visualizar la posición y el recorrido del vehículo en el momento.

No se están recibiendo datos 5

Posición y velocidad

Además de visualizar la trayectoria realizada por el vehículo, podemos obtener información sobre la posición (Lat,Lon) y la velocidad (Km/h) en cada uno de los puntos mostrados en el mapa, gracias a que nuestro sistema de navegación nos permite enviar información adjunta de posición y velocidad.

Error GPS

Comparamos los resultados obtenidos de nuestro sistema de navegación (Integración GPS/INS), con los datos obtenidos de un GPS tradicional. Ambas trayectorias se visualizan en Google map, los que nos permite apreciar la precisión que ofrece nuestro sistema.

Comunicación con la PDA

Tenemos la posibilidad de transmitir datos en sentido contrario. Esto significa que podremos manipular algunos aspectos del vehículo desde la página web, enviando los bits precisos hacia la PDA.

© Copyright Jesús Iglesias García 2010 Trabajo Final de Carrera - Ingeniería Técnica de Telecomunicaciones Especializada en Sistemas Electrónicos

Figura 18: Página web principal del proyecto

5.2.1 Estructura del sitio web

A continuación se detalla la estructura básica de funcionamiento de la página web del proyecto, tomando como referencia los diferentes puntos que se pueden observar en la *Figura 18*.

1. **Menú principal:** Este menú permite al usuario acceder a la página de rutas, gestión de rutas, tiempo real, comunicación PDA y una opción para volver a la página principal (Inicio).
 - **Rutas:** En este apartado se permite acceder a la visualización de todas las rutas cargadas en el servidor y toda su información relacionada.
 - **Gestión de rutas:** Mediante el acceso bajo verificación de usuario y contraseña se accede a la posibilidad de gestionar las rutas cargadas en el servidor (modificar los datos de las rutas, añadir nuevas rutas desde la página web y eliminarlas).
 - **Tiempo real:** Esta sección muestra la trayectoria que se esté transmitiendo en el mismo instante que se accede.
 - **Comunicación PDA:** En este apartado se permite enviar información hacia la PDA.

2. **Menú secundario:** Este menú contiene información teórica sobre el proyecto y la API de Google Maps, además de una serie de enlaces relacionados.
3. **Búsqueda de rutas:** Es posible buscar entre las rutas cargadas en el servidor, simplemente indicando el nombre de la ruta que se desea visualizar.
4. **Panel de últimas rutas cargadas:** Este panel, presente en prácticamente todas y cada una de las páginas de la interfaz web, muestra las tres últimas rutas cargadas, con la posibilidad de acceder a su visualización con un simple clic sobre ellas.
5. **Alerta de recepción de datos en tiempo real:** Este pequeño recuadro contiene el “chivato” que nos alerta de una transmisión de datos en tiempo real.

Todos estos apartados de la página web se verán con detalle a lo largo de este capítulo. No obstante, se puede observar el conjunto de páginas que conforman el sitio web del proyecto en el esquema de la *Figura 19*.

5.2.1.1 Árbol de contenidos

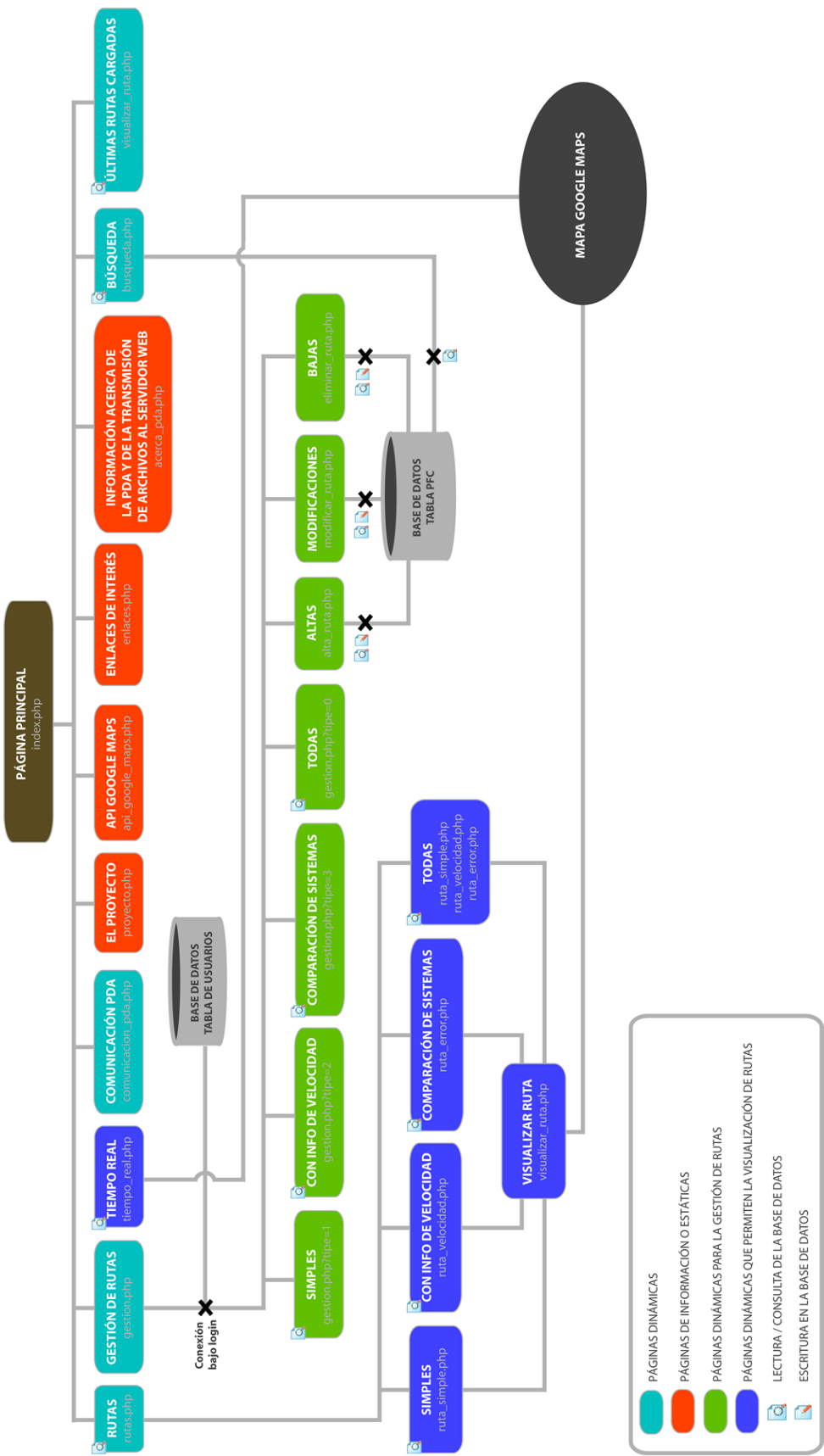


Figura 19: Esquema del árbol de contenidos de la página web

5.2.2 Base de datos

Las rutas que son cargadas en el servidor FTP se han de registrar en algún lugar para que, posteriormente, se pueda tener constancia de ellas; ya se hayan cargado desde la PDA o a través de la página web.

Este registro de rutas se lleva a cabo mediante una base de datos que está alojada en el mismo servidor, tal y como se comentaba al inicio de este capítulo. Esta interacción se realiza, por un lado, a partir de las funciones que PHP[7] nos propone para cada tipo de base de datos y, por otro, estableciendo un diálogo a partir de un idioma universal: SQL, común en todas las bases de datos.

Para nuestra aplicación web hemos elegido *MySQL*[10], sin duda la base de datos más extendida en combinación con PHP. Su gratuidad, eficiencia y simplicidad han hecho de ella una buena candidata.

La creación de dicha base de datos se realiza mediante la interfaz web *phpMyAdmin*, la cual se encuentra instalada en el servidor web que se utiliza.

5.2.2.1 Tablas

Las tablas se componen de dos estructuras:

- **Campo:** Corresponde al nombre de la columna. Debe ser único y tener, además, un tipo de dato asociado.
- **Registro:** Corresponde a cada fila que compone la tabla. Aquí se componen los datos y los registros. Eventualmente pueden ser nulos en su almacenamiento.

En la definición de cada campo, debe existir un nombre único, con su tipo de dato correspondiente. Esto es útil a la hora de manejar varios campos en la tabla, ya que cada nombre de campo debe ser distinto entre sí.

A los campos se les puede asignar, además, propiedades especiales que afectan a los registros insertados. El campo puede ser definido como *índice* o *autoincrementable*, lo cual permite que los datos de este campo cambien solos, o que el campo sea considerado el principal a la hora de ordenar los datos contenidos.

Dentro de la base de datos utilizada se han creado dos tablas: una para almacenar toda la información relativa a todos y cada uno de los archivos de ruta que se vayan cargando en el servidor (*Tabla 2*), y otra para registrar a los usuarios que tendrán acceso al apartado de gestión de la página web (*Tabla 3*).

| Campo | Tipo | Cotejamiento | Autoincremento | Descripción |
|----------------|----------|------------------------|----------------|-------------------------|
| id | int(11) | | Si | Índice de la tabla |
| archivo | text | utf8_spanish_ci | No | Nombre del archivo .txt |
| nombre | text | utf8_spanish_ci | No | Nombre de la ruta |
| tipo | text | utf8_spanish_ci | No | Tipo de ruta |
| fecha | datetime | dd/mm/yyyy 00:00:00 | No | Fecha y hora de carga |

Tabla 2: Tabla pfc de la base de datos

| Campo | Tipo | Cotejamiento | Autoincremento | Descripción |
|-------------------|------|-----------------|----------------|------------------------|
| usuario | text | utf8_spanish_ci | No | Nombre del usuario |
| contrasena | text | utf8_spanish_ci | No | Contraseña del usuario |

Tabla 3: Tabla admindb de la base de datos

Una vez creada la base de datos y las tablas mencionadas a través de la aplicación *phpMyAdmin*, podremos acceder a ambas tablas mediante el siguiente código php.

```
<?php
function Conectarse()
{
    if (!($link=mysql_connect("localhost","usuariobasededatos","contraseña")))
    {
        echo "Error al conectar con la base de datos.";
        exit();
    }
    if (!mysql_select_db("nombredelabasededatos",$link))
    {
        echo "Error al seleccionar la base de datos.";
        exit();
    }
    return $link;
}

$link=Conectarse();

mysql_close($link); //cierra la conexion
?>
```

Este código está contenido en el archivo denominado *bdpfc.phtml*, al cual se efectuará una llamada cada vez que se desee realizar cualquier tarea con la base de datos.

5.3 Monitorización de trayectorias

Ésta es una de las tareas más importantes realizadas en este proyecto. A través de la opción de menú “Rutas” se puede seleccionar la ruta que se desee visualizar de un listado ordenado. En el momento que se escoge la ruta que se quiere visualizar, se ejecuta el código de JavaScript que realiza la lectura del archivo de ruta y transmite su contenido a las funciones del *API de Google Maps* indicadas para la proyección de la ruta en un mapa.

El mapa aparece en el marco inferior derecho, con la trayectoria del vehículo graficada sobre éste en color azul, siempre ofreciendo las opciones básicas de cualquier mapa de *Google*, es decir, permitiendo el desplazamiento por el mapa, la ampliación de éste y el cambio de sistema de visualización (Mapa/Satélite/Híbrido), tal y como se observa en la *Figura 20*.

Las trayectorias y/o rutas que se muestran en el listado mencionado se dividen en tres tipos diferentes, correspondientes a las tres últimas columnas del listado. Estos tipos son: rutas con información de la velocidad del vehículo en cada punto de la trayectoria, comparación de sistemas (GPS vs. INS+GPS) mediante dos rutas de diferente color, y tiempo real, que muestra la trayectoria de un vehículo en tiempo real, siempre y cuando se estén recibiendo datos en el mismo instante.



Figura 20: Visualización de una trayectoria simple

5.3.1 Visualización de rutas en Google Maps

En el marco inferior derecho de la página de visualización de rutas (*visualizar_ruta.php*) se carga un mapa de Google de 600 x 400 pixeles, mediante un *iframe*⁶ (*Inline frame*).

```
<iframe src="<? if($tipe=="r"){echo ("ruta_simple.php?ruta="); }
if($tipe=="v"){echo ("ruta_velocidad.php?ruta="); }
if($tipe=="e"){echo ("ruta_error.php?ruta="); }
if($tipe=="t"){echo ("visualizacion_tiempo_real.php?ruta="); }
echo $archivo; ?>" frameborder="0" scrolling="no" width="610"
height="410" name="ruta"> </iframe>
```

A partir del código *php* que se incrusta dentro de la definición del *iframe*, se cargará una página de visualización de mapa u otra, en función del tipo de ruta (*ruta_simple.php*, *ruta_velocidad.php*, *ruta_error.php*, *visualizacion_tiempo_real.php*)

Estas páginas, que suponen los diferentes tipos de visualización de ruta, son las que contienen el código que interactúa con el *API de Google Maps*. Tal como se ha visto en el punto 4.4.1 *API de Google Maps*, el primer paso a la hora de insertar un mapa de Google en una página web es añadir el script que contiene la clave de registro a *API*

⁶ Es un elemento HTML que permite insertar o incrustar un documento HTML dentro de un documento HTML principal.

de *Google Maps*, y a partir de aquí se emplean las funciones del API necesarias para graficar la trayectoria del vehículo.

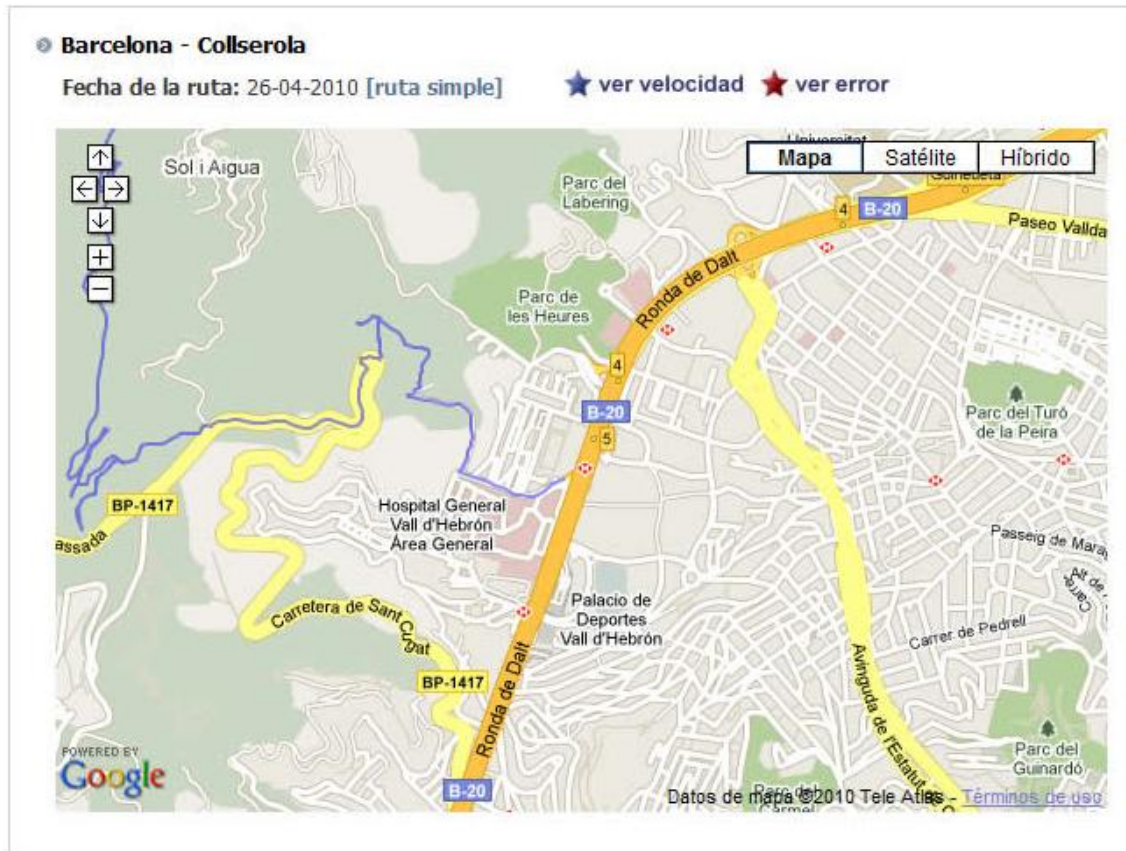


Figura 21: Marco de visualización de ruta (ruta simple)

Como se puede observar en la *Figura 21*, las páginas de visualización de ruta que se cargan dentro del *iframe* muestran la información asociada a la ruta que se carga en el mapa en la parte superior: el nombre de la ruta, la fecha en la que se cargó y las diferentes vistas disponibles para la ruta que se está visualizando (*ver velocidad*, *ver error*,...). En los siguientes puntos se profundizará en los detalles de cada uno de los tipos de visualización.

5.3.2 Tipos de rutas

Como ya se ha visto anteriormente, al acceder al apartado de rutas de la página web se muestra un listado con las rutas existentes en la base de datos del servidor. Tal como se muestra en la *Figura 22*, dicho listado dispone de cinco columnas, a través de las cuales se indica el nombre de la ruta, la fecha en la cual fue cargada, y si esta ruta dispone de información sobre la velocidad del vehículo, comparación entre sistemas, o visualización de trayectoria en tiempo real, utilizando la simbología indicada en la *Tabla 4*.

Trasmisión de Datos PDA-Servidor Web

Rutas

Busque la ruta que desee visualizar entre las rutas recibidas desde la PDA o cargadas a través de la web.

Listado de rutas

| Nombre | Fecha | Velocidad | Error | Tiempo Real |
|---------------------------------------------|------------|-----------|-------|-------------|
| martorell | 07-05-2010 | | | |
| Ruta con 50 puntos | 02-05-2010 | | ★ | |
| Barcelona - Collserola | 26-04-2010 | ★ | ★ | |
| Sitges - Garraf | 25-04-2010 | | ★ | |
| Sabadell - Manresa II | 25-04-2010 | | | |
| Vall d'Hebrón - Sant Cugat | 25-04-2010 | | | |
| Garraf - Sitges | 11-04-2010 | | ★ | |
| Vallgrassa (Garraf) - Plana Novella (Jafra) | 11-04-2010 | | | |

Páginas: 1 2

Figura 22: Apartado de Rutas de la interfaz web

Pulsando sobre el nombre de la ruta se accede a la visualización simple de la ruta. Las estrellas azules, rojas y verdes corresponden a la visualización de ruta con información sobre la velocidad del vehículo, comparación entre sistemas, o visualización de trayectoria en tiempo real, respectivamente.

| Icono/Acceso tipo de ruta | Descripción |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [ruta simple] | Visualización de la trayectoria definida en el archivo de ruta correspondiente a la ruta seleccionada, sobre el mapa. |
| ★ ver velocidad | Visualización de la trayectoria definida en el archivo de ruta correspondiente a la ruta seleccionada, sobre el mapa, con el añadido de que se permite ver la velocidad a la que circulaba el vehículo en cada punto de la trayectoria mostrada. |
| ★ ver error | Visualización de dos trayectorias en un mismo mapa, una correspondiente a los datos proporcionados por un sistema de GPS tradicional y otra correspondiente a los datos proporcionados por el sistema GPS/INS. |
| ★ ver tiempo real | Visualización de la trayectoria de un vehículo en tiempo real. Este icono/acceso está activo cuando se están recibiendo datos, en el caso contrario no aparece. |

Tabla 4: Tipos de visualización de ruta

5.3.2.1 Rutas simples



Figura 23: Mapa con ruta simple

La página de visualización de ruta simple se encarga de leer el archivo de ruta correspondiente y de transferir los datos de posicionamiento leídos a la función del API de *Google Maps* que sirve para dibujar rectas en el plano. De esta forma, la clase *GPolyline* permite crear una recta entre todos y cada uno de los puntos geográficos de la trayectoria almacenada en el archivo, dibujando así el recorrido completo sobre el mapa, tal y como se puede observar en la *Figura 23*.

5.3.2.2 Rutas con información de velocidad

En este caso, el archivo de ruta contiene un dato adicional junto a cada punto (Latitud, Longitud), la velocidad a la cual circulaba el vehículo en ese punto en concreto, de forma que se trabaja con tramas o paquetes de tres datos (latitud, longitud, velocidad).

El método que se emplea para graficar la trayectoria del vehículo sobre el mapa es el mismo que en el caso de las rutas simples. No obstante, el dato que corresponde a la velocidad se va almacenando en un vector de datos a medida que se van leyendo los diferentes puntos, con el fin de que posteriormente pueda ser asociado al punto correspondiente en el mapa.

Mediante el uso del control del evento *mousemove* sobre el mapa se detecta la posición del puntero del ratón sobre el mapa, y se muestra un marcador de posición sobre el punto de la trayectoria más cercano al puntero del ratón.



```
GEvent.addListener(me.map_, 'mousemove', Event.callback(me, me.updateMarkerLocation_));
```

Para situar el marcador sobre la trayectoria se hace uso de varias funciones que consiguen calcular el punto exacto de la trayectoria más cercano al puntero del ratón, basándose en el cálculo de distancias mínimas. Esta parte está basada en ejemplos que se pueden encontrar en la faq y librería de códigos y ejemplos de *googlecode* [12].



Figura 24: Mapa de ruta con información de la velocidad del vehículo

En la *Figura 24* se observa un ejemplo de una ruta con información de la velocidad del vehículo que recorre la trayectoria graficada. En el recuadro de la parte superior del mapa se muestra la posición del punto de la trayectoria en la que está situado el marcador y la velocidad a la que circulaba dicho vehículo en ese instante. Moviendo el cursor del ratón por encima del mapa se puede ir desplazando el marcador a lo largo de la trayectoria y visualizar los datos asociados a cada uno de los puntos de ésta.

5.3.2.3 Rutas de comparación de sistemas

Este modo de visualización de rutas ofrece información importante para el proyecto general desarrollado en el *Departamento de Microelectrónica y Sistemas Electrónicos*, ya que se permite observar la trayectoria obtenida a partir de los datos proporcionados por un sistema de GPS tradicional junto con la trayectoria que se obtiene del nuevo sistema GPS/INS, facilitando así la comparación y el análisis de ambos sistemas de posicionamiento.

El proceso de visualización de las dos trayectorias simultáneamente es muy similar al de una ruta simple, pues lo que se hace es graficar una trayectoria después de la otra, como dos objetos de la clase *GPolyline* anteriormente mencionada.

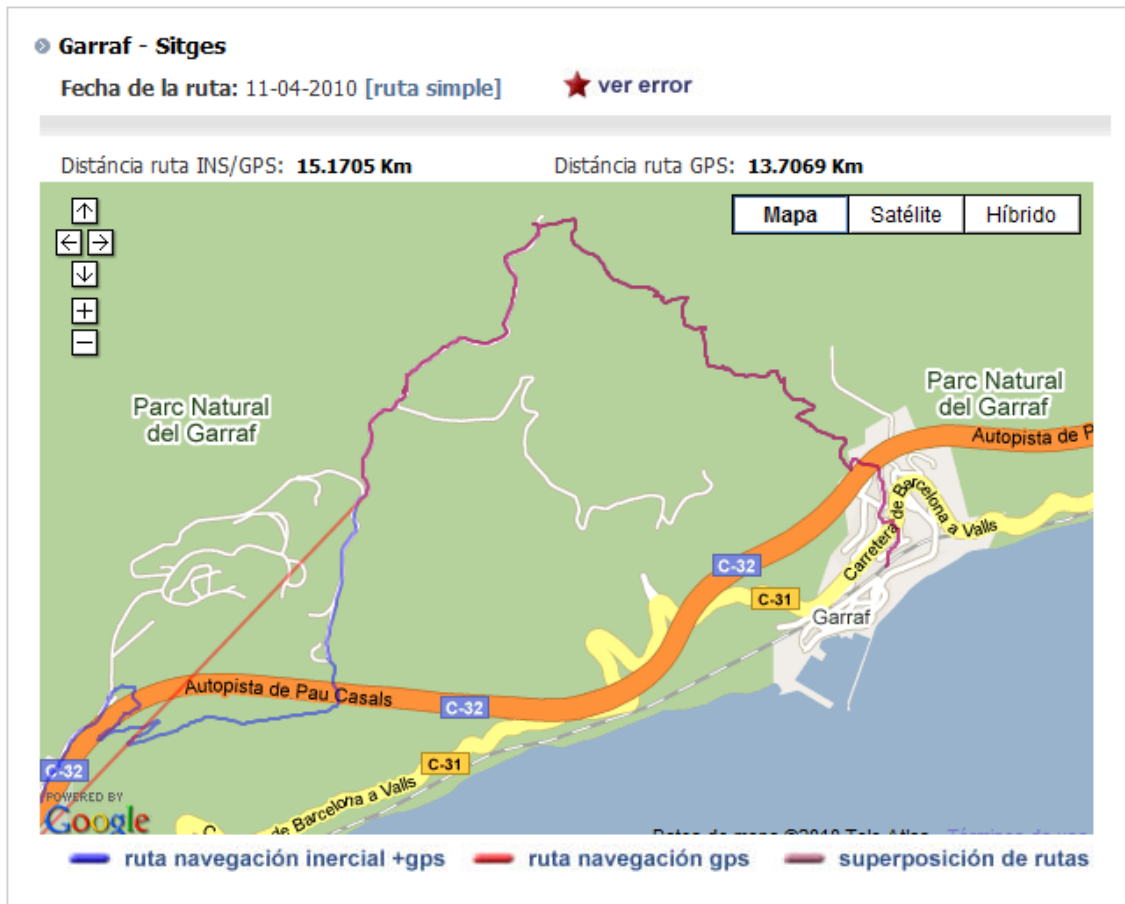


Figura 25: Mapa con comparación de sistemas de posicionamiento

Se utilizan colores distintos para cada una de las dos trayectorias, para que se puedan distinguir fácilmente, y tal y como se puede observar en la leyenda que aparece en la parte inferior de la *Figura 25*, en los instantes en que ambos sistemas coinciden se utiliza un tercer color. De esta manera, se puede visualizar muy fácilmente el error que comete el sistema de GPS tradicional frente al nuevo sistema propuesto, de aquí el segundo nombre de este tipo de visualización de ruta, "Ruta de Error".

Como información adicional también se muestra la distancia total que comporta cada una de las dos trayectorias en la parte superior del mapa, aprovechando la precisión que ofrece la propiedad *getLength()* del objeto *Polyline*.

```
var d1 = polyline.getLength();

d1 = Math.round(d1*10) / 10;
document.dist.km1.value = d1 / 1000;
document.dist.km1.value = document.dist.km1.value + " Km";
```

5.3.2.4 Rutas en tiempo real

Esta opción de visualización de rutas comprueba, a través de la base de datos, si existe alguna ruta que se esté transmitiendo en el instante que se accede a este modo de visualización, y en el caso de que sea así, se grafica como una ruta simple y se va actualizando la información de la trayectoria muestreada periódicamente.

En el caso de que en el momento de acceder al apartado de tiempo real no se esté transmitiendo ninguna ruta, se muestra un mensaje indicando la última ruta que se transmitió en tiempo real, dando la posibilidad de visualizarla, tal y como se puede observar en la *Figura 26*.

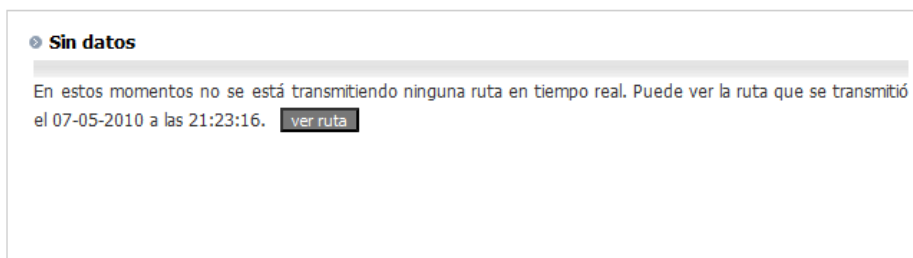


Figura 26: Visualización de trayectorias en tiempo real

En el momento que se inicia la transmisión de una ruta en tiempo real, con la correspondiente alta en la base de datos, en el recuadro de alerta de transmisión en tiempo real de la página principal del proyecto se avisa de esta actividad, posibilitando el acceso a su visualización. Como ya se ha comentado, cuando se detecta una **a** en el campo *tipo* de cualquiera de los registros de la base de datos es cuando aparece el aviso de transmisión en tiempo real en la página principal, y desaparece cuando este campo toma el valor **t**.

Mientras el estado del campo *tipo* de la ruta visualizada en tiempo real es **a** se va refrescando el marco o *iframe* que contiene el mapa con una frecuencia aproximada de 5 segundos, y de esta forma se permite ir visualizando el movimiento del vehículo en tiempo real.

5.4 Gestión de rutas



© Copyright Jesús Iglesias García 2010

Figura 27: Página de acceso a la gestión de rutas

En esta sección de la página web se permite manipular el contenido de la base de datos y los ficheros o archivos de ruta almacenados en el FTP del servidor. No

obstante, para poder acceder a estas opciones se precisa de un nombre de usuario y una contraseña, ya que se ha establecido que esta sección se ejecute bajo el inicio de una sesión con autenticación de usuario, tal como se muestra en la imagen de la *Figura 27*.

5.4.1 Alta de rutas

Una vez que el usuario se haya autenticado y haya conseguido acceder al apartado de gestión de rutas, una de las acciones que podrá realizar es el alta de nuevas rutas a través de un botón situado en la página principal de la gestión de rutas (Añadir nueva ruta). A través de este botón se accede al formulario que aparece en la parte inferior derecha de la *Figura 28*, cargando rutas que se encuentren almacenadas en el dispositivo informático desde el cual nos conectamos a la página web.

etse UAB
Universitat Autònoma de Barcelona

Inicio Rutas **Gestión de rutas** Tiempo real Comunicación PDA

Filtrar por...

- Rutas Simples
- Rutas con información de velocidad
- Comparación de sistemas
- Todas las rutas

Nombre de la ruta:

Últimas rutas cargadas

- [07-05-10] · 21:23 martorell
- [02-05-10] · 01:05 Ruta con 50 puntos
- [26-04-10] · 16:18 Barcelona - Collserola

Trasmisión de Datos PDA-Servidor Web

Gestión de Rutas

Cargar una nueva ruta en la página web.

Cargar nueva ruta

Tipo de ruta:
Ruta comparativa GPS tradicional

Nombre de la ruta:

Archivo de ruta:
 No se ha... archivo

Archivo de ruta (GPS tradicional):
 No se ha... archivo

jesus | Salir

© Copyright Jesús Iglesias García 2010 Trabajo Final de Carrera - Ingeniería Técnica de Telecomunicaciones Especializada en Sistemas Electrónicos

Figura 28: Página de gestión para cargar nuevas rutas

Este formulario permite cargar nuevos archivos de ruta en el servidor y darlos de alta en la base de datos simultáneamente, con los datos introducidos en éste. En el formulario aparecen uno o dos campos del tipo *file*, en función del tipo de ruta que se vaya a cargar. Esto se debe a que las “Rutas comparativas GPS tradicional” precisan dos archivos de ruta, uno con la ruta obtenida a partir de los datos obtenidos de un sistema GPS tradicional, y el otro que contiene los datos obtenidos de sistema GPS/INS.

5.4.2 Modificación de rutas

Como en el caso anterior, se dispone de un formulario (*Figura 29*), aunque en este caso éste ya contiene una serie de datos; los datos originarios de la ruta que hemos escogido para modificar.



Figura 29: Página de gestión para modificar rutas

Para efectuar la modificación de los datos de la ruta en cuestión, basta con cambiar el valor de los campos del formulario que se desee y pulsar en el botón de “Modificar ruta”.

5.4.3 Baja de rutas

En el momento que se decide dar de baja una de las rutas de la base de datos aparecen todos los datos relacionados a la ruta seleccionada (*Figura 30*), con el fin de que el usuario se asegure de que realmente desea eliminar dicha ruta.

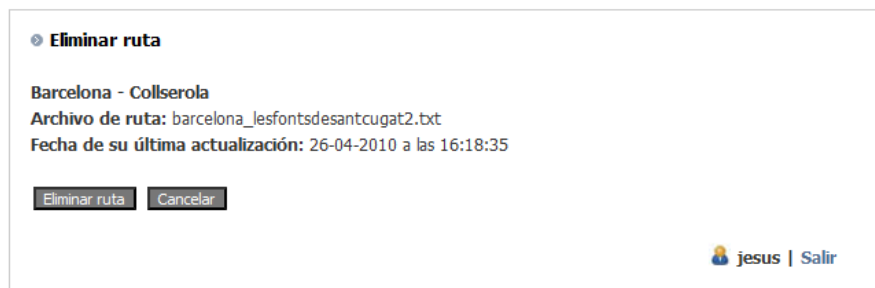


Figura 30: Página de gestión para eliminar rutas

Una vez se decide que la ruta escogida es la que se desea eliminar y se ejecuta la acción, ésta es dada de baja de la base de datos y el archivo de ruta relacionado se elimina del servidor.

5.5 Comunicación interfaz web – PDA



Figura 31: Página de comunicación con la PDA

Este apartado de la aplicación web permite transmitir datos hacia la PDA. Se disponen dos dispositivos del vehículo que pueden estar activos o inactivos y se permite cambiar el estado de éstos desde la página web. Como se puede observar en la *Figura 31* se dispone de un recuadro con una serie de botones que permiten activar/desactivar los dispositivos mencionados.

Para realizar el cambio de estado de estos dispositivos se han creado dos ficheros en el servidor, uno por cada dispositivo. En estos ficheros se guarda el estado de los dispositivos. A partir de esta página se cambia el estado de los dispositivos y, por tanto, el contenido de los ficheros.

Dichos ficheros son los que se leen desde el apartado de control de la interfaz de la PDA y se muestra el cambio de estado realizado desde la interfaz web. Esto mismo es lo que se muestra en el recuadro inferior derecho de la *Figura 31*, la pantalla de la PDA donde se muestra el cambio de estado de los dispositivos realizado.

5.6 Testeo y resultados finales

A lo largo del desarrollo del proyecto se han ido realizando diferentes pruebas y simulaciones con el fin de verificar el funcionamiento del sistema implementado, y ver si se cumple con los objetivos propuestos en un inicio.

Se pueden observar los resultados finales de la aplicación web implementada en la siguiente URL: <http://www.elperrodepavlov.com/PFC/>.

A continuación se muestra un resumen de los ensayos realizados, detallando los logros conseguidos y las especificaciones del sistema obtenidas.

5.6.1 Transmisión de archivos de ruta PDA-Servidor Web

Después de realizar numerosas pruebas de envío de trayectorias, con la aplicación se ha conseguido cargar archivos de ruta (.txt) almacenados previamente en la PDA, en el directorio del servidor web que se haya fijado mediante la configuración de los datos del servidor. De esta forma se ha conseguido la transmisión de este tipo de archivos a través de la red WIFI disponible en el lugar que se encuentre en el momento de la transmisión. Por otro lado, el sencillo formato de los archivos de ruta permite una transmisión ágil y rápida

5.6.2 Visualización de trayectorias en Google Maps

Para realizar la prueba del sistema de visualización de trayectorias sobre *Google Maps* se utilizan archivos de ruta de GPS (.gpx), previamente descargados de la página web *wikiloc* [14], un portal web que recopila y procesa información geográfica recogida por dispositivos GPS (Waypoints, rutas, etc...) aportada por los usuarios del portal, para agregarla y ponerla al alcance de otros usuarios.

Mediante una rutina programada en *Matlab* se extraen los puntos geográficos (latitud, longitud) del archivo .gpx y se guardan en un archivo .txt, que es utilizado por la interfaz web creada en este proyecto.

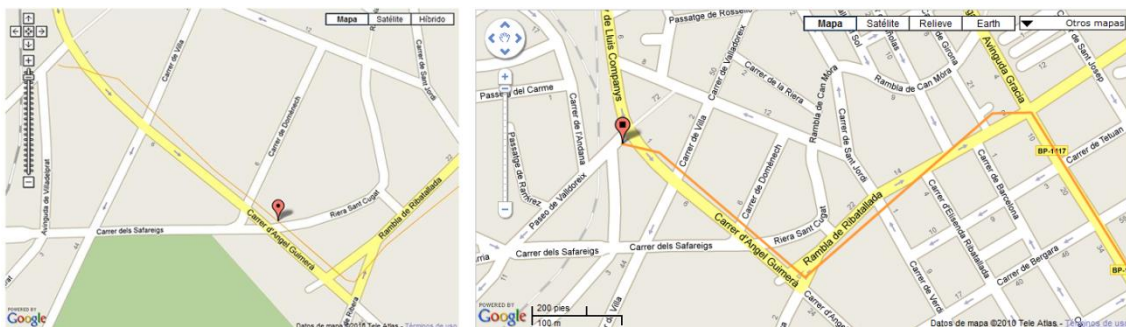


Figura 32: Comparación visualización de ruta en wikiloc vs. Interfaz web creada

En la *Figura 32* se pueden observar dos mapas correspondientes a una misma ruta. El mapa de la izquierda es el generado por la interfaz web implementada en este proyecto, y el de la derecha el que se puede visualizar en el portal de internet *wikiloc*. Se puede ver como la precisión obtenida es prácticamente la misma, con el añadido de disponer de un mayor nivel de zoom.

A partir de esta visualización básica sobre un mapa de Google Maps, se han desarrollado varios tipos de visualización de trayectorias aportando diferentes niveles de información.

En la *Figura 33* se puede observar un mapa de trayectoria que compara la trayectoria del vehículo obtenida por dos sistemas diferentes, facilitando así la comparación de estos. Además también se indica la distancia total de cada una de las dos trayectorias en la parte superior izquierda.

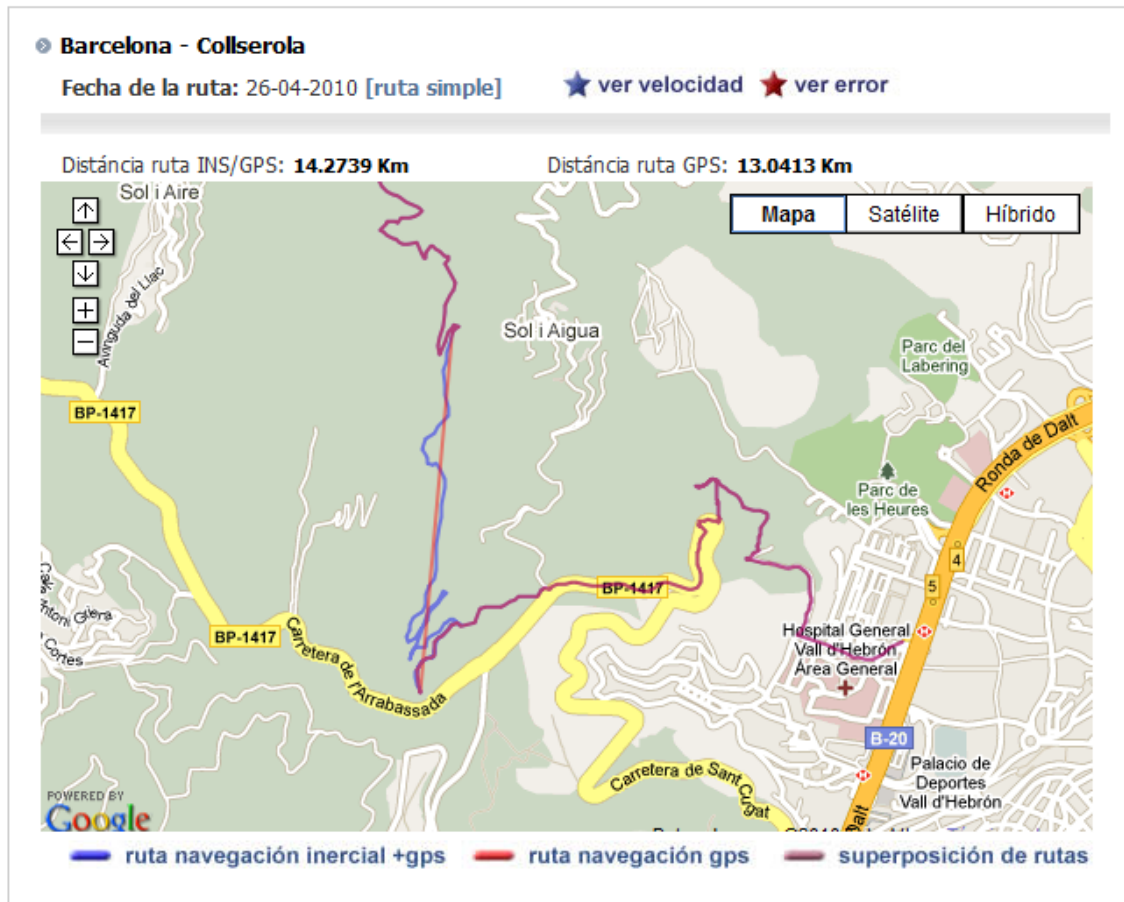


Figura 33: Mapa con comparación de trayectorias

A partir de estos resultados se puede decir que la integración del *API* de *Google Maps* en la aplicación web creada es satisfactoria, obteniendo unos resultados totalmente correctos.

5.6.3 Visualización de trayectorias en tiempo real

En cuanto a la visualización en tiempo real, desde el momento en que se inicia la transmisión en tiempo real en la PDA hasta que el recuadro de alerta de la página principal de la interfaz web avisa al usuario, transcurren entre 1 y 2 segundos, un tiempo de respuesta correcto, teniendo en cuenta que en la rutina/código que genera dicha alerta se ejecuta cada segundo.

Por otro lado tenemos la simulación de transmisión de datos en tiempo real que se realiza desde la PDA. Esta simulación divide el conjunto de puntos geográficos que contiene el archivo de ruta .txt en pequeños paquetes de 20 puntos cada uno, y cada 50ms se añade un paquete de 20 puntos al paquete a transmitir, de tal forma que en la interfaz web se va recibiendo la información de la ruta progresivamente, creando el efecto de movimiento del vehículo con el correspondiente muestreo de la trayectoria en tiempo real.

Después de realizar varios envíos con archivos de ruta de diferentes tamaños se obtiene una velocidad media de transmisión PDA-Servidor Web de 150 puntos/segundo, y teniendo en cuenta que la aplicación web que muestra la trayectoria en tiempo real se refresca cada 5 segundos aproximadamente es un valor que cumple con margen de sobras la velocidad de transmisión necesaria para que la visualización sea en tiempo real.

Teniendo en cuenta que 150 puntos equivale a unos 3,08 Km, se está ofreciendo la precisión correspondiente a la captación del movimiento de un vehículo que circule a un máximo de 3,08 Km/s, o lo que es lo mismo, 11.070 Km/h.

5.6.4 Transmisión de datos Servidor Web-PDA

Una vez cumplidos los objetivos planteados en un inicio se presenta la posibilidad de implementar la opción de transmisión de datos del servidor web hacia la PDA, lo que implica la posibilidad de tener cierto control sobre algunas de las características del vehículo desde la página web.

A partir del formulario expuesto en el recuadro inferior izquierdo de la *Figura 31* se permite modificar el estado de dos dispositivos del vehículo. En el apartado de “Control” de la interfaz de la PDA se muestra el estado de estos dos dispositivos, con lo que en el momento que se realice cualquier cambio a través de la página web se verá reflejado inmediatamente en la PDA, tal y como se puede observar en la *Figura 34*.



Figura 34: Pestaña de Control de la interfaz de usuario de la PDA

5.7 Problemas y soluciones en el desarrollo del proyecto

Durante el desarrollo del sistema propuesto en este proyecto se han encontrado diferentes complicaciones que han hecho replantear diferentes aspectos del trabajo. A continuación se detallan los problemas encontrados más relevantes.

Visualización de un mapa con la trayectoria graficada en la PDA

En el momento de desarrollar la interfaz de usuario de la PDA se decide crear una opción que permita **visualizar en la misma PDA la última ruta cargada en el servidor**. Lo primero que se hace es crear una página web con las dimensiones adecuadas para que se visualice correctamente en la PDA y se carga la última ruta cargada en el servidor mediante una consulta a la base de datos.

El problema surge a la hora de hacer la primera prueba, pues al acceder a esta opción en la aplicación programada **en la PDA no aparece nada**. Después de realizar varias consultas por internet y en foros especializados se llega a la conclusión de que el navegador del que dispone la PDA no es capaz de interpretar los scripts de la página que carga el mapa de la última ruta cargada en el servidor.

La versión de **IE Explorer** de la que dispone la **PDA no es capaz de cargar los mapas de Google**, por ello después de haber comprobado que no existe ninguna actualización de este navegador que lo permita, se decide buscar una alternativa. Finalmente se da con la aplicación **Skyfire** [13], un **navegador para dispositivos móviles y PocketPC** que permite interpretar los scripts que IE Explorer no permite.

Error de precisión del marcador sobre la trayectoria de ruta

En el tipo de visualización de ruta con información de la velocidad del vehículo en cada punto de la trayectoria realizada, se utiliza un marcador/puntero para ir desplazándose a lo largo de la trayectoria graficada e ir viendo la velocidad asociada a cada punto. El problema que se encuentra en este momento es que si se amplía un poco el mapa, se puede observar como **el marcador no recae precisamente sobre la trayectoria**, sino que se produce cierto error de precisión, tal como se observa en la *Figura 35*.



Figura 35: Error del marcador que recorre la trayectoria

Este error se debe al redondeo o aproximación de la latitud y longitud que provoca la función utilizada para ubicar el marcador en el punto concreto de la trayectoria. La solución que se aplica es el **cálculo de distancias mínimas**. Se dispone de un vector de datos con todos los puntos que forman la trayectoria y en el momento que se desplaza el puntero del ratón se ejecuta la función que **calcula el punto de la trayectoria más cercano al marcador** o puntero del ratón.

Actualizar una parte del contenido de una página sin tener que actualizarla por completo

Tal y como se ha comentado anteriormente, en la página principal se dispone de un recuadro donde se alerta del inicio de una transmisión en tiempo real. Ya que el funcionamiento de este sistema de alerta se basa en la continua comprobación del contenido de la base de datos, se precisa de una actualización periódica de este recuadro. El problema que se presenta es que esto implica la **continua actualización de la página completa**, acción que genera una serie de parpadeos en la página y demasiado tráfico, ya que se está cargando la página correctamente.

La primera solución que se encuentra pasa por la **colocación de dicho recuadro de alerta en un *iframe***, y entonces sólo actualizar este *iframe*. No obstante, aunque de esta manera ya no es necesario recargar la página por completo continuamente, este recuadro muestra un continuo parpadeo. Es entonces cuando se da con la solución definitiva: **utilizar el refresco del contenido de este *iframe* mediante código AJAX**.

```
<head>
<script type="text/javascript">

    function objAjax(){
        var xmlhttp=false;
        try {
            xmlhttp = new ActiveXObject("Msxml2.XMLHTTP");
        } catch (e) {
            try {
                xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
            } catch (E) {
                xmlhttp = false;
            }
        }

        if (!xmlhttp && typeof XMLHttpRequest!='undefined') {
            xmlhttp = new XMLHttpRequest();
        }
        return xmlhttp;
    }

    function xRefresh(xPage,xDiv){

        divResultado = document.getElementById(xDiv);
        ajax=objAjax();
        ajax.open("GET",xPage);
        ajax.onreadystatechange=function() {

            if (ajax.readyState==4) {
                divResultado.innerHTML = ajax.responseText
            }

        }

    }
```



```
        ajax.send(null)
        setTimeout("xRefresh('e_trans.php','resultado')",1000)
    }

</script>

</head>
<body onLoad="xRefresh('e_trans.php','resultado')">
    <div id="resultado" style="border: 0px; width:285px"></div>
</body>
```

Fallo en la visualización de trayectorias en tiempo real

Ya se ha explicado anteriormente que la simulación del envío en tiempo real que se realiza en la PDA trabaja con un mismo archivo de ruta durante todo el proceso, lo que implica que se vaya sustituyendo continuamente el archivo de ruta que se aloja en el servidor por el mismo actualizado con un paquete de puntos más.

Esto provoca que el archivo de ruta en tiempo real que ya se encuentra en el servidor esté continuamente en actualización hasta el momento que finaliza la transmisión. Teniendo en cuenta que la página de visualización de trayectorias en tiempo real se actualiza cada 5 segundos, recargando cada vez el contenido de dicho archivo de ruta, **si en el momento que se lee este archivo desde la página web éste está siendo actualizado por la PDA se produce un error y el recuadro que contiene el mapa en la interfaz web aparece en blanco.**

La solución a este error se encuentra con la **comprobación del estado del archivo en cuestión en el momento de abrirlo para realizar su lectura en la interfaz web.** De esta manera se detecta si el archivo está siendo actualizado por la PDA y si es así se espera a que ésta termine.

```
if (file_exists($ruta))
{
    $fclosed=1;
}
else{
    $fclosed=0;
    echo("refrescar()");
}
```

6. Planificación del proyecto

A continuación se detallan los requisitos que se deben cumplir para poder llevar a cabo el trabajo, las herramientas de desarrollo que se emplean, así como las diferentes tareas que se realizan durante el desarrollo de la solución, y los tiempos estimados para cada una de ellas.

6.1 Diagrama de Gantt

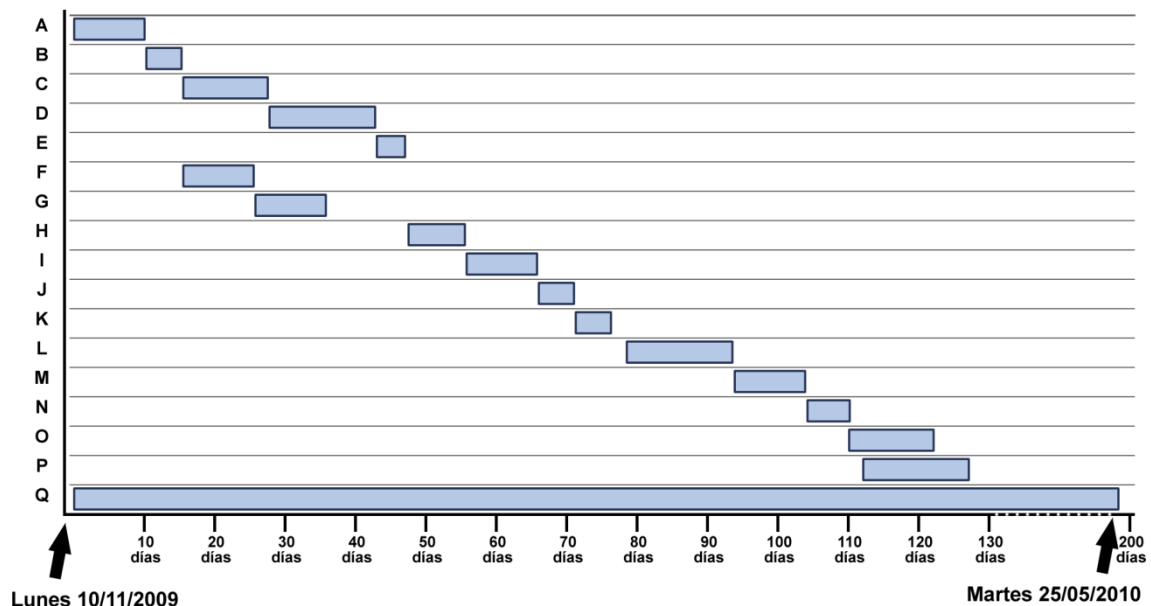


Figura 36: Diagrama de Gantt

El desarrollo del proyecto se ha realizado dedicando una media de 4 horas diarias de lunes a viernes, teniendo en cuenta las fechas de inicio i fin que se muestran en la *Figura 36*. De aquí obtenemos una dedicación aproximada de 600h de trabajo.

6.2 Fases del proyecto

Aunque son evidentes las dos grandes tareas de las que consta este proyecto (creación de la interfaz de usuario para la pda y de la interfaz web), se va a detallar el conjunto de fases que llevarán a la conclusión de éstas.

- Estudiar y analizar el tipo de datos con los que se trabajará, en este caso, puntos de posición geográfica (latitud,longitud), con la posibilidad de añadir más variables, como por ejemplo la velocidad del vehículo.
- Analizar las características de la PDA de la que disponemos y escoger la plataforma de programación más conveniente.
- Familiarización con el lenguaje de la plataforma de programación escogida y programación de la interfaz de usuario que permitirá la transmisión entre la PDA y un servidor web remoto.

- Realizar la conexión con el FTP del servidor para poder enviar los datos relativos a la trayectoria del vehículo.
 - Diseñar y programar un formulario/ventana para cada una de las siguientes funciones: envío de ruta, control (recepción de datos), configuración del servidor y salida del programa.
 - Programar la simulación de un envío de datos en tiempo real.
 - Pruebas y depuración de código.
- Diseño y programación de una interfaz web que gestionará la información recibida.
 - Recepción y almacenamiento de los datos transmitidos por la PDA.
 - Graficar los datos recibidos en un mapa de *Google Maps*.
 - Crear las páginas necesarias con tal de permitir la gestión de los datos (trayectorias) almacenados.
 - Crear una página que permita la transmisión de datos hacia la PDA.
 - Crear un rol de pruebas y aplicarlo a la interfaz web con el fin de realizar todas las pruebas de funcionamiento necesarias y depurar posibles errores.
 - Redacción de memorias.

6.3 Tiempos de realización

El proyecto se inicia el lunes 10 de noviembre de 2009, después de su asignación. El desarrollo de las diferentes tareas que lo conforman (*Tabla 5*) se realiza durante los seis meses siguientes, finalizando el 25 de mayo de 2010.

| Tarea | Descripción | Predecesoras | Tiempo |
|-------|---------------------------------------------------------------------------------------------|--------------|-----------------|
| A | Estudio de la situación y definición de objetivos | - | 10 días |
| B | Adquisición de conocimientos básicos de Visual Studio | A | 5 días |
| C | Conectar la PDA con el servidor FTP | B | 12 días |
| D | Diseño y programación de los diferentes formularios de la aplicación | C | 15 días |
| E | Pruebas de envío de archivos de ruta | D | 4 días |
| F | Programación y pruebas de la página que procesa la alta de las rutas recibidas desde la PDA | B | 10 días |
| G | Estudio del funcionamiento del API de Google Maps | F | 10 días |
| H | Visualización de un archivo de ruta en un mapa de Google Maps | E | 8 días |
| I | Añadir datos sobre la velocidad del vehículo a la ruta | H | 10 días |
| J | Visualización de comparación de rutas | I | 5 días |
| K | Programación de la simulación de envío en tiempo real en la interfaz de la PDA | J | 5 días |
| L | Programación de la visualización en tiempo real en la página web | J + 2 días | 15 días |
| M | Pruebas y depuración de códigos | L | 10 días |
| N | Desarrollo de la aplicación online que permite transmitir datos hacia la PDA | M | 6 días |
| O | Testeo y pruebas de todo el sistema | N | 12 días |
| P | Depuración de errores y cambios | N + 2 días | 15 días |
| Q | Redacción de memorias | - | 6 meses y medio |

Tabla 5: Relación de tareas y tiempos

Cabe destacar que aunque el proyecto es realizado por una única persona y por tanto las tareas se ejecutan de modo secuencial, existen algunas tareas que se realizan paralelamente a otras, como es el caso de la redacción de memorias, que se lleva a cabo durante todo el tiempo de desarrollo del proyecto.

6.4 Requerimientos

Para poder llevar a cabo este proyecto se precisa de una serie de materiales y conocimientos. En cuanto a los materiales que se necesitan, tenemos la PDA con la cual transmitiremos los datos de posicionamiento, un servidor web (*hosting*) y un PC con el cual programar las aplicaciones necesarias para el sistema propuesto.

La PDA



DELL Axim X51v

| | |
|--------------------------|---------------------------------------------|
| Procesador | Intel XScale PXA270, 624 Mhz |
| Memoria RAM | 64 MB |
| Puertos expansión | SDIO (SD Card / MMC) y CompactFlash Tipo II |
| Sistema operativo | Microsoft Windows Mobile 5.0 |
| Pantalla | 480 x 640 píxeles, 65536 colores |

El hosting

Apache/2.2.15 (Unix) mod_ssl/2.2.15
OpenSSL/0.9.8e-fips-rhel5 DAV/2 PHP/5.2.13



| | |
|------------------------------|---------------------------------------------|
| Procesador | QuadCore con 4Gb de RAM velocidad de 3Tb |
| Espacio en disco | 200 MB |
| Sistema Operativo | Linux (Fedora) |
| BBDD | phpMyAdmin (MySQL v 5.1.47) |
| Transferencia mensual | 2 GB |

El PC

Un Intel Dual Core 2GHz - 4Gb de RAM, dotado del software necesario para el desarrollo de la interfaz de usuario de la PDA, y de la página web.

6.5 Herramientas de desarrollo

A continuación se exponen las diferentes herramientas de desarrollo o plataformas de programación para el caso, empleadas para crear el software necesario para el sistema implementado.

6.5.1 Microsoft Visual Studio 2005

Mediante esta plataforma de programación se crea la interfaz de usuario para la PDA que permite navegar por los diferentes directorios locales para escoger el archivo de ruta que se desea enviar al servidor web y transmitirlo vía WIFI.

Para realizar la conexión con el FTP remoto y transmitir los archivos de ruta se emplea la librería que se ha comentado anteriormente, *Librería ClienteFTP para PC y PDA v1.1.0.0*. [3], la cual permite establecer una conexión segura.

6.5.2 Plataforma de desarrollo web

Para llevar a cabo el desarrollo de la aplicación web se emplea como plataforma de programación principal la aplicación de Adobe denominada Dreamweaver. Esta plataforma de programación permite programar en los diferentes lenguajes (html, php, javascript, css, ajax) necesarios para el desarrollo del proyecto facilitando la generación del código.

Para la creación de la base de datos que gestiona todas las rutas cargadas en el servidor web se hace uso de la aplicación web **phpMyAdmin** (*Figura 37*). Esta es una herramienta escrita en PHP con la intención de manejar la administración de *MySQL* a través de páginas web, utilizando Internet. Actualmente permite crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 50 idiomas.

The screenshot displays the phpMyAdmin interface. On the left is a sidebar with a tree view of the database structure, including tables like 'deluxebb_attachments', 'deluxebb_banned', etc. The main area shows a table with columns: id, archivo, nombre, tipo, and fecha. The table contains 15 records, with the first few visible as follows:

| id | archivo | nombre | tipo | fecha |
|-----|-------------------------------------------------------|--------------------------------------------|------|---------------------|
| 65 | TerrassaCanDeuCastellardeVallesSantFeliudelRacoMa... | Terrassa - Castell del Vallès | r | 2010-04-11 15:01:16 |
| 64 | Mollet-Prehistorica-Mollet.txt | Mollet - Prehistòrica - Mollet | r | 2010-04-11 15:00:16 |
| 63 | SabadellManresa.txt | Sabadell Manresa | r | 2010-04-07 12:59:20 |
| 62 | GranollersVilanovaTuronsdeCanBoscVelCanTarascCelle... | Granollers - Turó de Mataró | r | 2010-04-06 14:58:04 |
| 67 | GarrafSitges.txt | Garraf - Sitges | e | 2010-04-11 15:03:14 |
| 66 | GarrafValgrassaPlanaNovellaJafra.txt | Valgrassa (Garraf) - Plana Novella (Jafra) | r | 2010-04-11 15:02:31 |
| 58 | Palau_tordera_montseny.txt | Palautordera - Montseny | e | 2010-04-01 11:54:27 |
| 60 | barcelona_lesfontsdessancugat.txt | Les fonts de Collserola | v | 2010-04-02 14:56:09 |
| 61 | CanCollPorElPantanoCollserola.txt | Can Coll por el pantano de Collserola | r | 2010-04-05 18:57:20 |
| 97 | prueba11.txt | Vall d'Hebrón - Sant Cugat | t | 2010-04-25 11:14:26 |
| 99 | SabadellManresa2.txt | Sabadell - Manresa II | r | 2010-04-25 12:01:01 |
| 100 | GarrafSitges3.txt | Sitges - Garraf | e | 2010-04-25 20:32:41 |
| 101 | barcelona_lesfontsdessancugat2.txt | Barcelona - Collserola | w | 2010-04-26 16:18:35 |
| 120 | martorell.txt | martorell | t | 2010-05-07 21:23:16 |

Figura 37: Aplicación web phpMyAdmin [11]

La aplicación en si no es más que un conjunto de archivos escritos en PHP que se puede copiar directamente a un directorio del servidor web en el que se aloja la página web, de modo que, cuando se accede a estos archivos, se muestran unas páginas donde es posible encontrar las bases de datos a las que se tienen acceso en el servidor de bases de datos y todas sus tablas.

7. Conclusiones

A partir del testeo y de la verificación del funcionamiento del sistema implementado en este proyecto se han obtenido unos resultados satisfactorios que cumplen con los objetivos planteados en un principio.

Se ha conseguido implementar una interfaz de usuario en la PDA que no solo permite transmitir datos al servidor web vía WIFI, sino que además se presta a la posibilidad de configurar el acceso a este servidor, recibir datos enviados desde la aplicación web e incluso, permitir la visualización de una ruta graficada en *Google Maps* en la misma PDA.

La interfaz web creada va más allá de la simple visualización de las trayectorias almacenadas en los archivos de ruta recibidos de la PDA. Ofrece una multitud de opciones adicionales de análisis y gestión de estos datos, permitiendo así:

- Realizar búsquedas en la base de datos con el fin de visualizar una trayectoria concreta a partir de su nombre.
- Comparar trayectorias generadas a partir de los datos obtenidos de un sistema GPS tradicional con los del sistema GPS/INS.
- Visualizar la trayectoria que realiza un vehículo en tiempo real.
- Añadir, modificar y eliminar trayectorias de la base de datos y del servidor desde la página web.
- Modificar el estado de dispositivos del vehículo desde la interfaz web (transmisión Servidor Web-PDA).
- Asociar información adicional sobre el vehículo a cada punto geográfico de la trayectoria (Latitud, Longitud), como es el caso de la velocidad.

Todas estas características del sistema implementado, que superan los objetivos planteados, suponen un valor añadido para la solución tecnológica creada. Se obtiene entonces un sistema de transmisión de datos de posicionamiento a partir de un dispositivo informático y la visualización de estos en una interfaz de usuario *online* en una de las plataformas SIG más conocida y empleada de la red de redes, *Google Maps*.

El sistema obtenido finalmente ofrece una gran versatilidad y una facilidad de uso, presentando toda la información recogida en un entorno que combina sencillez, claridad y usabilidad.

8. Futuras ampliaciones

A pesar de todos los beneficios que ofrece el sistema creado, cabe destacar que existen una serie de limitaciones que presenta la utilización de una PDA con el sistema operativo *Windows Mobile 2005* frente a las tecnologías de vanguardia, como por ejemplo *Android*[15], un sistema operativo orientado a dispositivos móviles basado en una versión modificada del núcleo Linux, y que actualmente se encuentra en posesión de la compañía *Google*. Un claro ejemplo de estas limitaciones está en que un dispositivo móvil con el sistema operativo *Android* y una pantalla de unas dimensiones razonables permitiría la visualización tanto de los mapas generados en la interfaz web como el conjunto completo de la interfaz web.

De todas formas, a pesar de las limitaciones, este sistema es fácilmente adaptable a uno de estos dispositivos móviles más vanguardistas, ya que en su mayoría está asentado en una aplicación web compatible.

Una de las mejoras más notables que se podrían realizar sería cambiar la forma de transmisión de los datos de posicionamiento. En el proyecto realizado se utiliza la comunicación WIFI para la transmisión de datos hacia el servidor web. Aunque éste es un sistema de comunicación que se encuentra en auge actualmente, y cada vez existen más redes de acceso gratuito de este tipo, si se utilizase una transmisión mediante GPRS (*General Packet Radio Service*) se ampliaría la cobertura notablemente, sin depender de puntos de acceso WIFI. De esta forma también estaríamos hablando de integrar todo el sistema en un mismo dispositivo: la interfaz que carga los datos en el servidor web y la aplicación web que permite visualizarlos.

9. Bibliografía y/o recursos electrónicos

- [1] GARCÍA QUINCHÍA, Alex. *Integración GPS&INS Mediante DSP/FPGA para Aplicaciones Aeroespaciales*. Director: Carles Ferrer Ramis. Universidad Autónoma de Barcelona, Departamento de Microelectrónica y Sistemas Electrónicos, 2010.
- [2] PIMPLER, Eric. *Google Maps Api: The New World of Web Mapping*. GeoSpatial Training & Consulting, 2006.
- [3] ESQUIVEL, Manuel. *Librería ClienteFTP para PC y PDA v1.1.0.0*. Otaku Software MX, 2009.
(<http://www.otakusoftwaremx.hostzi.com/index.php?module=showTuto&Id=7&hash=388>)
Consultado el: 26/11/2009 – Última consulta: 7/04/2010
- [4] Foro de Smartphones Windows Mobile. Foro Todo Pocket PC
(<http://foro.todopocketpc.com/forumdisplay.php?f=39>)
Consultado el: 12/11/2009 – Última consulta: 2/12/2009
- [5] Foro de programación de dispositivos móviles. Foro Todo Pocket PC
(<http://foro.todopocketpc.com/forumdisplay.php?f=22>)
Consultado el: 11/01/2010 – Última consulta: 26/01/2010
- [6] API de Google Maps. Documentación oficial y grupo de noticias sobre el desarrollo de aplicaciones con el API de Google Maps.
(<http://code.google.com/intl/es-ES/apis/maps/>)
Consultado el: 3/02/2010 – Última consulta: 29/05/2010
- [7] Página web oficial del lenguaje PHP. Documentación, ejemplos, manuales...
(<http://php.net/index.php>)
Consultado el: 10/02/2010 – Última consulta: 06/06/2010
- [8] Tutoriales completos de programación de páginas web. Tutorial HTML
(<http://www.w3schools.com/html/default.asp>)
Consultado el: 10/02/2010 – Última consulta: 11/05/2010
- [9] Tutoriales completos de programación de páginas web. Tutorial JavaScript
(<http://www.w3schools.com/js/default.as>)
Consultado el: 3/02/2010 – Última consulta: 30/05/2010
- [10] Tutorial PHP MySQL. Inserción de información en una base de datos. Desarrolloweb, 2002
(<http://www.desarrolloweb.com/articulos/323.php>)
Consultado el: 18/02/2010
- [11] Página web oficial de la aplicación web PHPMyAdmin. phpMyAdmin 2010
(http://www.phpmyadmin.net/home_page/index.php)
Consultado el: 26/05/2010
- [12] Ejemplo de la librería de utilidades de Google Maps. GoogleCode, 2010
(<http://gmaps-utility-library.googlecode.com/svn/trunk/snaptoroute/1.0/examples/demonstration.html>)
Consultado el: 16/03/2010

- [13] Página web oficial del navegador web para dispositivos móviles Skyfire. Skyfire Labs, Inc, 2010
(<http://www.skyfire.com/>)
Consultado el: 22/04/2010

- [14] Portal web de rutas geográficas para GPS. Wikiloc, 2010
(<http://es.wikiloc.com/wikiloc/home.do>)
Consultado el: 3/02/2010 – Última consulta: 26/05/2010

- [15] Página web oficial de la plataforma Android. Google Inc. 2010
(<http://www.android.com/>)
Consultado el: 26/05/2010

10. Anexo

10.1 Código fuente de la aplicación de la PDA

```
Imports System 'Importación de librerías
Imports System.IO 'Importación de librerías
Imports System.Drawing 'Importación de librerías
Imports ClienteFTPPDA 'Importación de librería para trabajar con el
servidor FTP

Public Class Form1
    'Botón para cerrar el programa
    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs)
        Me.Close() 'Instrucción que finaliza el programa
    End Sub
    Dim MiFTP As ClienteFTP 'Creamos el objeto que servirá para
conectar con el servidor FTP mediante la librería ClienteFTPPDA
    Dim b As Boolean
    Dim psi As New ProcessStartInfo() 'Creamos el objeto que nos
servirá para ejecutar una ventana de IE
    Dim psi2 As New ProcessStartInfo() 'Creamos el objeto que nos
servirá para ejecutar una ventana de Skyfire (navegador que permite
visualizar mapas de Google Maps)
    Dim fechahora, hora, dia As String
    'Botón para enviar datos
    Private Sub Button8_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button8.Click

        fechahora = Date.Now
        dia = Date.Now.Year & "-" & Date.Now.Month & "-" &
Date.Now.Day & " "
        hora = Mid(fechahora, 10, 9)

        fechahora = dia & hora

        If TextBox1.Text = "" Or TextBox2.Text = "" Then
            TextBox3.Text = "Debe indicar el nombre de la ruta"
        Else
            TextBox3.Text = "" 'Limpiamos el contenido de la caja de
texto (Estado del proceso)
            b = False 'Con esta variable booleana en FALSE indicamos
que aún no se ha realizado el envío
            MiFTP = New ClienteFTP(server, user, pass, rute, 21)
            'Asignamos los datos de conexión al FTP
            If MiFTP.Login() Then 'Validamos la conexión con el
servidor FTP
                TextBox3.Text = "Enviando.." 'Indicamos que ha
comenzado el proceso de envío
                MiFTP.TipoServidor = True
                TextBox3.Text = "Enviando...."
                MiFTP.UploadFile(TextBox2.Text, False) 'Cargamos el
archivo en el servidor
                TextBox3.Text = "Enviando....."
                b = True 'Con esta variable booleana sabremos que el
envío se ha completado correctamente
            End If
            If b = True Then
                TextBox3.Text = "Envío finalizado correctamente"
```

```

        myUrl = "http://www." & server &
"/PFC/newroute.php?acces=pfc2010&docname=" & TextBox2.Text & "&nom=" &
TextBox1.Text & "&fechahora=" & fechahora 'Llamamos a la página que
dará de alta en la base de datos el archivo cargado mediante código
PHP y MySQL
        myUri = New Uri(myUrl)
        WebBrowser2.Navigate(myUri)
        TextBox1.Text = ""
        TextBox2.Text = ""
    Else
        TextBox3.Text = "Envío Incorrecto: Ha fallado la
conexión con el servidor. Comprueba que la conexión Wifi está activado
y vuelve a probar."
        TextBox1.Text = ""
        TextBox2.Text = ""
    End If
    MiFTP.LogOut() 'Finalizamos la conexión con el servidor
FTP
End If

End Sub
'Rutina de inicio:
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Button8.Enabled = False 'Inhabilitamos el botón de enviar.
    Button1.Enabled = False 'Inhabilitamos el botón de enviar en
tiempo real.
    Call cargar_configuracio()
    se.Text = server
    us.Text = user
    pa.Text = pass
    ru.Text = rute
    f = 0
End Sub
'Botón para salir del programa
Private Sub Button4_Click_1(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button4.Click
    Me.Close() 'Instrucción para cerrar y salir del programa
End Sub
'Botón para cargar un archivo .txt con la finalidad de enviarlo al
servidor FTP
Private Sub Button9_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button9.Click
    TextBox3.Text = "" 'Limpiamos el contenido de la caja de texto
(Estado del proceso)
    'Instrucción que solo permite abrir archivos .txt
    OpenFileDialog1.Filter = "Arxius de text (*.txt)|*.txt" &
"|Todos los tipos de archivo (*.*)|*.*"
    If OpenFileDialog1.ShowDialog = Windows.Forms.DialogResult.OK
Then 'Instrucción para mostrar los archivos que hay en el dispositivo
        TextBox2.Text = OpenFileDialog1.FileName 'Muestra en el
TextBox el archivo seleccionado
        OpenFileDialog1.Dispose() 'Liberamos los recursos del
FileDialog
        Button8.Enabled = True 'Habilitamos el botón de enviar.
    Else
        MsgBox("No hay ningún archivo registrado",
MsgBoxStyle.Exclamation) 'Mensaje de error: no existe ningún archivo
    End If
End Sub
Dim m As Integer

```

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button1.Click
    m = MsgBox("Está seguro de que desea transmitir en tiempo
real?", MsgBoxStyle.OkCancel, "Transmisión en tiempo real")

    If m = 1 Then
        Call envio_tiempo_real() 'Llamamos a la función que
emulará la transmisión de una ruta en tiempo real
        m = 0
    End If

End Sub

Dim coords As String
Dim tramo As String
Dim ruta As String
Dim myUrl As String
Dim myUri As Uri
Dim i, c, p, ps, total, parcial, media, n, f As Integer
Const fic1 As String = "\Archivos de
programa\pda_ftp_connection\docs\RUTA_COMPLETA.txt" 'Este archivo txt
contiene todos los puntos de la ruta que vamos a enviar en tiempo real
Const fic2 As String = "\Archivos de
programa\pda_ftp_connection\docs\COORDENADAS_TIEMPO_REAL.txt" 'Este
archivo estará vacío en un inicio y se irá completando con
puntos/coordenadas y enviando para emular la transmisión en tiempo
real
Dim sw As New System.IO.StreamReader(fic1, True)
Dim sw2 As New System.IO.StreamWriter(fic2, False)
Public Sub envio_tiempo_real()

    If f = 0 Then
        MiFTP = New ClienteFTP(server, user, pass, rute, 21)
'Asignamos los datos de conexión al FTP
        f = f + 1
    End If
    sw2.Write("") 'Limpiamos el contenido del archivo que vamos a
ir completando y enviando progresivamente
'Iniciamos variables
    coords = ""
    tramo = ""
    i = 1
    c = 0
    p = 0
    ms = 0
    total = 0
    parcial = 0
    media = 0
    n = 0
    ruta = ""
    Timer3.Enabled = True
    coords = sw.ReadToEnd() 'Metemos todo el contenido del archivo
que contiene la ruta completa en la variable coords
    sw.Close() 'Cerramos el archivo

    'Los tres bucles siguientes servirán para acumular en la
variable tramo más puntos para transmitir
    While Mid(coords, i, 1) <> ","
        tramo = tramo & Mid(coords, i, 1)
        i = i + 1
    End While
    tramo = tramo & Mid(coords, i, 1)

```

```

i = i + 1
While Mid(coords, i, 1) <> ","
    tramo = tramo & Mid(coords, i, 1)
    i = i + 1
End While
tramo = tramo & Mid(coords, i, 1)
i = i + 1
While Mid(coords, i, 1) <> ","
    tramo = tramo & Mid(coords, i, 1)
    i = i + 1
End While
tramo = tramo & ","
p = p + 1

sw2.Write(tramo) 'Escribimos en el archivo que enviamos el
contenido de la variable tramo
TextBox2.Text = fic2 'Asignamos al TextBox el nombre del
archivo que enviamos
sw2.Close() 'Cerramos el archivo

TextBox3.Text = "" 'Limpiamos el contenido de la caja de texto
(Estado del proceso)
If MiFTP.Login() Then 'Validamos la conexión con el servidor
FTP
    TextBox3.Text = "Enviando.." 'Indicamos que ha comenzado
el proceso de envío
    MiFTP.TipoServidor = True
    TextBox3.Text = "Enviando...."
    MiFTP.UploadFile(TextBox2.Text, False) 'Cargamos el
archivo en el servidor
    TextBox3.Text = "Enviando....."
    b = True 'Con esta variable booleana sabremos que el envío
se ha completado correctamente
End If
MiFTP.LogOut() 'Finalizamos la conexión con el servidor FTP
fechahora = Date.Now
dia = Date.Now.Year & "-" & Date.Now.Month & "-" &
Date.Now.Day & " "
hora = Mid(fechahora, 10, 9)
fechahora = dia & hora
myUrl = "http://www." & server &
"/PFC/newroute.php?acces=pfc2010&docname=" & TextBox2.Text & "&nom=" &
TextBox1.Text & "&tipt=tt&fechahora=" & fechahora
myUri = New Uri(myUrl)
WebBrowser2.Navigate(myUri)
Timer1.Enabled = True

End Sub
'El objeto Timer llamará a la función envio_tramos() cada 50
milisegundos para enviar el archivo de coordenadas
Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
    Call envio_tramos() 'Llamamos a la función envio_tramos()
End Sub
Public Sub envio_tramos()

    total = Len(coords) 'Asignamos a la variable total la
longitud/tamaño/número de caracteres de la variable coords
    If i < total Then
        i = i + 1

```

```

While c < 40
    While Mid(coords, i, 1) <> ","
        tramo = tramo & Mid(coords, i, 1)
        i = i + 1
    End While
    p = p + (1 / 2)
    c = c + 1
    tramo = tramo & ","
    i = i + 1
    If (i = total) Or (i > total) Then
        media = media / n
        TextBox3.Text = "Envío de la ruta completado" & "
[ " & media & " p/s ]"
        fechahora = Date.Now
        dia = Date.Now.Year & "-" & Date.Now.Month & "-" &
Date.Now.Day & " "
        hora = Mid(fechahora, 10, 9)
        fechahora = dia & hora
        myUrl = "http://www." & server &
"/PFC/newroute.php?acces=pfc2010&docname=" & TextBox2.Text & "&nom=" &
TextBox1.Text & "&time=t&fechahora=" & fechahora
        myUri = New Uri(myUrl)
        WebBrowser2.Navigate(myUri)
        TextBox2.Text = ""
        TextBox1.Text = ""
        MiFTP.LogOut() 'Finalizamos la conexión con el
servidor FTP
        Timer1.Enabled = False
        Exit Sub
    End If
End While
c = 0
i = i - 1

Dim sw2 As New System.IO.StreamWriter(fic2, False)
sw2.Write(tramo)
sw2.Close()

parcial = (100 / total) * i 'Cálculo para averiguar el
porcentaje del archivo original enviado
ps = p / ms * 1000
TextBox3.Text = "Enviando [ " & parcial & "% ] - [ " & ps
& " p/s ]" 'Indicamos que ha comenzado el proceso de envío
n = n + 1
media = media + ps
MiFTP.TipoServidor = True
MiFTP.UploadFile(TextBox2.Text, False) 'Cargamos el
archivo en el servidor
MiFTP.LogOut() 'Finalizamos la conexión con el servidor
FTP
b = True 'Con esta variable booleana sabremos que el envío
se ha completado correctamente

Else
    media = media / n
    MiFTP.LogOut() 'Finalizamos la conexión con el servidor
FTP
    TextBox3.Text = "Envío de la ruta completado" & "
[ " & media & " p/s ]"
    fechahora = Date.Now

```

```

        dia = Date.Now.Year & "-" & Date.Now.Month & "-" &
Date.Now.Day & " "
        hora = Mid(fechahora, 10, 9)
        fechahora = dia & hora
        myUrl = "http://www." & server &
"/PFC/newroute.php?acces=pfc2010&docname=" & TextBox2.Text & "&nom=" &
TextBox1.Text & "&tipt=t&fechahora=" & fechahora
        myUri = New Uri(myUrl)
        WebBrowser2.Navigate(myUri)
        TextBox2.Text = ""
        TextBox1.Text = ""
        Timer1.Enabled = False
    End If

End Sub

Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs)
    WebBrowser1.Refresh()
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button2.Click
    psi2.FileName = "\Archivos de programa\Skyfire\Skyfire.exe"
'Indicamos el programa con el que vamos a comunicarnos
    psi2.Arguments = "http://www." & server & "/PFC/ruta_pda.php"
'Llamamos a la página que nos mostrará el mapa en la PDA
    Process.Start(psi2) 'Abrimos una ventana con el navegador
Skyfire
End Sub
Const fic3 As String = "\Archivos de
programa\pda_ftp_connection\docs\config.txt" 'Definimos el archivo que
contiene los datos de configuración para el acceso al servidor FTP
Dim sw3 As New System.IO.StreamReader(fic3, True)
Dim conf, server, user, pass, rute, confi As String
Dim v As Integer
'Botón para actualizar los datos de acceso al servidor FTP
Private Sub Button3_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button3.Click
    Dim sw4 As New System.IO.StreamWriter(fic3, False)
    confi = ""
    confi = se.Text & "," & us.Text & "," & pa.Text & "," &
ru.Text & "," 'Asignamos a la variable confi todos los datos de acceso
al FTP
    sw4.Write(confi) 'Escribimos en el archivo los datos
    sw4.Close() 'Cerramos el archivo
    MsgBox("Los datos de configuración se han actualizado
correctamente.", MsgBoxStyle.Information, "Configuración")

End Sub
'Esta función carga los datos de configuración del acceso al FTP
en el apartado correspondiente del programa
Public Sub cargar_configuracio()
    i = 1
    conf = sw3.ReadToEnd() 'Asignamos a la variable conf el
contenido del archivo que contiene los datos de acceso al FTP
    'Iniciamos variables
    server = ""
    user = ""
    pass = ""
    rute = ""

    While Mid(conf, i, 1) <> ","

```

```

        server = server & Mid(conf, i, 1)
        i = i + 1
    End While
    i = i + 1
    While Mid(conf, i, 1) <> ","
        user = user & Mid(conf, i, 1)
        i = i + 1
    End While
    i = i + 1
    While Mid(conf, i, 1) <> ","
        pass = pass & Mid(conf, i, 1)
        i = i + 1
    End While
    i = i + 1
    While Mid(conf, i, 1) <> ","
        rute = rute & Mid(conf, i, 1)
        i = i + 1
    End While
    sw3.Close() 'Cerramos el archivo
End Sub

'Timer2 refresca la pantalla de Control cada 2 segundos para
actualizar la información
Private Sub Timer2_Tick_1(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Timer2.Tick
    WebBrowser1.Refresh()
End Sub

'Botón que detiene el refresco de la pantalla de Control
Private Sub Button6_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button6.Click
    Timer2.Enabled = True
End Sub

'Botón que inicia el refresco de la pantalla de Control
Private Sub Button5_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles Button5.Click
    Timer2.Enabled = False
End Sub

Private Sub TextBox1_TextChanged(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles TextBox1.TextChanged
    If TextBox1.Text <> "" Then
        Button1.Enabled = True 'Habilitamos el botón de enviar en
tiempo real.
    Else
        Button1.Enabled = False 'Inhabilitamos el botón de enviar
en tiempo real.
    End If
End Sub

Dim ms As Integer
Private Sub Timer3_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer3.Tick
    ms = ms + 1
End Sub
End Class

Public Class ObjectDisposedExceptionTest

    Public Shared Sub Main()
        Dim ms As New MemoryStream(16)
        ms.Close()
        Try
            ms.ReadByte()

```



```

        Catch e As ObjectDisposedException
            Console.WriteLine("Caught: {0}", e.Message)
        End Try
    End Sub 'Main
End Class 'ObjectDisposedExceptionTest

```

10.2 Código fuente de la interfaz de usuario / página web

10.2.1 [newroute.php] Página recepción de rutas y alta en la base de datos

```

<html>
<head>
<title>PDA-FTP Connection</title>
<link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body bgcolor="#6D81A3">
<?
include("pages/bdtfc.phtml");
$link=Conectarse();

$acc = $HTTP_GET_VARS["acces"];
$dn = $HTTP_GET_VARS["docname"];
$nom = $HTTP_GET_VARS["nom"];
$tipe = $HTTP_GET_VARS["tipe"];
$fechahora = $HTTP_GET_VARS["fechahora"];
$c = "";
$tamano = strlen($dn) +1;
$arquivo = "";
$archivok = "";

For ($i=$tamano;$i>=0;$i--)
{
    if($c == "/" || $c == "\\")
    {
        $i=-1;
    }
    else
    {
        $c = substr($dn,$i,1);
        $arquivo= $arquivo .$c;
    }
}
$tamano = strlen($arquivo) - 2;
For ($i=$tamano;$i>=0;$i--)
{
    $c = substr($arquivo,$i,1);
    $archivok= $archivok .$c;
}

$dn=$archivok;
$arquivo =
file_get_contents("http://www.elperrodepavlov.com/PFC/rutas/" .$dn);
$tipo=substr($arquivo,0,1);

echo ("<h3>Transmisión de trayectoria desde PDA</h3>");

if ($HTTP_GET_VARS["acces"]=="pfc2010") {

```

```

        if($tipe!="t" && $tipe!="tt")
        {
            $sql = "INSERT INTO pfc VALUES
('','$dn','$nom','$tipo','$fechahora)";
            mysql_query($sql);
            echo "<p style='text-align:justify'><b>Ruta cargada:
</b><br /><br />";
            echo $nom;
            echo "</p>";
        }
        else
        {
            if($tipe=="tt"){
                $tipo="a";
                $sql = "INSERT INTO pfc VALUES
('','$dn','$nom','$tipo','$fechahora)";
                mysql_query($sql);
                echo "<p style='text-align:justify'><b>Se ha iniciado
la transmisión de la ruta en tiempo real ";
                echo $nom;
                echo "</b></p>";
            }
            else
            {
                $narchivo="";
                $tn = strlen($nom);
                For($i=0;$i<$tn;$i++)
                {
                    $c = substr($nom,$i,1);
                    if($c==" "){ $narchivo.="_"; }else{
$narchivo.=$c; }
                }
                $narchivo .= ".txt";
                $archivo =
file_get_contents("http://www.elperrodepavlov.com/PFC/rutas/" . $dn);
                $ar =
"/home/elperrodep/domains/elperrodepavlov.com/public_html/PFC/rutas/"
.$narchivo;
                $fp = fopen($ar, "w");
                fputs($fp, $archivo);
                fclose($fp);

                $tipo="a";
                $sql = "UPDATE pfc SET archivo='$narchivo',
nombre='$nom', fecha='$fechahora', tipo='t' WHERE tipo='$tipo'";
                mysql_query($sql);
                echo "<p style='text-align:justify'><b>Ha finalizado
la transmisión de la ruta en tiempo real ";
                echo $nom;
                echo "</b></p>";
            }
        }
    }
}
else
{
    echo "<p style='text-align:justify'><b>Error: </b>La transmisión
del archivo a fallado.</p>";
}

mysql_close();

```

```
?>
</body>
</html>
```

10.2.2 [e_trans.php] Alerta sobre el estado de la transmisión en tiempo real

```
<body>
<?php
    $onthismoment=0;
    include("bdtfc.phtml");
    $link=Conectarse();

    $resultComen = mysql_query("SELECT * FROM pfc ORDER BY
fecha DESC",$link);
    while($rowComen = mysql_fetch_array($resultComen))
    {
        if($rowComen["tipo"]=="a")
        {
            $onthismoment=1;
        }
    }
    mysql_free_result($resultComen);
?>
<table width="285" border="1" bordercolor="#003366">
    <tr><td align="center"><span style="text-align:center; font:
normal 12px Tahoma, Verdana, Arial, Helvetica, Sans-Serif; text-
decoration:blink">
        <? if($onthismoment==1){ echo ("<a href='tiempo_real.php'
target='_blank'><span style='text-align:center; color: #CC0000;'><b>Se
están recibiendo datos</b></span></a>"); }else{ echo ("<span
style='text-align:center; color: #546F92;'><b>No se están recibiendo
datos</b></span>"); } ?>
        </span></td></tr></table>
</body>
```

10.2.3 Panel de últimas rutas cargadas

```
<?
    $i=0;
    include("pages/bdtfc.phtml");
    $link=Conectarse();

    $resultComen = mysql_query("SELECT * FROM pfc ORDER BY fecha
DESC",$link);
    while($rowComen = mysql_fetch_array($resultComen))
    {
        if($i < 3)
        {
            $name = utf8_encode($rowComen["nombre"]);
            $t = strlen($name);
            if($t > 30){ $name = substr($name,0,30) . "..."; }
            $fecha = substr($rowComen["fecha"],0,10);
            $fecha = preg_replace("/([0-9]{4})[\/|\-]([0-
9]{2})[\/|\-]([0-9]{2})/", "\$3-\$2-\$1", $fecha);
            $f = substr($fecha,0,6) . substr($fecha,8,2);
            $fecha = $f;
        }
    }
}
```

```

        $hora = substr($rowComen["fecha"],11,5);
        echo ("<table border='0'
        background='images/barra_ruta_cargada.jpg'
        width='220px' cellpadding='0' cellspacing='0'><tr><td
        valign='top' align='left' height='15px'
        width='120'><span class='panel'>&nbsp;&nbsp;&nbsp;<b>[ ");
        echo $fecha;
        echo (" ] &middot; ");
        echo $hora;
        echo ("</td><td valign='bottom'
        align='left'>&nbsp;&nbsp;<img src='images/hora.jpg'
        width='15px' height='15px'></td></tr><tr><td
        colspan='2' height='26' valign='top'>&nbsp;&nbsp;&nbsp;<a
        href='pages/visualizar_ruta.php?id=");
        echo $rowComen["id"];
        echo ("&tipo=r' style='color:#9AB7DA; text-
        decoration:none;'><span class='panel2'><b>");
        echo $name;
        echo ("</a></b></span></td></tr></table></span>");
    }
    $i=$i+1;
}mysql_free_result($resultComen);
?>

```

10.2.4 [ruta_simple.php] Visualización de una ruta simple

```

<script
src="http://maps.google.com/maps?file=api&v=2&key=ABQIAAAA5pMd
D-
0D3ORNPkKHszZ_2hRd8JQVcE9gzblpFMbRBds2PGSpBRQGHPyF0W1K9RFdrwRp0DFq4zV
LQ"
        type="text/javascript"></script>
<script type="text/javascript">

function initialize() {
    if (GBrowserIsCompatible()) {
        var map = new GMap2(document.getElementById("map_canvas"));
        map.addControl(new GSmallMapControl());
        map.addControl(new GMapTypeControl());
        map.setMapType(G_NORMAL_MAP);

<?php
$ruta = "http://www.elperrodepavlov.com/PFC/rutas/";
$ruta .= $_GET_VARS["ruta"];
$archivo = file_get_contents($ruta); //Guardamos archivo.txt en
$archivo
$tamano =strlen($archivo); //obtenemos la longitud de la cadena de
texto contenida en $archivo

$i=0;
$lat="";
$lon="";
$c="";
$b=0;
$a=0;
$tipo=substr($archivo,0,1);
$s=0;

if( $tipo == "r" || $tipo == "e" || $tipo == "t" || $tipo == "z" )

```

```
{
    $s=0;
}else{
    $s=1;
}

For ($i=2;$i<=$tamano;$i++)
{
    $c = substr($archivo,$i,1);

    if( $c == "," ){
        $a=$a+1;

        if( $b == 0 )
        {
            $b=1;
        }
        else
        {
            $b=0;
        }
    }

    if( $c != "," ){
        if( $b == 0 ){
            $lat = $lat .$c;
        }
        else
        {
            $lon = $lon .$c;
        }
    }
}

if($a==2)
{

    if($i < 30)
    {
        echo("map.setCenter(new GLatLng(");
        echo $lat;
        echo(",");
        echo $lon;
        echo("), 14);\n");
        echo("var polyline = new GPolyline([\n");
    }
    else
    {
        echo ("new GLatLng(");
        echo $lat;
        echo ("");
        echo $lon;

        if($i == ($tamano - 1))
        {
            echo (")\n");
        }
        else
        {
            echo ("),\n");
        }
    }
}
```

```

    }
    }
    $lat="";
    $lon="";
    $b=0;
    $a=0;

    if($s == 1)
    {
        $i++;
        $c = substr($archivo,$i,1);
        if( $c != "," )
        {
            $i++;
            $c = substr($archivo,$i,1);
            if( $c != "," )
            {
                $i++;
                $c = substr($archivo,$i,1);
                if( $c != "," )
                {
                    $i++;
                    $c = substr($archivo,$i,1);
                }
            }
        }
    }
}
?>

    ], "#0000ff", 2);
    map.addOverlay(polyline);
}

</script>

<body onload="initialize()" onunload="GUnload()">
    <div id="map_canvas" style="width: 600px; height: 390px"></div>
</body>

```

10.2.5 [ruta_velocidad.php] Visualización de una ruta con información de velocidad

```

<script
src="http://maps.google.com/maps?file=api&v=2.x&key=ABQIAAAA5p
MdD-
0D3ORNPkKHszZ_2hRd8JQVcE9gzblpFMbRBdS2PGSpBRQGHPPyf0WlK9RFdrwRp0DFq4zV
LQ" type="text/javascript"></script>
    <script type="text/javascript">
document.write('<' + 'script type="text/javascript"
src="distancia_a_ruta.js"><'+'/script>');
    </script>
    <script type="text/javascript">
var map;
<?
$ruta = "http://www.elperrodepavlov.com/PFC/rutas/";
$ruta .= $HTTP_GET_VARS["ruta"];

```

```
$archivo = file_get_contents($ruta); //Guardamos archivo.txt en
$archivo
$tamano =strlen($archivo); //obtenemos la longitud de la cadena de
texto contenida en $archivo

$a=0;
$b=0;
$lat="";
$lon="";

For ($i=2;$i<=$tamano;$i++)
{
    $c = substr($archivo,$i,1);

    if( $c == "," ){
        $a=$a+1;

        if( $b == 0 )
        {
            $b=1;
        }
        else
        {
            $b=0;
        }
    }

    if( $c != "," ){
        if( $b == 0 ){
            $lat = $lat .$c;
        }
        else
        {
            $lon = $lon .$c;
        }
    }

    if($a==2)
    {
        echo("var centerPoint = new GLatLng(");
        echo $lat;
        echo (",");
        echo $lon;
        echo (");");
        $i=$tamano+1;
    }
}
?>
var routeVertices = [];
var routePixels = [];
var routeOverlay = null;
var normalProj = G_NORMAL_MAP.getProjection();
var threshold = 15; //pixels for proximity
var snapToRoute;
function load() {
map = new GMap2(document.getElementById("map"),
{draggableCursor:"crosshair"});
map.setCenter(centerPoint, 13);
map.addControl(new GLargeMapControl());
map.addControl(new GMapTypeControl());
```



```

GEvent.addListener(map, 'mousemove', mouseMove); //detección del
evento mover el ratón
cargarruta();
}

function mouseMove(latlng) {
if(snapToRoute){
document.getElementById("infoPanel").innerHTML = ""; //mensaje
inferior
}
}
function addRouteVertex(vertex) {
routeVertices.push(vertex);
var pix = G_NORMAL_MAP.getProjection().fromLatLngToPixel(vertex,
map.getZoom());
routePixels.push(pix);
if (routeVertices.length > 1) {
plotRoute();
if (!snapToRoute) {
snapToRoute = new SnapToRoute(map, startMarker, routeOverlay);
} else {
snapToRoute.updateTargets(null, routeOverlay);
}
} else {
startMarker = createMarker(vertex, 'Start')
map.addOverlay(startMarker);
}
}
function plotRoute() {
if ( routeOverlay ) {
map.removeOverlay(routeOverlay);
}
routeOverlay = new GPolyline(routeVertices, '#FF9601', 1, 1);
map.addOverlay(routeOverlay);
}
function createMarker(latlng, title) {
var marker = new GMarker(latlng, {title:title});
return marker;
}
function resetRoute() {
routeVertices = [];
routePixels = [];
map.clearOverlays();
document.getElementById("infoPanel").innerHTML = '';
}

<?php

$i=0;
$lat="";
$lon="";
$vel="";
$c="";
$b=0;
$a=0;
$n=0;

For ($i=2;$i<=$tamano;$i++)
{
    $c = substr($archivo,$i,1);

```

```

        if( $c == "," ){
            $a=$a+1;
        }

        if($a==3)
        {
            $n = $n +1;
            $a=0;
        }
    }

    echo ("var v = new Array(");
    echo $n;
    echo (");\n");
    echo ("var total = ");
    echo $n;
    echo (";\n");
?>

```

```

function cargarruta() {
<?php

For ($i=2;$i<=$tamano;$i++)
{
    $c = substr($archivo,$i,1);

    if( $c == "," ){
        $a=$a+1;
    }

    if($a==3)
    {
        $n = $n +1;
        $a=0;
    }
}

$a=0;
$n=0;

For ($i=2;$i<=$tamano;$i++)
{

    $c = substr($archivo,$i,1);

    if( $c == "," ){
        $a=$a+1;

        if( $b == 0 )
        {

            $b=1;
        }
        else
        {
            if($b == 1)
            {
                $b=2;
            }
            else

```

```

        {
            $b=0;
        }
    }

    if( $c != "," ){
        if( $b == 0 ){
            $lat = $lat .$c;
        }
        else
        {
            if($b==1)
            {
                $lon = $lon .$c;
            }
            else
            {
                $vel = $vel .$c;
            }
        }
    }

    if($a==3)
    {
        echo ("v[" );
        echo $n;
        echo ("]= '");
        echo $lat;
        echo (",");
        echo $lon;
        echo (",");
        echo $vel;
        echo ("';\n");
        echo ("addRouteVertex(new google.maps.LatLng(");
        echo $lat;
        echo (",");
        echo $lon;
        echo (")); \n");

        $lat="";
        $lon="";
        $vel="";
        $b=0;
        $a=0;

        $n = $n +1;
    }
}

?>
}
</script>
</head>

<body onload="load()" onunload="GUnload()">
    <div id="infoPanel" style="border: 1px solid gray; border-
color:#003366; padding: 3px; width: 592px; height: 20px; font-family:
arial; font-weight: bold; font-size: 12px; color:#006699"></div>
    <div id="map" style="width: 600px; height: 370px;"></div>

```

```
</body>
```

10.2.6 [distancia_a_ruta.js] Distancia del puntero del ratón a la trayectoria

```
function SnapToRoute(map, marker, polyline) {
    this.routePixels_ = [];
    this.normalProj_ = G_NORMAL_MAP.getProjection();
    this.map_ = map;
    this.marker_ = marker;
    this.polyline_ = polyline;

    this.init_();
}

SnapToRoute.prototype.init_ = function () {
    this.loadLineData_();
    this.loadMapListener_();
};

SnapToRoute.prototype.updateTargets = function (marker, polyline) {
    this.marker_ = marker || this.marker_;
    this.polyline_ = polyline || this.polyline_;
    this.loadLineData_();
};

SnapToRoute.prototype.loadMapListener_ = function () {
    var me = this;
    GEvent.addListener(me.map_, 'mousemove',
        GEvent.callback(me, me.updateMarkerLocation_));
    GEvent.addListener(me.map_, 'zoomend',
        GEvent.callback(me, me.loadLineData_));
};

SnapToRoute.prototype.loadLineData_ = function () {
    var zoom = this.map_.getZoom();
    this.routePixels_ = [];
    for (var i = 0; i < this.polyline_.getVertexCount(); i++) {
        var Px =
this.normalProj_.fromLatLngToPixel(this.polyline_.getVertex(i), zoom);
        this.routePixels_.push(Px);
    }
};

SnapToRoute.prototype.updateMarkerLocation_ = function (mouseLatLng) {
    var markerLatLng = this.getClosestLatLng(mouseLatLng);
    this.marker_.setLatLng(markerLatLng);
    var velocity = velocidad_puntual(markerLatLng);
    document.getElementById("infoPanel").innerHTML = "<b>Posición
[Latitud, Longitud]:</b> " + markerLatLng + " <b>Velocidad:</b> " +
velocity + "Km/h";
};

SnapToRoute.prototype.getClosestLatLng = function (latlng) {
    var r = this.distanceToLines_(latlng);
    return this.normalProj_.fromPixelToLatLng(new GPoint(r.x, r.y),
this.map_.getZoom());
};
```

```

SnapToRoute.prototype.getDistAlongRoute = function (latlng) {
    if (typeof(opt_latlng) === 'undefined') {
        latlng = this.marker_.getLatLng();
    }

    var r = this.distanceToLines_(latlng);
    return this.getDistToLine_(r.i, r.to);
};

SnapToRoute.prototype.distanceToLines_ = function (mouseLatLng) {
    var zoom = this.map_.getZoom();
    var mousePx = this.normalProj_.fromLatLngToPixel(mouseLatLng, zoom);
    var routePixels_ = this.routePixels_;
    return this.getClosestPointOnLines_(mousePx, routePixels_);
};

SnapToRoute.prototype.getDistToLine_ = function (line, to) {
    var routeOverlay = this.polyline_;
    var d = 0;
    for (var n = 1; n < line; n++) {
        d += routeOverlay.getVertex(n -
1).distanceFrom(routeOverlay.getVertex(n));
    }
    d += routeOverlay.getVertex(line -
1).distanceFrom(routeOverlay.getVertex(line)) * to;

    return d;
};

SnapToRoute.prototype.getClosestPointOnLines_ = function (pXy, aXys) {
    var minDist;
    var to;
    var from;
    var x;
    var y;
    var i;
    var dist;

    if (aXys.length > 1) {
        for (var n = 1; n < aXys.length ; n++) {
            if (aXys[n].x !== aXys[n - 1].x) {
                var a = (aXys[n].y - aXys[n - 1].y) / (aXys[n].x - aXys[n -
1].x);
                var b = aXys[n].y - a * aXys[n].x;
                dist = Math.abs(a * pXy.x + b - pXy.y) / Math.sqrt(a * a + 1);
            } else {
                dist = Math.abs(pXy.x - aXys[n].x);
            }

            // length^2 of line segment
            var rl2 = Math.pow(aXys[n].y - aXys[n - 1].y, 2) +
Math.pow(aXys[n].x - aXys[n - 1].x, 2);

            // distance^2 of pt to end line segment
            var ln2 = Math.pow(aXys[n].y - pXy.y, 2) + Math.pow(aXys[n].x -
pXy.x, 2);

            // distance^2 of pt to begin line segment
            var lnm12 = Math.pow(aXys[n - 1].y - pXy.y, 2) + Math.pow(aXys[n
- 1].x - pXy.x, 2);

```

```

    // minimum distance^2 of pt to infinite line
    var dist2 = Math.pow(dist, 2);

    // calculated length^2 of line segment
    var calcrl2 = ln2 - dist2 + lnml2 - dist2;

    // redefine minimum distance to line segment (not infinite line)
    if necessary
        if (calcrl2 > rl2) {
            dist = Math.sqrt(Math.min(ln2, lnml2));
        }

        if ((minDist == null) || (minDist > dist)) {
            to = Math.sqrt(lnml2 - dist2) / Math.sqrt(rl2);
            from = Math.sqrt(ln2 - dist2) / Math.sqrt(rl2);
            minDist = dist;
            i = n;
        }
    }

    if (to > 1) {
        to = 1;
    }

    if (from > 1) {
        to = 0;
        from = 1;
    }

    var dx = aXys[i - 1].x - aXys[i].x;
    var dy = aXys[i - 1].y - aXys[i].y;

    x = aXys[i - 1].x - (dx * to);
    y = aXys[i - 1].y - (dy * to);

}

return {'x': x, 'y': y, 'i': i, 'to': to, 'from': from};
}

function velocidad_puntual(latlong) {

    var mlatlong="";
    mlatlong=String(latlong);
    var mlat="";
    var mlon="";
    var a=0;
    var b=0;

    for (i=0;i<mlatlong.length;i++)
    {
        if(a<1){

            if(mlatlong.charAt(i) != "(")
            {
                if(mlatlong.charAt(i) == ")")
                {
                    a++;
                    b++;
                }
            }
        }
    }
}

```

```

else
{
    if(mlatlong.charAt(i) == ",")
    {
        b++;
        i++;
    }
}

if(b==0)
{
    mlat =mlat + mlatlong.charAt(i);
}
if(b==1)
{
    mlon =mlon + mlatlong.charAt(i);
}
}
}

var a=0;
var lat="";
var lon="";
var vel=""
var difini=0;
var difact=0;
var vel_punto=0;

var i=0;
var e=0;
var t="";
var la1=0;
var la2=0;
var lo1=0;
var lo2=0;

for (i=0;i<total;i++)
{
    t = String(v[i]);

    for(e=0;e<t.length;e++)
    {
        if(a<3){
            if(t.charAt(e) == ",")
            {
                a++;
            }
        }
        else
        {
            if(a==0)
            {
                lat = lat + t.charAt(e);
            }
            if(a==1)
            {
                lon = lon + t.charAt(e);
            }
            if(a==2)
            {
                vel = vel + t.charAt(e);
            }
        }
    }
}

```



```

        }
    }
}
a=0;

la1=mlat;
lo1=m lon;
la2=lat;
lo2=lon;

if(i<=1)
{
    difini = Math.abs(la1-la2) + Math.abs(lo1-lo2);
    vel_punto=vel;
}
else
{
    difact = Math.abs(la1-la2) + Math.abs(lo1-lo2);

    if(difact<difini)
    {
        difini=difact;
        vel_punto=vel;
    }
}

e=0;
vel="";
lat="";
lon="";
}

return vel_punto;
}

```

10.2.7 [ruta_error.php] Visualización de ruta de comparación de sistemas o error

```

<script
src="http://maps.google.com/maps?file=api&v=2.91&key=ABQIAAAA5
pMdD-
0D3ORNPkKHszZ_2hRd8JQVce9gzblpFMbRBdS2PGSpBRQGHPPyf0W1K9RFdrwRp0DFq4zV
LQ"
    type="text/javascript"></script>
<script type="text/javascript">

function initialize() {
    if (GBrowserIsCompatible()) {
        var map = new GMap2(document.getElementById("map_canvas"),
{Cursor: 'poniter'});
        map.addControl(new GSmallMapControl());
        map.addControl(new GMapTypeControl());
        map.setMapType(G_NORMAL_MAP);

<?php
$ruta = "http://www.elperrodepavlov.com/PFC/rutas/";
$ruta .= $_GET_VARS["ruta"];

```

```
$archivo = file_get_contents($ruta); //Guardamos archivo.txt en
$archivo
$tamano =strlen($archivo); //obtenemos la longitud de la cadena de
texto contenida en $archivo

$i=0;
$lat="";
$lon="";
$c="";
$b=0;
$a=0;
$tipo=substr($archivo,0,1);
$s=0;

if( $tipo == "r" || $tipo == "e" || $tipo == "t" || $tipo == "z" )
{
    $s=0;
}else{
    $s=1;
}

For ($i=2;$i<=$tamano;$i++)
{
    $c = substr($archivo,$i,1);

    if( $c == "," ){
        $a=$a+1;

        if( $b == 0 )
        {
            $b=1;
        }
        else
        {
            $b=0;
        }
    }

    if( $c != "," ){
        if( $b == 0 ){
            $lat = $lat .$c;
        }
        else
        {
            $lon = $lon .$c;
        }
    }
}

if($a==2)
{
    if($i < 30)
    {
        echo("map.setCenter(new GLatLng(");
        echo $lat;
        echo(",");
        echo $lon;
        echo("), 14);\n");
        echo("var polyline = new GPolyline([\n");
    }
}
```

```

else
{

echo ("new GLatLng(");
echo $lat;
echo (",");
echo $lon;

if($i == ($tamano - 1))
{
    echo (")\n");
}
else
{
    echo (",\n");
}
}
$lat="";
$lon="";
$b=0;
$a=0;

if($s == 1)
{
    $i++;
    $c = substr($archivo,$i,1);
    if( $c != "," )
    {
        $i++;
        $c = substr($archivo,$i,1);
        if( $c != "," )
        {
            $i++;
            $c = substr($archivo,$i,1);
            if( $c != "," )
            {
                $i++;
                $c = substr($archivo,$i,1);
            }
        }
    }
}

}
}
?>

], "#0000ff", 2);
map.addOverlay(polyline);

polyline.clickable=true;

var d1 =polyline.getLength();
d1=Math.round(d1*10) / 10;
document.dist.km1.value = d1 / 1000;
document.dist.km1.value = document.dist.km1.value + " Km";

<?php
$ruta = "http://www.elperrodepavlov.com/PFC/rutas/";
$rutagpslong = strlen($HTTP_GET_VARS["ruta"]);
$rutagps.= $ruta;
For ($i=0;$i<$rutagpslong;$i++)
{

```

```
$c = substr($HTTP_GET_VARS["ruta"],$i,1);
if($c==".")
{
    $rutagps.="_gps";
}
$rutagps=$c;
}
$archivo = file_get_contents($rutagps); //Guardamos archivo.txt en
$archivo
$tamano =strlen($archivo); //obtenemos la longitud de la cadena de
texto contenida en $archivo

$i=0;
$lat="";
$lon="";
$c="";
$b=0;
$a=0;
$tipo=substr($archivo,0,1);
$s=0;

if( $tipo == "r" || $tipo == "e" || $tipo == "t" || $tipo == "z" )
{
    $s=0;
}else{
    $s=1;
}

For ($i=2;$i<=$tamano;$i++)
{
    $c = substr($archivo,$i,1);

    if( $c == "," ){
        $a=$a+1;

        if( $b == 0 )
        {
            $b=1;
        }
        else
        {
            $b=0;
        }

    }

    if( $c != "," ){
        if( $b == 0 ){
            $lat = $lat .$c;
        }
        else
        {
            $lon = $lon .$c;
        }
    }

    if($a==2)
    {

        if($i < 30)
        {
```

```

        echo("map.setCenter(new GLatLng(");
        echo $lat;
        echo(",");
        echo $lon;
        echo("), 14);\n");
        echo("var polyline = new GPolyline([\n");
    }
    else
    {

        echo ("new GLatLng(");
        echo $lat;
        echo ("");
        echo $lon;

        if($i == ($tamano - 1))
        {
            echo (")\n");
        }
        else
        {
            echo ("),\n");
        }
    }
    $lat="";
    $lon="";
    $b=0;
    $a=0;

    if($s == 1)
    {
        $i++;
        $c = substr($archivo,$i,1);
        if( $c != "," )
        {
            $i++;
            $c = substr($archivo,$i,1);
            if( $c != "," )
            {
                $i++;
                $c = substr($archivo,$i,1);
                if( $c != "," )
                {
                    $i++;
                    $c = substr($archivo,$i,1);
                }
            }
        }
    }
}
?>

], "#ff0000", 2);
map.addOverlay(polyline);

polyline.clickable=true;

GEvent.addListener(map, "click", function(overlay,latlng) {
if (latlng) {
    var myHtml = "<b>Coordenadas:</b> " + latlng;
    map.openInfoWindow(latlng, myHtml);
}
}

```



```
function initialize() {
    if (GBrowserIsCompatible()) {
        map = new GMap2(document.getElementById("map_canvas"));
        map.addControl(new GSmallMapControl());
        map.addControl(new GMapTypeControl());
        map.setMapType(G_NORMAL_MAP);
    }
}

<?php
$ruta = "http://www.elperrodepavlov.com/PFC/rutas/";
$ruta .= $_GET_VARS["ruta"];

if (file_exists($ruta)){
    $fclosed=1;
}else{
    $fclosed=0;
    echo("refrescar()");
}

if($onthismoment==1 && $fclosed==1){
    $archivo = file_get_contents($ruta); //Guardamos archivo.txt en
    $archivo
    $tamano =strlen($archivo); //obtenemos la longitud de la cadena de
    texto contenida en $archivo

    $i=0;
    $lat="";
    $clon="";
    $clat="";
    $lon="";
    $c="";
    $b=0;
    $a=0;

    For ($i=2;$i<=$tamano;$i++)
    {
        $c = substr($archivo,$i,1);

        if( $c == "," ){
            $a=$a+1;

            if( $b == 0 )
            {
                $b=1;
            }
            else
            {
                $b=0;
            }
        }

        if( $c != "," ){
            if( $b == 0 ){
                $lat = $lat . $c;
            }
            else
            {
                $lon = $lon . $c;
            }
        }
    }
}
```



```

        if ($a==2)
        {
            if ($i < 30)
            {
                echo ("map.setCenter(new GLatLng(");
                echo $lat;
                echo (","");
                echo $lon;
                echo ("), 14);\n");
                $clat=$lat;
                $clon=$lon;
                echo ("var polyline = new GPolyline([\n");
            }
            else
            {
                echo ("new GLatLng(");
                echo $lat;
                echo (","");
                echo $lon;

                if ($i == ($tamano - 1))
                {
                    echo (")\n");
                }
                else
                {
                    echo ("),\n");
                }
            }
            $clat=$lat;
            $clon=$lon;
            $lat="";
            $lon="";
            $b=0;
            $a=0;
        }
    }
}
?>

        ], "#0000ff", 2);
        map.addOverlay(polyline);

        point = new GLatLng(<?php echo $clat; echo (",""); echo $clon;
echo ("");"> ?>
        marker = new GMarker(point);
        map.addOverlay(marker);
        marker.openInfoWindowHtml('<b>Coordenadas: </b> ' + point);
    }
}

var r=0;
var v=0;

function estado() {
    if (r==0)
    {
        r=1;
        document.panel.stop.value="Reiniciar visualización";
    }
}

```

```

        else
        {
            r=0;
            document.panel.stop.value="Detener visualización";
        }
    }

    function refrescar() {
    <? if($onthismoment==0){ echo ("r=1;"); } ?>
        if(r==0)
        {
            v=v+1;
            if(v==5)
            {
                document.location.reload();
                v=0;
            }
        }
    }
    setInterval("refrescar()",1000);
</script>
<style type="text/css">
<!--
.Estilo1 {
    font-family: Arial, Helvetica, sans-serif;
    font-weight: bold;
}
-->
</style>
</head>

    <body onload="initialize()" onunload="GUnload()">
        <div id="map_canvas" style="width: 610px; height: 370px;
display:<? if($HTTP_GET_VARS["ruta"]=="" || $onthismoment==0 ){echo
("none");}else{echo ("");} ?>;"></div>
        <div style="width: 610px; height: 50px">
            <form name="panel">
                <?
                    $b=0;
                    $resultComen = mysql_query("SELECT * FROM pfc ORDER BY
fecha DESC",$link);
                    while($rowComen = mysql_fetch_array($resultComen))
                    {
                        if( $rowComen["tipo"]=="a" )
                        {
                            $archivo = $rowComen["archivo"];
                            $hora = substr($rowComen["fecha"],11,8);
                            $fecha = substr($rowComen["fecha"],0,10);
                            $fecha = preg_replace("/([0-9]{4})[\/|\\-]([0-
9]{2})[\/|\\-]([0-9]{2})/","\\$3-\\$2-\\$1",$fecha);
                            $fecha_act = date("y-m-d h:m:s");
                            echo("<p style='text-align:justify;font: normal
11px Tahoma, Verdana, Arial, Helvetica, Sans-Serif; line-height:
1.6em; color: #333;'><b>Último punto recibido: </b>");
                            echo $fecha;
                            echo (" a las ");
                            echo $hora;
                            $b=1;
                            $tipo="a";
                            break;

```

[illegible]

10.2.9 [alta_ruta.php / alta.php] Gestión de rutas - Altas

```
<?php

$use = $_SESSION['aduser'];
$pas = $_SESSION['adpas'];

if(($use=="") || ($pas=="")){
    printf("<p style='text-align:center'><b>:: Acceso
restringido ::<b><br /><br /><br /><img
src='../images/restringido.jpg'><br/><b><a
href='acceso_gestion.php'>Acceder con mi perfil de
usuario</a></b></p><br />");
```

```

    }else{

        echo ("
        <h3>Cargar nueva ruta</h3>
        <br />
        <form name='alta' action='alta.php' method='post'
        enctype='multipart/form-data'>
            <table border='0' width='100%' height='210'>
                <tr>
                    <td width='47%' rowspan='4'
                    valign='top'><p><strong>Tipo de ruta:</strong><br />
                        <select name='tipo_ruta' onchange='cambio_tipo()'>
                            <option>Ruta simple</option>
                            <option>Ruta con velocidad</option>
                            <option>Ruta comparativa GPS tradicional</option>
                            <option>Ruta con velocidades y
                            comparativa GPS</option>
                        </select>
                    </p>
                    <p><strong>Nombre de la ruta:</strong><br />
                        <b>
                            <input name='nombre' type='text' size='30' />
                        </b></p>
                    <p><strong>Archivo de ruta:</strong><br />
                        <input name='ruta' type='file' size='16'
                        accept='text/plain' />
                    </p></td>
                    <td width='53%'> </td>
                </tr>
                <tr>
                    <td height='17'></td>
                </tr>
                <tr>
                    <td height='25'></td>
                </tr>
                <tr>
                    <td height='116' valign='top'><p>&nbsp;</p>
                    <p><strong><input type='text' name='label'
                    value='Archivo de ruta (GPS tradicional):' size='28'
                    style='border:none; font: normal 11px Tahoma, Verdana, Arial,
                    Helvetica, Sans-Serif; font-weight:bold; color: #333;
                    visibility:hidden' readonly=''/></strong><br />
                        <input name='ruta2' type='file' size='16'
                        accept='text/plain' style='visibility:hidden' />
                    </p></td>
                    <td> </td>
                </tr>
                <tr>
                    <td height='40'><input type='submit' value='Cargar
                    ruta'> </td>
                    <td> </td>
                </tr>
            </table>
        </form>");

        echo ("<br /><table border='0' width='597px'><tr><td
        width='500px'>&nbsp;</td><td width='17px'><img
        src='../images/user.jpg' /></td><td valign='center' width='80px'><b>
        ");
        echo ($use);
    }
}

```

```

        echo (" | <a href='desconexion.php' style='text-
decoration:none'>Salir</a> </b></td><tr></table>");

    }
?>

```

10.2.9.1 alta.php

```

<?php

    $use = $_SESSION['aduser'];
    $pas = $_SESSION['adpas'];

    if(($use=="") || ($pas=="")){
        printf("<p style='text-align:center'><b>:: Acceso
restringido ::<b><br /><br /><br /><img
src='../images/restringido.jpg'><br/><b><a
href='acceso_gestion.php'>Acceder con mi perfil de
usuario</a></b></p><br />");
    }else{

        $nombre = $_REQUEST["nombre"];
        $tipo_ruta = $_REQUEST["tipo_ruta"];
        $archivo = $_FILES['ruta']['name'];
        $archivo2 = $_FILES['ruta2']['name'];
        $remoto2 = "";

        $local1 = $_FILES['ruta']['tmp_name'];
        $remoto1 = $_FILES['ruta']['name'];
        $local2 = $_FILES['ruta2']['tmp_name'];
        $id_ftp = ftp_connect("elperrodepavlov.com",21);

        if($archivo2!=""){
            $rutagpslong = strlen($archivo);
            $rutagps="";
            For ($i=0;$i<$rutagpslong;$i++)
            {
                $c = substr($archivo,$i,1);
                if($c==".")
                {
                    $rutagps.="_gps";
                }
                $rutagps.=$c;
            }
            $remoto2=$rutagps;
        }

        if($tipo_ruta=="Ruta simple"){ $tipo_ruta="r"; }
        if($tipo_ruta=="Ruta con velocidad"){ $tipo_ruta="v"; }
        if($tipo_ruta=="Ruta comparativa GPS tradicional"){
$tipo_ruta="e"; }
        if($tipo_ruta=="Ruta con velocidades y comparativa GPS"){
$tipo_ruta="w"; }

        $s=0;
        $fich = file_get_contents($local1);

```

```

    $tipe=substr($fich,0,1);
    if( $tipe!=$tipo_ruta ){ $s=1;}

    $x=0;
    $y=0;
    $z=0;
    $b=0;

    if( $tipo_ruta=="e" || $tipo_ruta=="w"){ if($remoto2==""){
$z=1; } }

    $resultComen = mysql_query("SELECT * FROM pfc ORDER BY
fecha DESC",$link);
    while($rowComen = mysql_fetch_array($resultComen))
    {
        $n = $rowComen["nombre"];
        $a = $rowComen["archivo"];

        if($nombre==$n){ $x=1; }

        if($archivo==$a){ $y=1; }

    }mysql_free_result($resultComen);

    if( $x==1 || $y==1 || $z==1 || $s==1)
    {
        if($x==1){ echo ("<p style='text-align:center'>No se
ha dado de alta la ruta indicada. Ya existe una ruta con el nombre
indicado.<br/><a href='alta_ruta.php'>Volver</a></p>"); }
        if($y==1 && $x!=1){ echo ("<p style='text-
align:center'>No se ha dado de alta la ruta indicada. Ya existe el
archivo de ruta indicado.<br/><a
href='alta_ruta.php'>Volver</a></p>"); }
        if($z==1 && $x!=1 && $y!=1){ echo ("<p style='text-
align:center'>No se ha dado de alta la ruta indicada. Debe indicar
también el archivo de la ruta tomada del sitema de GPS
tradicional.<br/><a href='alta_ruta.php'>Volver</a></p>"); }
        if($s==1 && $z!=1 && $x!=1 && $y!=1){ echo ("<p
style='text-align:center'>No se ha dado de alta la ruta indicada. El
tipo de archivo cargado no coincide con el indicado.<br/><a
href='alta_ruta.php'>Volver</a></p>"); }
    }
    else
    {
        if( $nombre!="" && $tipo_ruta!="" && $archivo!="" &&
$remoto1!="" ){
            ftp_login ($id_ftp, "elperrodep", "campana150406");
            ftp_pasv ($id_ftp, false);
            ftp_chdir ($id_ftp,
"domains/elperrodepavlov.com/public_html/PFC/rutas/");

            if(ftp_put($id_ftp,$remoto1,$local1,FTP_BINARY))
            {
                $b=1;
            }else{
                $b=0;
            }

            if($remoto2!=""){

if(ftp_put($id_ftp,$remoto2,$local2,FTP_BINARY))
            {

```

```

        $b=1;
    }else{
        $b=0;
    }
    } ftp_quit($id_ftp);

    if($b==1)
    {
        $fecha= date("y-m-d h:m:s");
        $sql = "INSERT INTO pfc VALUES
('','$archivo','$nombre','$tipo_ruta','$fecha')";
        mysql_query($sql);
        mysql_close();
        echo ("<p style='text-align:center'>La ruta se
ha cargado correctamente.<br/><a href='gestion.php'>Volver</a></p>");
    }
    else
    {
        echo ("<p style='text-align:center'>Ha ocurrido
un error al cargar el fichero de ruta y no se ha dado de alta la ruta
indicada.<br/><a href='alta_ruta.php'>Volver</a></p>");
    }
}
else
{
    echo ("<p style='text-align:center'>DEBE RELLENAR
TODOS LOS CAMPOS DEL FORMULARIO.<br/><a
href='alta_ruta.php'>Volver</a></p>");
}
}

echo ("<br /><table border='0' width='597px'><tr><td
width='500px'>&nbsp;</td><td width='17px'><img
src='../images/user.jpg' /></td><td valign='center' width='80px'><b>
");
echo ($use);
echo (" | <a href='desconexion.php' style='text-
decoration:none'>Salir</a> </b></td><tr></table>");
}
?>

```

10.2.10 [eliminar_ruta.php /eliminar.php] Gestión de rutas - Bajas

```

<?php

    $use = $_SESSION['aduser'];
    $pas = $_SESSION['adpas'];

    if(($use=="") || ($pas=="")){
        printf("<p style='text-align:center'><b>:: Acceso
restringido ::<b><br /><br /><br /><img
src='../images/restringido.jpg'><br/><b><a
href='acceso_gestion.php'>Acceder con mi perfil de
usuario</a></b></p><br />");
    }else{

```



```

echo("
<h3>Eliminar ruta</h3>
<br />");

    $ruta = $HTTP_GET_VARS["id"];

    $resultComen = mysql_query("SELECT * FROM pfc WHERE id LIKE
'%$ruta%' ORDER BY fecha",$link);
    while($rowComen = mysql_fetch_array($resultComen))
    {
        $fecha = substr($rowComen["fecha"],0,10);
        $hora = substr($rowComen["fecha"],11,8);
        $fecha = preg_replace("/([0-9]{4})[\/|\-]([0-
9]{2})[\/|\-]([0-9]{2})/", "\$3-\$2-\$1", $fecha);
        $nombre = $rowComen["nombre"];
        $archivo = $rowComen["archivo"];
        $tipo = $rowComen["tipo"];

    }mysql_free_result($resultComen);

    $archivo2="";
    echo("<b>");
    echo $nombre;
    echo("</b><br/><b>Archivo de ruta: </b>");
    echo $archivo;
    echo("<br/><b>Fecha de su última actualización: </b>");
    echo $fecha;
    echo(" a las ");
    echo $hora;

    if( $tipo_ruta=="e" || $tipo_ruta=="w" || $tipo_ruta=="x"
|| $tipo_ruta=="z")
    {
        $rutagpslong = strlen($archivo);
        $rutagps="";
        For ($i=0;$i<$rutagpslong;$i++)
        {
            $c = substr($archivo,$i,1);
            if($c==".")
            {
                $rutagps.="_gps";
            }
            $rutagps.=$c;
        }
        $archivo2=$rutagps;
    }

    echo ("
<br/><br/>
<form action='eliminar.php' enctype='multipart/form-data'>
        <input type='submit' name='eliminar'
class='boton' value='Eliminar ruta'>
        <input type='button' name='cancelar'
class='boton' value='Cancelar'
onClick=\"javascript:window.location=('gestion.php');\">
        <input name='id' type='hidden' value=''>; echo
$ruta; echo("'" />
        <input name='archivo' type='hidden' value=''>;
echo $archivo; echo("'" />

```

```

        <input name='archivo2' type='hidden' value='');
echo $archivo2; echo ("' />
        </form>");
        echo ("<br /><table border='0' width='597px'><tr><td
width='500px'>&nbsp;</td><td width='17px'><img
src='../images/user.jpg' /></td><td valign='center' width='80px'><b>
");
        echo ($use);
        echo (" | <a href='desconexion.php' style='text-
decoration:none'>Salir</a> </b></td><tr></table>");
    }
    ?>

```

10.2.10.1 eliminar.php

```

<?php

    $use = $_SESSION['aduser'];
    $pas = $_SESSION['adpas'];

    if(($use=="") || ($pas=="")){
        printf("<p style='text-align:center'><b>:: Acceso
restringido ::<b><br /><br /><br /><img
src='../images/restringido.jpg'><br/><b><a
href='acceso_gestion.php'>Acceder con mi perfil de
usuario</a></b></p><br />");
    }else{

        echo ("<h3>Eliminar ruta</h3>
<br />");

        $ruta = $_HTTP_GET_VARS["id"];
        $path = $_SERVER['DOCUMENT_ROOT'].'/PFC/rutas/';
        $path .= $_HTTP_GET_VARS["archivo"];
        $path2 = "";
        if($_HTTP_GET_VARS["archivo2"]!="")
        {
            $path2 = $_SERVER['DOCUMENT_ROOT'].'/PFC/rutas/';
            $path2 .= $_HTTP_GET_VARS["archivo2"];
        }
        $b=0;

        $result=mysql_query("select * from pfc",$link);

        while($row = mysql_fetch_array($result)) {
            if ($ruta == $row["id"]) {
                mysql_query("delete from pfc where
id='$ruta'", $link);
                echo "<p style='text-align:center'><b>La ruta
ha sido eliminada correctamente</b></p>";
                echo "<p style='text-align:center'><a
href='gestion.php'><b>Volver</b></a></p>";
                unlink($path);
                if($path2!=""){ unlink($path2); }
                $b=1;
            }
        }
    }

```

```

    }
    if ($b == 0){
        echo "<p style='text-align:center'>Ha ocurrido un
error y no se ha podido llevar a cabo su petición</b></p>";
        echo "<p style='text-align:center'><a
href='gestion.php'><b>Volver</b></a></p>";
    }else{
        $b=0;
    }
    echo ("<table border='0' width='597px'><tr><td
width='500px'>&nbsp;</td><td width='17px'><img
src='../images/user.jpg' /></td><td valign='center' width='80px'><b>
");
        echo ($use);
        echo (" | <a href='desconexion.php' style='text-
decoration:none'>Salir</a> </b></td><tr></table>");
    }
?>

```

10.2.11 [modificar_ruta.php/modificar.php] Gestión de rutas - Modificaciones

```

<?php

    $use = $_SESSION['aduser'];
    $pas = $_SESSION['adpas'];

    if(($use=="") || ($pas=="")){
        printf("<p style='text-align:center'><b>:: Acceso
restringido ::<b><br /><br /><br /><img
src='../images/restringido.jpg'><br /><b><a
href='acceso_gestion.php'>Acceder con mi perfil de
usuario</a></b></p><br />");
    }else{

        echo("
<h3>Modificación de ruta</h3>
<br />
");
        ?>
        <?

        $ruta = $_HTTP_GET_VARS["id"];

        $resultComen = mysql_query("SELECT * FROM pfc WHERE id LIKE
'%$ruta%' ORDER BY fecha",$link);
        while($rowComen = mysql_fetch_array($resultComen))
        {
            $fecha = substr($rowComen["fecha"],0,10);
            $hora = substr($rowComen["fecha"],11,8);
            $fecha = preg_replace("/([0-9]{4})[\/|\-]([0-
9]{2})[\/|\-]([0-9]{2})/", "\$3-\$2-\$1", $fecha);
            $nombre = $rowComen["nombre"];
            $archivo = $rowComen["archivo"];
            $tipo = $rowComen["tipo"];

        }mysql_free_result($resultComen);
    }

```

```

if( $tipo!="r" && $tipo!="v" && $tipo!="t" && $tipo!="y" )
{
    $rutagpslong = strlen($archivo);
    $rutagps="";
    For ($i=0;$i<$rutagpslong;$i++)
    {
        $c = substr($archivo,$i,1);
        if($c==".")
        {
            $rutagps.="_gps";
        }
        $rutagps.=$c;
    }
}

?>
<?
echo ( "
<form name='modificacion' action='modificar.php'
method='post' enctype='multipart/form-data'>
<table border='0' width='100%' height='210'>
<tr>
    <td width='47%' rowspan='4'
valign='top'><p><strong>Tipo de ruta:</strong><br />
        <select name='tipo_ruta' onchange='cambio_tipo()'>
            <option id='1'><? if($tipo=="r"){ echo
("selected='selected'"); } echo (">Ruta simple</option>
            <option id='2'><? if( $tipo=="v" || $tipo=="y" ){
echo ("selected='selected'"); } echo (">Ruta con velocidades</option>
            <option id='3'><? if( $tipo=="e" || $tipo=="z" ){
echo ("selected='selected'"); } echo (">Ruta comparativa GPS
tradicional</option>
            <option id='4'><? if( $tipo=="w" ||
$tipo=="x" ){ echo ("selected='selected'"); } echo (">Ruta con
velocidades y comparativa GPS</option>
        </select>
    </p>
        <p><strong>Nombre de la ruta:</strong><br />
        <b>
        <input name='nombre_ruta' type='text' value='';
echo $nombre; echo ("' size='30' />
        </b></p>
        <p><strong>Archivo de ruta:</strong><br />
        <input name='archivo_ruta' type='file' size='16'
accept='text/plain'/>"); echo $archivo;
        echo ( "
    </p></td>
    <td width='53%'> </td>
</tr>
<tr>
    <td height='17'></td>
</tr>
<tr>
    <td height='25'></td>
</tr>
<tr>
    <td height='116' valign='top'>
        <p><strong><input type='text' name='label'
value='Archivo de ruta (GPS tradicional):' size='28'
style='border:none; font: normal 11px Tahoma, Verdana, Arial,

```

```

Helvetica, Sans-Serif; font-weight:bold; color: #333;"); if(
$tipo=="w" || $tipo=="x" || $tipo=="e" || $tipo=="z" ){ echo
("visibility:visible"); }else{ echo ("visibility:hidden"); } echo ("'
readonly=''/></strong><br />
        <input name='ruta2' type='file' size='16'
accept='text/plain'"); if( $tipo=="w" || $tipo=="x" || $tipo=="e" ||
$tipo=="z" ){ echo ("style='visibility:visible'"); }else{ echo
("style='visibility:hidden'"); } echo (" />"); if( $tipo=="w" ||
$tipo=="x" || $tipo=="e" || $tipo=="z" ){ echo $rutagps; }
        echo ("
        </p></div><input name='a1' type='hidden' value=''; echo
$archivo; echo ("' /><input name='a2' type='hidden' value=''); echo
$rutagps; echo ("' /><input name='id' type='hidden' value=''); echo
$ruta; echo ("' /></td>
    </tr>
    <tr>
        <td height='40'><input type='submit' value='Modificar
ruta'> </td>
        <td> </td>
    </tr>
</table>
</form>");

        echo ("<br /><table border='0' width='597px'><tr><td
width='500px'>&nbsp;</td><td width='17px'><img
src='../images/user.jpg' /></td><td valign='center' width='80px'><b>
");
        echo ($use);
        echo (" | <a href='desconexion.php' style='text-
decoration:none'>Salir</a> </b></td><tr></table>");
    }
    ?>

```

10.2.11.1 modificar.php

```

<?php

    $use = $_SESSION['aduser'];
    $pas = $_SESSION['adpas'];

    if(($use=="") || ($pas=="")){
        printf("<p style='text-align:center'><b>:: Acceso
restringido ::<b><br /><br /><br /><img
src='../images/restringido.jpg'><br/><b><a
href='acceso_gestion.php'>Acceder con mi perfil de
usuario</a></b></p><br />");
    }else{

        echo ("
        <h3>Modificación de ruta</h3>
        <br />");

        $tipo_ruta = $_REQUEST['tipo_ruta'];
        $nombre_ruta = $_REQUEST['nombre_ruta'];
        $a1 = $_REQUEST['a1'];
        $a2 = $_REQUEST['a2'];
    }
}

```

```

$cid = $_REQUEST['id'];
$fecha=date("y-m-d h:m:s");
$b=0;
$n=0;
$local1 = $_FILES['archivo_ruta']['tmp_name'];
$local2 = $_FILES['ruta2']['tmp_name'];
$archivo_ruta = $_FILES['archivo_ruta']['name'];
$remoto1 = $_FILES['archivo_ruta']['name'];
$archivo_ruta2 = $_FILES['ruta2']['name'];
$remoto2 = $_FILES['ruta2']['name'];
$cid_ftp = ftp_connect("elperrodepavlov.com",21);

if($archivo_ruta==""){ $archivo_ruta=$a1; }
if($archivo_ruta2==""){ $archivo_ruta2=$a2; }

if($archivo_ruta2!=""){
    $rutagpslong = strlen($archivo_ruta);
    $rutagps="";
    For ($i=0;$i<$rutagpslong;$i++)
    {
        $c = substr($archivo_ruta,$i,1);
        if($c==".")
        {
            $rutagps.="_gps";
        }
        $rutagps.=$c;
    }
    $archivo_ruta2=$rutagps;
    $remoto2=$rutagps;
}

if($tipo_ruta=="Ruta simple"){ $tipo_ruta="r"; }
if($tipo_ruta=="Ruta con velocidades"){ $tipo_ruta="v"; }
if($tipo_ruta=="Ruta comparativa GPS tradicional"){
$tipo_ruta="e"; }
if($tipo_ruta=="Ruta con velocidades y comparativa GPS"){
$tipo_ruta="w"; }

$s=0;
if($local1!="")
{
    $fich = file_get_contents($local1);
    $tipe=substr($fich,0,1);
    if( $tipe!=$tipo_ruta && $local1!="" ){ $s=1;}
}else{
    $tipe=$tipo_ruta;
}

$resultComen = mysql_query("SELECT * FROM pfc ORDER BY
fecha DESC",$link);
while($rowComen = mysql_fetch_array($resultComen))
{
    $nombre = $rowComen["nombre"];
    $archivo = $rowComen["archivo"];

    if($archivo_ruta==$archivo)
    {
        if($cid!=$rowComen["id"])
        {
            $n=1;
        }
    }
}

```

```

    }
    if($nombre_ruta==$nombre)
    {
        if($id!=$rowComen["id"])
        {
            $n=1;
        }
    }
    if( $id==$rowComen["id"] &&
$archivo_ruta!=$rowComen["archivo"])
    {
        $path =
$_SERVER['DOCUMENT_ROOT'].'/PFC/rutas/';
        $path .= $rowComen["archivo"];
        unlink($path);

        if( $tipo_ruta!="r" && $tipo_ruta!="v" &&
$tipo_ruta!="t" && $tipo_ruta!="y" )
        {
            $path =
$_SERVER['DOCUMENT_ROOT'].'/PFC/rutas/';
            $rutagpslong =
strlen($rowComen["archivo"]);
            $rutagps="";
            For ($i=0;$i<$rutagpslong;$i++)
            {
                $c =
substr($rowComen["archivo"],$i,1);

                if($c==".")
                {
                    $rutagps.="_gps";
                }
                $rutagps.=$c;
            }
            $path .= $rutagps;
            unlink($path);
        }
    }

}mysql_free_result($resultComen);

if($n==0 && $s==0){
    $sql = "SELECT * FROM pfc WHERE id = $id";
    $result = mysql_query($sql);

    if( $tipo_ruta=="r" || $tipo_ruta=="v" )
    {
        if($nombre_ruta!="" && $archivo_ruta!="")
        {
            $sql = "UPDATE pfc SET archivo='$archivo_ruta',
nombre='$nombre_ruta', tipo='$tipo_ruta', fecha='$fecha' WHERE
id='$id'";

            $result = mysql_query($sql);
            ftp_login ($id_ftp, "elperrodep",
"campana150406");

            ftp_pasv ($id_ftp, false);
            ftp_chdir ($id_ftp,
"domains/elperrodepavlov.com/public_html/PFC/rutas/");
            if($remoto1=="")
            {
                $b=1;
            }
        }
    }
}

```

```

        }
        else
        {
            if
(ftp_put($id_ftp,$remoto1,$local1,FTP_BINARY))
            {
                $b=1;
            }else{
                $b=0;
            }
        }
    }
    else
    {
        echo ("<p style='text-align:center'>DEBE
RELLENAR TODOS LOS CAMPOS DEL FORMULARIO</p><br /><a
href='modificar_ruta.php?ruta='");
        echo $id;
        echo ("'><b>Volver</b></a>");
    }ftp_quit($id_ftp);
}else{
    if($nombre_ruta!="" && $archivo_ruta2!="" )
    {
        $sql = "UPDATE pfc SET archivo='$archivo_ruta',
nombre='$nombre_ruta', tipo='$tipo_ruta', fecha='$fecha' WHERE
id='$id'";

        $result = mysql_query($sql);
        ftp_login ($id_ftp, "elperrodep",
"campana150406");

        ftp_pasv ($id_ftp, false);
        ftp_chdir ($id_ftp,
"domains/elperrodepavlov.com/public_html/PFC/rutas/");
        if($remoto1=="" || $local1=="")
        {
            $b=1;
        }
        else
        {
            if
(ftp_put($id_ftp,$remoto1,$local1,FTP_BINARY))
            {
                $b=1;
            }else{
                $b=0;
            }
        }
        if($remoto2=="" || $local2=="")
        {
            $b=1;
        }
        else
        {
            if
(ftp_put($id_ftp,$remoto2,$local2,FTP_BINARY))
            {
                $b=1;
            }else{
                $b=0;
            } ftp_quit($id_ftp);
        }
    }
}

```



```

        else
        {
            echo ("<p style='text-align:center'>DEBE
RELLENAR TODOS LOS CAMPOS DEL FORMULARIO</p><br /> <a
href='modificar_ruta.php?ruta='");
            echo $id;
            echo ("'><b>Volver</b></a>");
        }

    }

    if($b==1)
    {
        echo "<p style='text-align:center'>La ruta se ha
modificado correctamente.</p>";
    }
    else
    {
        echo ("<p style='text-align:center'>Ocurrió algún
error al subir el fichero de ruta y no pudo guardarse.</p><br /> <a
href='modificar_ruta.php?ruta='");
        echo $id;
        echo ("'><b>Volver</b></a>");
    }
}
else
{
    if($n==1)
    {
        echo ("<p style='text-align:center'>Ya existe una
ruta con el nombre indicado. Modifique el nombre de ruta y el del
archivo y vuelva a intentarlo.</p><br /><a
href='modificar_ruta.php?ruta='");
        echo $id;
        echo ("'><b>Volver</b></a>");
    }else
    {
        if($s==1)
        {
            echo ("<p style='text-align:center'>No se ha
dado de alta la ruta indicada. El tipo de archivo cargado no coincide
con el indicado.<br/><a href='alta_ruta.php'>Volver</a></p>");
        }
    }
}

    echo ("<br /><table border='0' width='597px'><tr><td
width='500px'>&nbsp;</td><td width='17px'><img
src='../images/user.jpg' /></td><td valign='center' width='80px'><b>
");
    echo ($use);
    echo (" | <a href='desconexion.php' style='text-
decoration:none'>Salir</a> </b></td><tr></table>");
}
?>

```

10.2.12 [busqueda.php] Búsqueda de rutas

```
<h3>Listado de rutas</h3>
```

```

<br />
<table width="606" border="0" cellpadding="0" cellspacing="0">
<tr>
<td width="239" bgcolor="#8288A6"><span class="Estilo7
Estilo9">|Nombre</span></td>
<td width="90" bgcolor="#8288A6"><span class="Estilo7
Estilo9">|Fecha</span></td>
<td width="80" bgcolor="#8288A6"><span class="Estilo7
Estilo9">|Velocidad</span></td>
<td width="70" bgcolor="#8288A6"><span class="Estilo7
Estilo9">|Error</span></td>
<td width="80" bgcolor="#8288A6"><span class="Estilo9
Estilo7">|Tiempo Real </span></td>
</tr>
<?
    $ruta=$_REQUEST['nombre_ruta'];
    $_pagi_sql = "SELECT * FROM pfc WHERE nombre LIKE '%$ruta%'
ORDER BY fecha";
    $_pagi_cuantos = 8;
    $_pagi_nav_num_enlaces = 6;
    $_pagi_nav_anterior = "<img src='../images/anterior.jpg'
border='0'>";
    $_pagi_nav_siguiete = "<img src='../images/siguiente.jpg'
border='0'>";
    $_pagi_nav_primera="<img src='../images/primero.jpg'
border='0'>";
    $_pagi_nav_ultima="<img src='../images/ultimo.jpg'
border='0'>";
    $_pagi_separador = " ";
    $_pagi_nav_estilo = "paginacion";

    include("paginator.inc.php");
    $b=0;

    while($rowComen = mysql_fetch_array($_pagi_result))
    {
        if($i%2 != 0)
        {
            echo("<tr><td bgcolor='#BBC0D7'> ");
            echo("&nbsp;&nbsp;&nbsp;");
            echo("<a href='visualizar_ruta.php?id='");
            echo($rowComen["id"];
            echo("&tipo=r' style='text-
decoration:none'>");
            echo utf8_encode($rowComen["nombre"]);
            echo("</a></td><td bgcolor='#BBC0D7'>");
            $fecha = substr($rowComen["fecha"],0,10);
            $fecha = preg_replace("/([0-9]{4})[\/|\\-]([0-
9]{2})[\/|\\-]([0-9]{2})/", "\$3-\$2-\$1", $fecha);
            echo("&nbsp;&nbsp;&nbsp;");
            echo($fecha);
            echo("</td><td bgcolor='#BBC0D7'><div
align='center'>");
            if( $rowComen["tipo"]=="v" ||
$rowComen["tipo"]=="w" )
            {
                echo("<a
href='visualizar_ruta.php?id='");
                echo($rowComen["id"];
                echo("&tipo=v'><img
src='../images/velinfopeq2.jpg' border='0'></a>");

```

```

    }
    echo ("</div>");
    echo ("</td><td bgcolor='#BBC0D7'><div
align='center'>");
    if( $rowComen["tipo"]=="e" ||
$rowComen["tipo"]=="w" )
    {
        echo ("<a
href='visualizar_ruta.php?id=');
        echo $rowComen["id"];
        echo("&tipo=e'><img
src='../images/errorinfopeq2.jpg' border='0'></a>");
    }
    echo ("</div>");
    echo ("</td><td bgcolor='#BBC0D7'><div
align='center'>");
    if( $rowComen["tipo"]=="t" )
    {
        echo ("<a
href='visualizar_ruta.php?id=');
        echo $rowComen["id"];
        echo("&tipo=t'><img
src='../images/tiemporealinfopeq2.jpg' border='0'></a>");
    }
    echo ("</div></td></tr>");
    $b=1;
}
else
{
    echo ("<tr><td bgcolor='#DADDE4'> ");
    echo ("&nbsp;&nbsp;&nbsp;");
    echo ("<a href='visualizar_ruta.php?id=');
    echo $rowComen["id"];
    echo ("&tipo=r' style='text-
decoration:none'>");
    echo utf8_encode($rowComen["nombre"]);
    echo ("</a></td><td bgcolor='#DADDE4'>");
    $fecha = substr($rowComen["fecha"],0,10);
    $fecha = preg_replace("/([0-9]{4})[\/|\\-]([0-
9]{2})[\/|\\-]([0-9]{2})/", "\$3-\$2-\$1", $fecha);
    echo ("&nbsp;&nbsp;&nbsp;");
    echo $fecha;
    echo ("</td><td bgcolor='#DADDE4'><div
align='center'>");
    if( $rowComen["tipo"]=="v" ||
$rowComen["tipo"]=="w" )
    {
        echo ("<a
href='visualizar_ruta.php?id=');
        echo $rowComen["id"];
        echo("&tipo=v'><img
src='../images/velinfopeq.jpg' border='0'></a>");
    }
    echo ("</div>");
    echo ("</td><td bgcolor='#DADDE4'><div
align='center'>");
    if( $rowComen["tipo"]=="e" ||
$rowComen["tipo"]=="w" )
    {
        echo ("<a
href='visualizar_ruta.php?id=');

```

```

        echo $rowComen["id"];
        echo("&tipo=e"><img
src='../images/errorinfopeq.jpg' border='0'></a>");
    }
    echo ("</div>");
    echo ("</td><td bgcolor='#DADDE4'><div
align='center'>");
        if( $rowComen["tipo"]=="t" )
        {
            echo ("<a
href='visualizar_ruta.php?id=");
            echo $rowComen["id"];
            echo("&tipo=t"><img
src='../images/tiemporealinfopeq.jpg' border='0'></a>");
        }
        echo ("</div></td></tr>");
        $b=1;
    }
    $i=$i+1;
}
if($b==0){
    echo ("<tr><td bgcolor='#DADDE4' colspan='5'><div
align='center'><b>NO SE HAN ENCONTRADO
COINCIDENCIAS</b></div></td><td>&nbsp;</td></tr>");
}
?>
</table>
<? echo"<p style='text-align:left'><b>Páginas:</b>
".$_pagi_navegacion."</p>"; ?>
<? if($ruta==""){ echo ("<p style='text-align:center'>DEBE
INTRODUCIR UN NOMBRE DE RUTA PARA PODER REALIZAR LA BÚSQUEDA</p>"); }
?>
</div>

```

10.2.13 [control.php / s_pda.php] Comunicación con la PDA

```

<script>

function dispositivoa() {
    if(document.form1.a.value == "ACTIVO")
    {
        document.form1.a.value = "INACTIVO";
    }
    else
    {
        document.form1.a.value = "ACTIVO";
    }
}

function dispositivob() {
    if(document.form1.b.value == "ACTIVO")
    {
        document.form1.b.value = "INACTIVO";
    }
    else
    {
        document.form1.b.value = "ACTIVO";
    }
}

```

```

}

<?
$a = $_REQUEST["a"];
$b = $_REQUEST["b"];

$aa = file_get_contents("../rutas/dispositivoa.txt"); //Guardamos
archivo.txt en $archivo
$bb = file_get_contents("../rutas/dispositivob.txt"); //Guardamos
archivo.txt en $archivo

    if($a=="")
    {
        $a=$aa;
    }

    if($b=="")
    {
        $b=$bb;
    }

    $fo = fopen("../rutas/dispositivoa.txt","w");
    fputs($fo,$a);
    fclose($fo);

    $fo = fopen("../rutas/dispositivob.txt","w");
    fputs($fo,$b);
    fclose($fo);
?>
</script>
</head>

<body>
<h3> Control de dispositivos </h3>
<p style="text-align:justify">
<form name="form1" method="post" action="control.php">
<table width="310" border="0">
    <tr>
        <td width="13">&nbsp;</td>
        <td width="90"><b>Dispositivo 1: </b></td>
        <td><input type="text" name="a" value="<? echo $a; ?>"
style="border:none;" /></td>
        <td width="10">&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>
            <input type="button" value="activar / desactivar"
onClick="dispositivoa()" class="boton" /></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td><b>Dispositivo 2: </b></td>
        <td><input type="text" name="b" value="<? echo $b; ?>"
style="border:none;" /></td>
        <td>&nbsp;</td>
    </tr>
    <tr>

```

```

        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>
            <input type="button" value="activar / desactivar"
onClick="dispositivob()" class="boton"/>

        </td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td colspan="2" rowspan="2"><p><input type="submit" class="boton"
value="Cambiar estado"/></p>
        <p><table border="0" bordercolor="#2D536F" bgcolor="#C6D1DF"
cellpadding="0" cellspacing="0">
            <tr><td width="9">&nbsp;</td>
            <td width="231"> <div align="justify">
                <p>
                    Para variar el estado de los dos
dispositivos debe pulsar en el botón <em>activar/desactivar</em> del
dispositivo y seguidamente pulsar en el botón <em>Cambiar
estado</em>.</p>
                </div></td><td width="10">&nbsp;</td><td width="23"
bgcolor="#FFFFFF">&nbsp;</td>
            </tr></table></p></td>
        <td>&nbsp;</td>
    </tr>
</table>
</form>
</p>
</body>

```

10.2.13.1 s_pda.php

```

<?php

$a =
file_get_contents("http://www.elperrodepavlov.com/PFC/rutas/dispositiv
oa.txt"); //Guardamos archivo.txt en $archivo
$b =
file_get_contents("http://www.elperrodepavlov.com/PFC/rutas/dispositiv
ob.txt"); //Guardamos archivo.txt en $archivo

?>
<br/><br/><br/><br/>
<table width="301" border="0">
    <tr>
        <td width="48">&nbsp;</td>
        <td width="73"><span class="Estilo18"><font
size="1px"><b>Dispositivo 1: </b></font></span></td>
        <td colspan="3"><div align="left"><span class="Estilo18"><font
size="1px"><b><? echo $a; ?></b></font></span></div></td>
        <td width="36">&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td width="14">&nbsp;</td>
        <td width="45">&nbsp;</td>
        <td width="59">&nbsp;</td>
    </tr>

```

```

        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td><span class="Estilo18"><font size="1px"><b>Dispositivo 2:
</b></font></span></td>
        <td colspan="3"><div align="left"><span class="Estilo18"><font
size="1px"><b>? echo $b; ?</b></font></span></div></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td colspan="4" rowspan="2"><p>&nbsp;</p>
        <p align="center">&nbsp;</p></td>
        <td>&nbsp;</td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
</table>
</p>
</body>

```

10.2.14 [acceso_gestion.php/acceso_correcto.php/desconexión.php] Control de acceso a la gestión de rutas

```

<h3>Acceder a Gestión de rutas</h3>
<br />

    <form action="acceso_correcto.php">
        <p><b>Usuario:</b><br />
        <input name="usuario" type="text" style="border:1px
solid #666666; font-size:12px" size="30" /></p>
        <p><b>Contraseña:</b><br />
        <input name="password" type="password"
style="border:1px solid #666666; font-size:12px" size="30" /></p>
        <input name="reg" class="boton" value="Aceptar"
type="submit" />
    </form>

```

10.2.14.1 acceso_correcto.php

```

<?php

$result=mysql_query("select * from admindb",$link);

$use = $_REQUEST['usuario'];
$pas = $_REQUEST['password'];
$b = 0;

```

```

        if(($use == "") || ($pas == "")){

            echo "<p align='center'><b>Debe rellenar los dos campos
[Usuario y Contraseña].</b><br /><img
src='../images/denied.jpg'></p>";
            echo("<p align='center'><b><a href='acceso_gestion.php'
style='text-decoration:none'>Volver</a></b></p>");

        }else{

            while($row = mysql_fetch_array($result)) {
                if(($use == $row["usuario"]) && ($pas ==
$row["contrasena"])){

                    printf("<p align='center'><b>Identificación
correcta</b><br /><img src='../images/tic.png'></p>");
                    printf("<p align='center'><a href='gestion.php'
style='text-decoration:none'><b>:: Acceder a la Gestión de rutas
::</b></a></p>");

                    $_SESSION['aduser']=$row["usuario"];
                    $_SESSION['adpas']=$row["contrasena"];

                    $b = 1;
                    break;

                }

            }

            if($b == 0){

                printf("<p align='center'>El nombre de usuario y la
contraseña no coinciden o son incorrectos.<br /><img
src='../images/denied.jpg'></p>");
                printf("<p align='center'><b><a
href='acceso_gestion.php' style='text-
decoration:none'>Volver</a></b></p>");
            }

            mysql_free_result($result);
            mysql_close($link);

        }

    ?>

```

10.2.14.2 desconexión.php

```

<?
session_start(); // inicio de sesion
?>
<?php
    session_unset();//para eliminar las variables de sesion
    session_destroy();//con esto destruyes la sesion
    header("Location: acceso_gestion.php");
?>

```


Resum:

En aquest projecte es visualitza la trajectòria d'un vehicle (aeri o terrestre) en una pàgina web. Per això es disposa d'una PDA (*Personal Digital Assistant*), en la qual es té informació actualitzada de la posició i de la velocitat d'aquest vehicle. Aquestes dades són obtingudes d'un sistema que combina la navegació inercial i el GPS (*Global Position System*), els quals estimen de manera precisa la trajectòria del vehicle.

A fi d'oferir una visualització en temps real, versàtil, accessible i amigable a l'usuari de la trajectòria del vehicle, s'ha desenvolupat un sistema de visualització on-line que proporciona un millor rendiment en comparació amb la qual es venia fent en la PDA. Per a dur-lo a terme s'implementa una interfície d'usuari en la PDA que ens permet transmetre aquesta informació via WIFI a la pàgina web, d'igual forma al servidor web es crea una interfície que interpreta i gestiona aquestes dades per a posteriorment ser graficats a *Google Maps*.

Resumen:

En este proyecto se visualiza la trayectoria de un vehículo (aéreo o terrestre) en una página web. Para ello se dispone de una PDA (*Personal Digital Assistant*), en la cual se tiene información actualizada de la posición y la velocidad de dicho vehículo, estos datos son obtenidos de un sistema que combina la navegación inercial y el GPS (*Global Position System*) los cuales estiman de manera precisa la trayectoria del vehículo.

Con el fin de ofrecer una visualización en tiempo real, versátil, accesible y amigable al usuario de la trayectoria del vehículo, se ha desarrollado un sistema de visualización on-line que proporciona un mejor rendimiento en comparación con la que se venía haciendo en la PDA. Para llevarlo a cabo se implementa una interfaz de usuario en la PDA que nos permite transmitir esta información vía WIFI a la página web, de igual forma en el servidor web se crea una interfaz que interpreta y gestiona estos datos para posteriormente ser graficados en *Google Maps*.

Summary:

This project displays the path of a vehicle (aircraft or land vehicle) in a web page. To do this we have a PDA (*Personal Digital Assistant*), where the information is updated with the position and the speed of the vehicle, these dates are obtained from a system that combines inertial navigation and GPS (*Global Position System*) which estimates an optimal position of the vehicle.

In order to provide a real-time display of the vehicle path, versatile, accessible and user friendly, we have developed an on-line interface that provides better performance compared to the one we had done on the PDA. To do this we implemented a user interface on the PDA that allows us to transmit this information wirelessly to the web server, where we have created a interface that interprets and manages these dates to then be plotted on *Google Maps*.