

ESCUELA TÉCNICA SUPERIOR DE INGENIERIA DE TELECOMUNICACION  
UNIVERSIDAD POLITECNICA DE CARTAGENA



**Proyecto Fin de Carrera**

**Aplicación distribuida para el seguimiento, control y  
localización de flotas de empresas de logística**



AUTOR: Sergio Moral Ruiz  
DIRECTOR: Javier Vales Alonso

Enero 2006



<b>Autor</b>	Sergio Moral Ruiz
<b>E-mail del Autor</b>	<a href="mailto:semoru@hotmail.com">semoru@hotmail.com</a>
<b>Director(es)</b>	Javier Vales Alonso
<b>E-mail del Director</b>	<a href="mailto:javier.vales@upct.es">javier.vales@upct.es</a>
<b>Título del PFC</b>	Aplicación distribuida para el seguimiento, control y localización de flotas de empresas de logística
<b>Descriptor</b>	Localización, aplicaciones distribuidas

### Resumen

Partiendo de la problemática del control de flotas de transporte, los sistemas de localización -pensados en un principio para fines militares- han supuesto una revolución para afrontar dicho problema. Cada vez más PYMES de logística recurren a la localización de sus vehículos no sólo como control interno de la empresa, sino por los nuevos servicios que de ello pueden derivar. Diversas tecnologías son aplicables para el uso de sistemas de localización tales como GPS o GSM, cuya implementación en un sistema de gestión mas generalizado puede suponer una práctica herramienta de trabajo.

En este proyecto se ha desarrollado una aplicación distribuida que implementa dos funcionalidades. Por una parte la gestión integral de los datos de una compañía de logística: clientes, pedidos, vehículos, pedidos, viajes, etc, que permite la inserción, modificación y eliminación de estos registros en una base de datos. Y, por otro lado, una herramienta de localización de flotas de transporte. Para ello se han usado diversas tecnologías tales como HTML, PHP, AMFPHP, Flash, KML, MySQL y Java. Esta última ha sido la usada para implementar el sistema de localización, mediante una API de Vodafone que permite la localización de teléfonos móviles mediante llamadas a RedBox (servicio de conexión de datos en movilidad de Vodafone) que comunica directamente con la red GSM de dicha compañía. El resultado son las coordenadas geográficas aproximadas donde se encuentra el teléfono móvil, asociado a un conductor y un vehículo. Mediante la interfaz de usuario diseñada (con el resto de tecnologías), se procesan los valores almacenados en la base de datos, construyendo una estructura de posiciones interpretables por la aplicación Google Earth.

<b>Titulación</b>	Ingeniería Técnica de Telecomunicación, especialidad Telemática
<b>Intensificación</b>	-
<b>Departamento</b>	Tecnologías de la Información y las Comunicaciones

*A mis padres y familiares por estar ahí.*

*A mis amigos y compañeros porque su apoyo ha sido fundamental.*

*Especialmente agradecido a Rubén y Juan Diego que me ofrecieron esta oportunidad y su ayuda en todo momento.*

# Tabla de contenidos

## 1. Introducción

1.1	Problema propuesto .....	I
1.2	Posibles problemas existentes.....	2
1.3	Alternativas y solución propuesta .....	4
1.3.1	¿Qué sistema de bases de datos usar?.....	5
1.3.2	Bases de datos Access .....	5
1.3.3	Bases de datos MS SQL Server.....	6
1.3.4	Bases de datos MySQL .....	6
1.3.5	Bases de datos PostgreSQL.....	7
1.3.6	Elección BBDD .....	7
1.3.7	Elección lenguaje.....	7
1.3.8	GPS .....	8
1.3.9	GSM .....	8
1.3.10	Principales diferencias GPS – GSM.....	9
1.3.11	Conclusión tecnología de localización .....	9
1.3.12	Solución básica de este proyecto.....	9

## 2. Arquitecturas.....11

2.1	Tecnologías utilizadas .....	11
2.2	Estructura de la base de datos.....	19
2.2.1	Particularidades de la implementación en Flash con respecto a la bbdd.....	21
2.3	Integración del sistema de localización.....	24

<b>3. Manual de usuario</b> .....	29
3.1 Introducción.....	29
3.2 Acceso .....	30
3.3 Menú Inicio .....	31
3.4 Gestión de clientes, conductores y móviles.....	32
3.5 Vehículos.....	34
3.5.1 Historial y avisos .....	35
3.5.2 Rentabilidad.....	36
3.6 Inserción y modificación de registros.....	37
3.7 Eliminación de registros .....	38
3.8 Pedidos .....	39
3.8.1 Nuevo pedido .....	40
3.8.2 Pedidos Pendientes .....	41
3.8.3 Pedidos Realizados.....	42
3.9 Viajes.....	43
3.9.1 Nuevo viaje.....	44
3.9.2 Viajes pendientes.....	46
3.9.3 Viajes en curso .....	47
3.9.4 Viajes realizados.....	48
3.9.5 Localización .....	49
3.9.6 Cuentas .....	50
3.10 Anexo .....	51
3.10.1 Consideraciones especiales .....	51
3.10.2 Leyenda de colores .....	51
3.10.3 Definiciones.....	52
<b>4. Manual de administrador</b> .....	53
4.1 Introducción.....	53
4.2 Instalación de los módulos .....	54
4.2.1 Servidor Web + Php + MySQL.....	54
4.2.2 AMFPHP .....	54
4.2.3 Google Earth.....	55
4.2.4 Instalación de la aplicación .....	55
4.2.5 Anexo de instalación de la base de datos MySQL + AMFPHP .....	55
4.3 Herramienta de configuración .....	56
4.3.1 Acceso .....	56
4.3.2 Menú.....	57
4.3.3 Configurar Tablas.....	58
4.3.4 Configurar perfiles .....	59
4.3.5 Añadir usuarios.....	60
4.3.6 Eliminar usuarios.....	61
4.4 Sistema de localización .....	62

---

<b>5. Conclusión</b> .....	63
<b>6. A</b> .....	65
<b>7. B</b> .....	67
<b>8. Bibliografía</b> .....	68

# Lista de Figuras

## 1. Introducción

1.1 Aplicación Rutasoft.....	2
1.2 Aplicación Interlogística .....	3
1.3 Boceto del esquema básico.....	10

## 2. Arquitectura e implementación del sistema

2.1 Esquema de funcionamiento en primer nivel.....	12
2.2 Esquema de funcionamiento con HTML, PHP y MySQL.....	14
2.3 Primera captura del modelo desarrollado en PHP.....	15
2.4 Segunda captura del modelo desarrollado en PHP.....	15
2.5 Captura del modelo diseñado en Flash.....	17
2.6 Decisión de la arquitectura .....	18
2.7 Esquema de la base de datos. ....	19
2.8 Estructura ejemplo de la tabla 'Pedidos'.....	20
2.9 Esquema de 'conf_vista' y 'conf_perfiles'.....	23
2.10 Esquema de comunicación de Plug InRed. ....	25
2.11 Esquema de comunicación con sistema de localización implementado. ....	26
2.12 Esquema de comunicación con interpretación de coordenadas. ....	28

## 3. Manual de usuario

3.1 Acceso .....	30
3.2 Menú inicio.....	31
3.3 Gestión de datos: clientes, conductores y móviles.....	32
3.4 Gestión de vehículos.....	34
3.5 Gestión de avisos e historial de los vehículos. ....	35

3.6 Visualización de las rentabilidades. ....	36
3.7 Inserción y modificación de registros.....	37
3.8 Eliminación de registros.....	38
3.9 Menú Pedidos .....	39
3.10 Gestión de un nuevo pedido. ....	40
3.11 Gestión de los pedidos pendientes.....	41
3.12 Gestión de los pedidos realizados.....	42
3.13 Menú de viajes.....	43
3.14 Primer paso en la creación de un nuevo viaje. ....	44
3.15 Segundo paso en la creación de un nuevo viaje. Elección de los pedidos. ....	45
3.16 Gestión de viajes pendientes. ....	46
3.17 Gestión de viajes en curso. ....	47
3.18 Gestión de viajes realizados. ....	48
3.19 Gestión de viajes localizables.....	49
3.20 Gestión de cuentas.....	50

## 4. Manual de administrador

4.1 Acceso a configuración. ....	56
4.2 Menú inicial.....	57
4.3 Configuración de tablas.....	58
4.4 Menú de configuración de perfiles.....	59
4.5 Creación de nuevos usuarios. ....	60
4.6 Eliminación de usuarios. ....	61



# Lista de Tablas

## 1. Introducción

1.1 Comparativa de bases de datos extraída de [1]. .....	5
--	---

# Capítulo 1

## Introducción

### 1.1 Problema propuesto

Este proyecto aborda la construcción de una aplicación para PYMES de logística que permita:

- Gestión de datos.
- Servicio de localización de flotas.

Es decir, una aplicación que conste de funciones básicas de tratamiento de información tales como inserción, modificación y eliminación de registros. De forma que podamos administrar los clientes de la empresa, los pedidos que hacen dichos clientes, etc; para poder llevar a cabo ciertas operaciones que faciliten la obtención de resultados estadísticos tales como la rentabilidad obtenida a lo largo del año o el número de pedidos realizados cada mes, entre otros.

Las ventajas de esta aplicación repercuten en la optimización del control que pueda ejercer la PYME sobre su propia actividad, así como en el ahorro de tiempo que conlleva la automatización de tareas.

En lo referente al tema de localización se pretende obtener un sistema que permita verificar la situación geográfica de cada una de las flotas. Esto favorece el control que desarrolla la PYME, antes mencionado, sobre sus vehículos desplegados, así mismo, supone la apertura de nuevos servicios a los clientes, como por ejemplo, ofrecer un estado de situación de los pedidos realizados por cada uno, y la visualización geográfica de los pedidos en curso.

En definitiva, se requiere una aplicación que procure el aumento de beneficios a través de:

- Agilizar la gestión.
- Automatización de tareas.
- Obtención de estadísticos.
- Mejorar el control de la actividad de la PYME.
- Integrar un sistema de localización de flotas.

Así pues, sería conveniente unificar en una única aplicación toda la funcionalidad que satisfaga los servicios propuestos, de forma que el usuario de dicha aplicación pueda realizar un control avanzado tanto de la actividad comercial de la empresa como de la situación de la flota activa de forma clara y sencilla, es decir, sin necesidad de una formación extra del personal.

## 1.2. Posibles soluciones existentes

Una posible solución es “Rutasoft”, que ofrece una aplicación que cumple con los requisitos. No obstante, tras ponerse en contacto con dicha empresa, no se obtuvo una respuesta en relación con las expectativas.

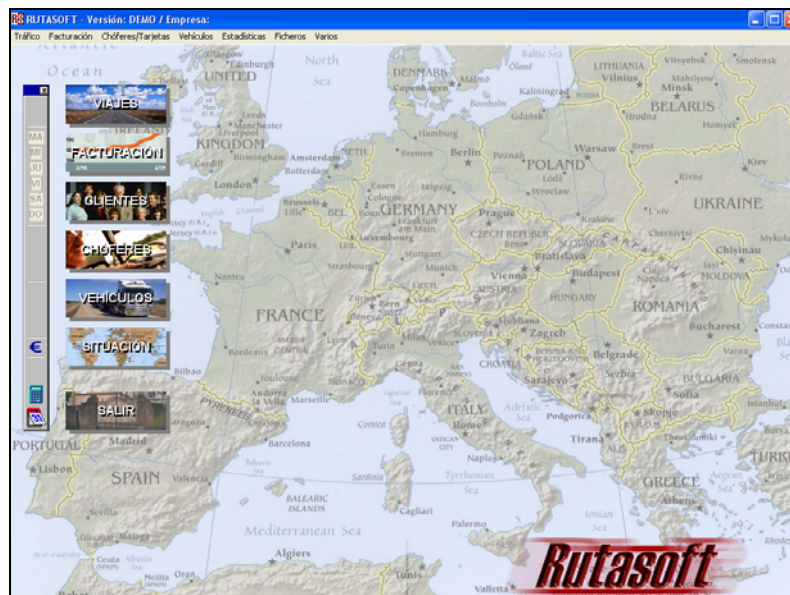


Figura 1.1 Aplicación Rutasoft.

Existen diversos programas de contabilidad, que podrían cubrir las necesidades de gestión que se proponen, sin embargo, tampoco incluyen un sistema de localización de flotas, y además presentan una característica que podemos considerar ventaja e inconveniente, y es que son programas mas enfocados a gestión de empresas en general, lo cual presenta mas opciones de las, “a priori” necesarias, por lo que se podrá abordar otros parámetros en principio no exigidos, sin embargo también aporta opciones extra en principio no dedicadas a PYMES de logística cuya no utilización afecta a la claridad de uso.

En lo referente a servicios de localización podemos encontrar diferentes soluciones que proponen empresas privadas como Interlogística, que propone un servicio de localización a través de GPS/GPRS y a través de GSM.

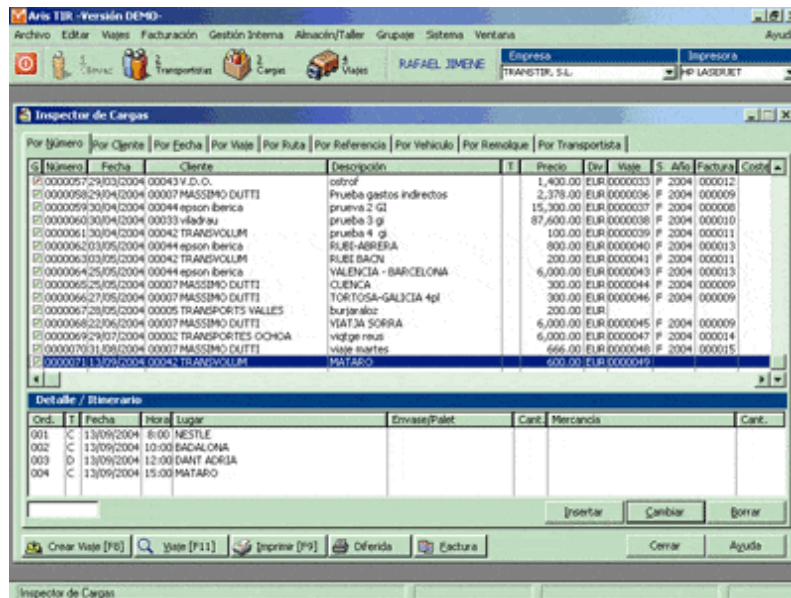


Figura 1.2 Aplicación Interlogística.

Ambas tecnologías presentan diferentes características, la decisión de adoptar una u otra depende de las necesidades de la empresa. Como principales diferencias se destacan:

- GPS ofrece cobertura nacional e internacional. En cambio GSM solo ofrece cobertura nacional.
- GPS es un sistema mas preciso para obtener las coordenadas de localización. Pero requiere de GSM como tecnología de soporte para la transmisión de la información
- El sistema GPS requiere instalación de dispositivos en cada vehículo, lo cual supone mayores gastos, tanto por dicha instalación como por mantenimiento. Sin

embargo el sistema GSM necesita únicamente la presencia de un teléfono móvil, teniendo en cuenta que hoy en día es posible disponer de uno a un precio reducido y con bajo coste de mantenimiento. Esto supone un ahorro proporcional al número de vehículos a localizar en comparación con el sistema GPS.

Hay que destacar que todas las empresas privadas que ofrecen un sistema de localización basado en GSM lo hacen a través de operadoras de telefonía móvil, como Movistar o Vodafone, que también disponen a los clientes este servicio, sin embargo, las mencionadas empresas ajenas a estos operadores como Interlogística o MoviDream han realizado su propia versión, creando productos con extras para la captación del cliente, consiguiendo así aplicaciones más comerciales y dedicadas a sectores más especializados, como en este caso las PYMES de logística.

Con todo esto, no conseguimos lograr nuestro objetivo, aunque podríamos resolver las necesidades propuestas con varias aplicaciones proporcionadas por diferentes empresas, se quiere un sistema único que integre las funcionalidades principales descritas en el apartado anterior.

### **1.3. Alternativas y solución propuesta**

Llegado a este punto se planteó la posibilidad de crear una aplicación que gestione una base de datos enfocada a los parámetros necesarios utilizados por PYMES de logística, y que integre un sistema de localización de flotas.

Existen varias soluciones a este problema, las alternativas difieren principalmente en:

- Concretar un lenguaje de programación para desarrollar la aplicación.
- Escoger la base de datos a utilizar con la que se comunicará el lenguaje seleccionado.
- Elegir la tecnología de localización a implementar: GPS ó GSM.

En principio se consideraron los lenguajes de programación más populares como Java, C, C++, cualquiera de estos deberá comunicarse con la base de datos. Sin embargo como punto de partida del proyecto se pensó en desarrollarlo como una aplicación distribuida, es decir, una aplicación Web, ya que contaría con las ventajas que ofrece Internet, como la accesibilidad desde cualquier punto del mundo que disfrutara de conexión, o contar con compatibilidad con prácticamente todos los equipos conectados. Además con una sola copia en un servidor se podría usar la aplicación en diversos equipos simultáneamente sin preocuparse de instalaciones. Así pues, como primer paso se decidirá la base de datos a utilizar.

### 1.3.1. ¿Qué sistema de bases de datos usar?

Aunque se dispone de varias alternativas a la hora de decidir el sistema de bases de datos sobre el que desea trabajar para su proyecto, se debe saber que cada uno de ellos tiene sus peculiaridades que lo convierten en el más adecuado según para qué tipo de proyectos o en función de los intereses del programador.

A continuación se presenta un pequeño resumen, en modo de tabla comparativa, con los criterios principales a tener en cuenta a la hora de tomar una decisión:



























Criterios	Bases de datos			
	Access	SQL Server	MySQL	PostgreSQL
Plataforma			 / 	 / 
Velocidad				
Volumen Datos				
Integridad				
Potencia				
Coste/MB				
↑ Positivo ↓ Negativo				

Tabla 1.1 Comparativa de bases de datos extraída de [1].

### 1.3.2. Bases de datos Access

Disponibles sólo con sistema operativo Windows.

Realizar una aplicación ASP sobre bases de datos Access es recomendable en los casos en que sea especialmente cómoda la actualización de la información por el procedimiento de enviar el archivo .mdb al servidor mediante FTP.

Ahora bien, para que la aplicación sobre base de datos Access no tenga problemas, es recomendable que cumpla estas condiciones:

- El volumen de datos a manejar es pequeño. (Además así será más rápida su actualización por FTP).
- El número de visitantes simultáneos no es muy alto.

- La aplicación ASP no cambia la base de datos, simplemente muestra datos. Esto es consistente con el hecho de enviar periódicamente el archivo .mdb al servidor, pues si la aplicación ASP cambiase la base de datos, esos cambios se perderían al sobrescribirse con la nueva base de datos.

Cuando por alguno de los anteriores motivos, su aplicación no es consistente, o no va a poder cumplir alguno de los mismos, es preferible el uso de un sistema de base de datos más robusto, como SQL Server.

### 1.3.3. Bases de datos MS SQL Server

Disponibles con sistema operativo Windows.

SQL Server es el sistema de bases de datos ideal para los programadores especializados en productos Microsoft: ASP, Visual Basic, modelos de objetos componentes, etc. Además, es un sistema de base de datos perfectamente adecuado para aplicaciones críticas y con cualquier grado de complejidad.

SQL Server utiliza una parte del espacio de la base de datos para guardar el log de transacciones con los comandos pendientes, lo que asegura que, independientemente de si el programador usa o no transacciones en su código, en ningún caso la base de datos quedaría en un estado inconsistente debido a una ejecución parcial de comandos.

También ofrece otras muchas características avanzadas orientadas a mantener la integridad de la base de datos, como son los triggers, y ofrece soporte completo ACID (Atomicity Consistency Isolation Durability).

### 1.3.4. Bases de datos MySQL

Disponibles con sistema operativo Windows o Linux.

MySQL tiene como principales características su velocidad además de ser de distribución libre. Es el servidor de bases de datos más rápido de todos.

MySQL es muy utilizado en aplicaciones PHP o Perl en servidores Linux. En general, si no necesita características como transacciones, procedimientos almacenados, triggers o sentencias SQL complejas, MySQL cumplirá la misma función que otras bases de datos más potentes, pero de forma más rápida y con un coste menor.

Para aplicaciones Windows, MySQL es una alternativa económica a SQL Server pues se puede aprovechar todo el espacio para datos, mientras que SQL Server necesita una parte del espacio para el log de transacciones.

Las limitaciones de MySQL vienen dadas por sus carencias respecto de los otros sistemas de bases de datos y por el grado de criticidad de su aplicación. MySQL no es adecuada para aplicaciones críticas. Al no utilizar transacciones, un problema de cualquier tipo que interrumpiese una serie de comandos podría dejar su base de datos en un estado inconsistente, lo cual nunca ocurriría con SQL Server o PostgreSQL. Tampoco tiene triggers por lo que no se pueden establecer reglas de integridad y consistencia a nivel de servidor.

### 1.3.5. Bases de datos PostgreSQL

Disponibles con sistema operativo Windows o Linux.

PostgreSQL es el servidor de bases de datos de código abierto más potente que existe y es por tanto la alternativa a MySQL cuando se necesitan características avanzadas como transacciones, procedimientos almacenados, triggers, vistas, etc.

PostgreSQL es el servidor de bases de datos más utilizado por los programadores de servlets de Java y, en general, por todos aquellos que realizan aplicaciones cliente servidor complejas o críticas en el mundo Linux/Unix.

Para aplicaciones Windows, PostgreSQL es una alternativa económica a SQL Server y tiene similares prestaciones. Esta diferencia económica es especialmente sustancial si se necesita un Servidor Dedicado de bases de datos.

La mayor limitación de PostgreSQL viene dada por su velocidad: es el sistema de bases de datos más lento.

### 1.3.6. Elección BBDD

Tras observar los sistemas de bases de datos más populares y sus principales diferencias se considera como principales opciones MySQL y PostgreSQL, ambas de distribución libre, por lo que se consigue acotar gastos en el desarrollo de este proyecto, además de ser de los sistemas más utilizados por la comunidad de programadores, y ser compatibles con los sistemas operativos de mayor uso. A la hora de elegir entre ambos, se comprueba una mayor velocidad de la BBDD MySQL frente a PostgreSQL, por lo que finalmente se seleccionó **MySQL** como base de datos para este proyecto.

### 1.3.7. Elección lenguaje

MySQL es accesible desde una gran cantidad de lenguajes de programación, como los anteriormente citados (Java, C++,...), incluso se entiende muy bien con lenguajes destinados a aplicaciones distribuidas como PHP que ofrece funciones específicas para la



conexión con MySQL proporcionando un procesamiento de consultas y respuestas mas rápido que otros lenguajes. Por lo tanto, y tras estas consideraciones se decide usar **PHP** para acceder a la base de datos.

Una vez decidido el “esqueleto arquitectónico”, se ha de decidir lo necesario referente al sistema de localización. Lo principal es escoger la tecnología de localización mas apropiada, actualmente podemos elegir entre dos opciones, GPS y GSM. A continuación se comenta una breve explicación de ambos sistemas así como las principales diferencias.

### 1.3.8. GPS

El Global Positioning System (GPS) o Sistema de Posicionamiento Global originalmente llamado NAVSTAR, es un Sistema Global de Navegación por Satélite (GNSS) el cual permite determinar en todo el mundo la posición de una persona, (en todo su conjunto incluyendo sus extremidades de ahí que se denomine global) un vehículo o una nave, con una desviación de cuatro metros. El sistema fue desarrollado e instalado, y actualmente es operado, por el Departamento de Defensa de los Estados Unidos.

El GPS funciona mediante una red de satélites que se encuentran orbitando alrededor de la tierra. Cuando se desea determinar la posición, el aparato que se utiliza para ello localiza automáticamente como mínimo cuatro satélites de la red, de los que recibe unas señales indicando la posición y el reloj de cada uno de ellos. En base a estas señales, el aparato sincroniza el reloj del GPS y calcula el retraso de las señales, es decir, la distancia al satélite.

Por "triangulación" calcula la posición en que éste se encuentra. La triangulación en el caso del GPS, a diferencia del caso 2-D que consiste en averiguar el ángulo respecto de puntos conocidos, se basa en determinar la distancia de cada satélite respecto al punto de medición. Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los tres satélites. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtienen las posiciones absolutas o coordenadas reales del punto de medición. También se consigue una exactitud extrema en el reloj del GPS, similar a la de los relojes atómicos que desde tierra sincronizan a los satélites.

### 1.3.9. GSM

La localización **GSM** es un servicio ofrecido por las empresas operadoras de telefonía móvil que permite determinar, con una cierta precisión, donde se encuentra físicamente un terminal móvil determinado.

Los distintos métodos que se emplean para la localización GSM son los siguientes:

- Célula de origen (Cell of Origin), en el que se incluyen ID de célula (Cell ID) e ID de célula mejorada (Enhanced Cell ID)...

- ID de célula: La precisión de este método es de 200 metros en áreas urbanas, 2 Km en áreas suburbanas y varía entre 3 - 4km en entornos rurales.
- ID de célula mejorada: Con este método se consigue una precisión muy parecida a la que ofrece el Cell ID para zonas urbanas, y en entornos rurales ofrece sectores circulares de 550 metros.
- Diferencia de tiempo observada o E-OTD (Enhanced-Observed Timed Difference): La precisión de este método depende del número de LMUs disponibles en la red, variando entre 50 metros y 200 metros con un LMU por cada 3 estaciones base.
- Tiempo de llegada (Time of Arrival)
- Angulo de llegada (Angle of Arrival)
- Enhanced Observed Time Difference (estimación mejorada de la diferencia de tiempo)
- Asistido por GPS (Assisted GPS)

### **1.3.10. Principales diferencias GPS – GSM**

El sistema GPS presenta un margen de error de aproximadamente de 10m, mientras que GSM tiene un margen de error mayor, que depende de la zona, de unos 300m para zonas urbanas y entorno a 5 o 10 km en zonas suburbanas. Como principal ventaja de GSM se destaca que basta con disponer de un teléfono móvil, lo cual hoy en día resulta posible a unos costes muy reducidos, en comparación con el equipo que necesariamente hay que instalar para poder disfrutar de la tecnología GPS. También es cierto que el uso de GPS proporciona la opción de poder incorporar servicios extra, como seguimiento a tiempo real, o control de frío de las cámaras frigoríficas, sin embargo el mantenimiento de estos equipos y servicios también suponen un gasto adicional.

### **1.3.11. Conclusión tecnología de localización**

Con todo esto, se diseñó en principio la aplicación que permita la localización de flota a través de la tecnología GSM, principalmente por la diferencia de coste de terminales móviles frente a equipos GPS. Aun así, no se dará la espalda a esta tecnología, no cerrando puertas a una posible incorporación futura otras tecnologías.

### **1.3.12. Solución básica de este proyecto**

Ya se tiene pues, el primer “boceto” de lo que es este proyecto:

- Una aplicación Web, basada en una base de datos MySQL para el almacenamiento y procesamiento de datos, tales como inserción, modificación y eliminación de información, a través de funciones dedicadas, implementadas en PHP, con las que el usuario interactuará y accederá a la base de datos.
- Una servicio integrado de localización de flotas mediante tecnología GSM, que permitirá el control de los vehículos. Es decir, se podrá visualizar gráficamente la posición de los vehículos, así como almacenar los puntos de recorrido para poder visualizar las rutas seguidas. Esta opción de localización se puede conseguir únicamente mediante el uso del servicio de localización ofrecido por operadoras de telefonía móvil, Movistar o Vodafone, en el caso de España.

Gráficamente, el funcionamiento básico del programa:

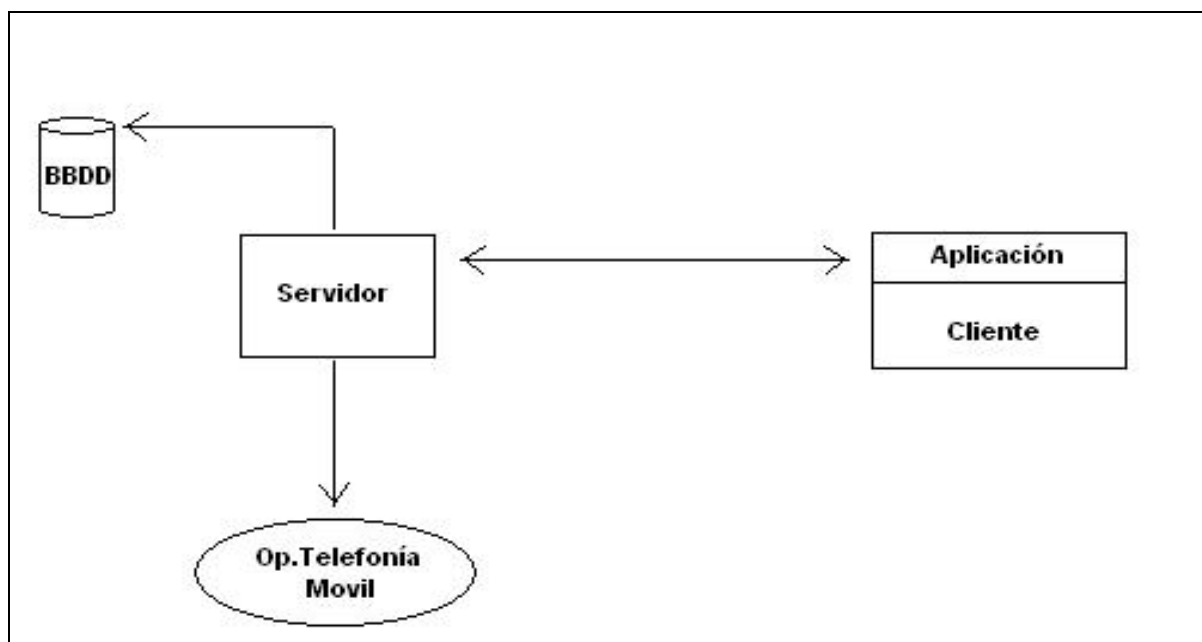


Figura 1.3 Boceto del esquema básico.

La estructura final de este proyecto se detallará en el siguiente apartado, “Arquitectura e implementación del sistema”.

## Capítulo 2

# Arquitectura e implementación del sistema

En el capítulo anterior se definió una arquitectura básica formada por MySQL, PHP y un sistema de localización basado en tecnología GSM. En este apartado se contemplan diversos aspectos referentes al funcionamiento de este proyecto:

1. Descripción de las diversas tecnologías que componen la arquitectura del sistema.
  - Funcionalidad de cada uno de los elementos que la componen
  - Representación gráfica de cómo se comunican entre sí los diferentes módulos
2. Estructura de la base de datos
3. Explicación de la forma en que se integrará el servicio de localización

## 2.1. Tecnologías utilizadas

Esta sección se detalla mediante un esquema gráfico de funcionamiento de la aplicación. En un primer nivel, se puede considerar como verá el usuario la aplicación, es

decir como una aplicación Web, que únicamente necesitará una dirección WWW para poder utilizar los servicios proporcionados. Así pues la imagen que tendrá el usuario será similar al siguiente esquema:

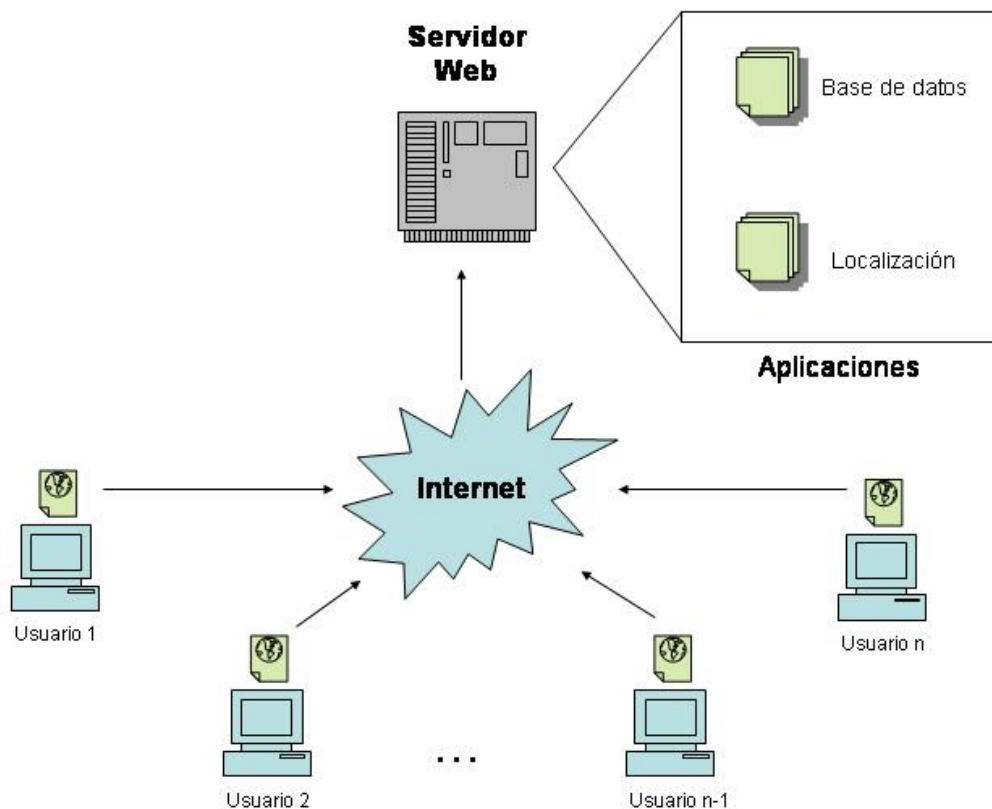


Figura2.1 Esquema de funcionamiento en primer nivel.

Dejando para su apartado correspondiente el tema de la localización, se necesita en un primer momento, un servidor Web, en el que instalar la base de datos MySQL, para poder acceder a la información desde Internet. Se pueden contemplar diferentes opciones en este apartado:

- El alquiler de un servidor a una empresa privada para su uso.
- Existen varias aplicaciones que permiten instalar un servidor en un computador propio. Se debe administrar y configurar correctamente según las necesidades.

Se considera mas apropiada la segunda opción ya que así se podrá tener total acceso al apartado de administración y configuración del servidor, con lo que se puede modelar la forma en que se proveerá la aplicación, es decir, la definición de parámetros tales como numero de conexiones permitidas, direcciones IP validas, puerto de utilización, etc.

Dentro de los programas que ofrecen la instalación propia de un servidor Web, se tiene como referencia el uso de Apache, de uso muy extendido y de libre distribución, por lo que esta será la opción mas idónea, ya que cubre con creces las necesidades y no incrementa el coste del desarrollo.

Como se propuso en la “solución básica”, se usó PHP como interfaz entre aplicación Web y el acceso a la base de datos. PHP es un lenguaje de programación, que se ejecuta en el servidor, es decir, el cliente solo indica que funciones se realizan o los parámetros a utilizar, pero no ejecuta ningún código localmente. Por tanto es muy importante que el servidor esté configurado de manera que se permita tanto el uso de PHP, como de MySQL.

La sistema Apache + PHP + MySQL es una herramienta muy extendida entre las soluciones existentes. Existen una gran cantidad de servidores montados sobre Linux que están compuestos por esta tecnología, por lo que existen paquetes de instalación que comprenden las tres aplicaciones. Esto se conoce con las siglas ‘LAMP’, Linux + Apache + PHP + MySQL, incluso podemos encontrar paquetes para el sistema operativo Windows, que proporcionan los mismos servicios. Este paquete se llama ‘WinLAMP’, consiste en una aplicación que instala:

- Apache con soporte para PHP y MySQL
- El servidor de base de datos MySQL
- Una interfaz Web, ‘PhpMyAdmin’ o ‘EsKuel’ con la poder acceder a la base de datos, y realizar multitud de operaciones.

Por tanto, esta será la herramienta inicial. El punto de partida de desarrollo se iniciará utilizando Windows XP como sistema operativo, sin embargo el objetivo es que la aplicación final sea compatible, en la mayor medida posible, con cualquier sistema operativo y con la mayoría de equipos existentes.

Por lo tanto, se usó ‘WinLAMP’. Este paquete es de libre distribución y se puede encontrar en Internet, válido para Windows 95/98/NT/2000/XP. Con una simple búsqueda se consiguen diversas direcciones donde descargar el programa para instalar. En este caso se descargó el paquete (versión 3.01) de [2].

Hay que considerar que las aplicaciones Web usan el lenguaje de marcas HTML en el que se implementa el código PHP, por lo que el primer desarrollo se realizó usando estas tecnologías, es decir, una página Web en la que mediante PHP el usuario podrá realizar operaciones en la base de datos.

Gráficamente este esquema de funcionamiento queda:

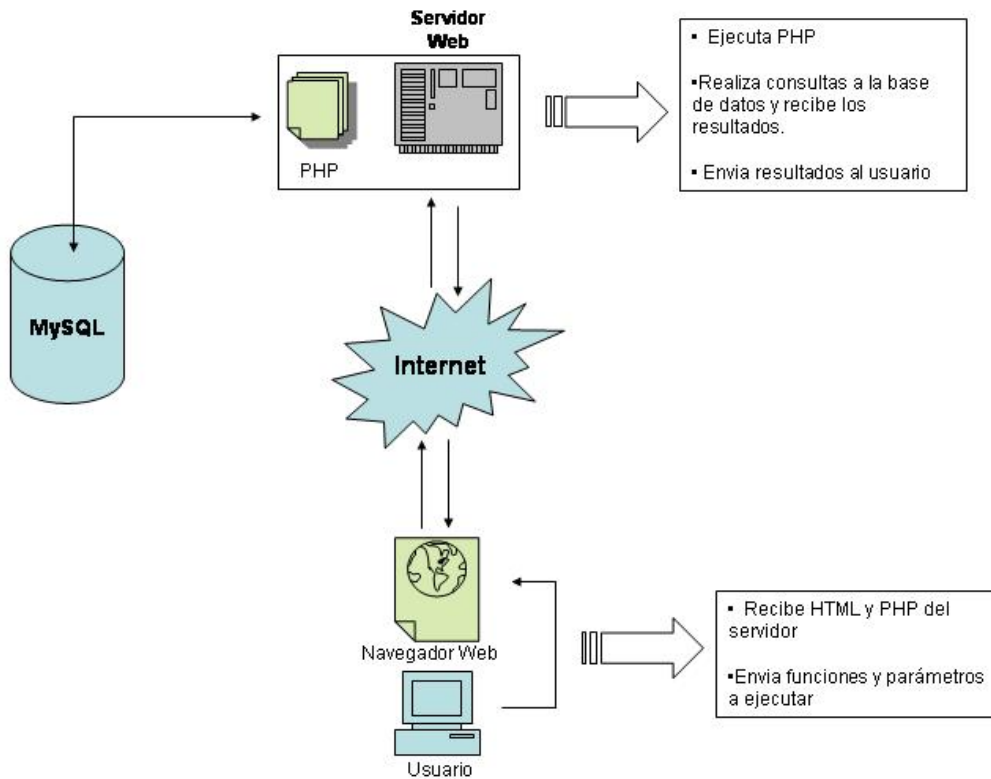


Figura 2.2 Esquema de funcionamiento con HTML, PHP y MySQL.

A partir de aquí se empezó a desarrollar una página en HTML con implementación de PHP correspondiente que ejecutara “quers” a la base de datos MySQL. La funcionalidad principal radica en permitir al usuario las siguientes funciones:

- Visualizar la información
- Insertar registros
- Modificar registros
- Eliminar registros
- Realizar búsquedas específicas
- Posibilidad de seleccionar información según género (clientes, vehículos...)

Se muestran a continuación algunas capturas de lo que es el primer boceto, en el que se implementó el campo “clientes”, y donde se podía ver, añadir y modificar registros:

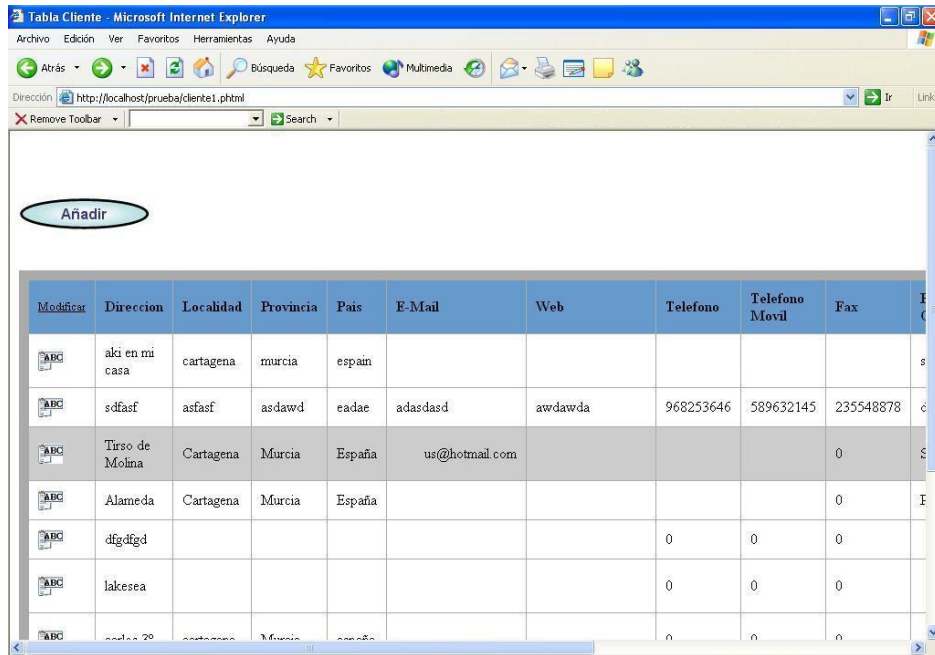


Figura 2.3 Primera captura del modelo desarrollado en PHP.

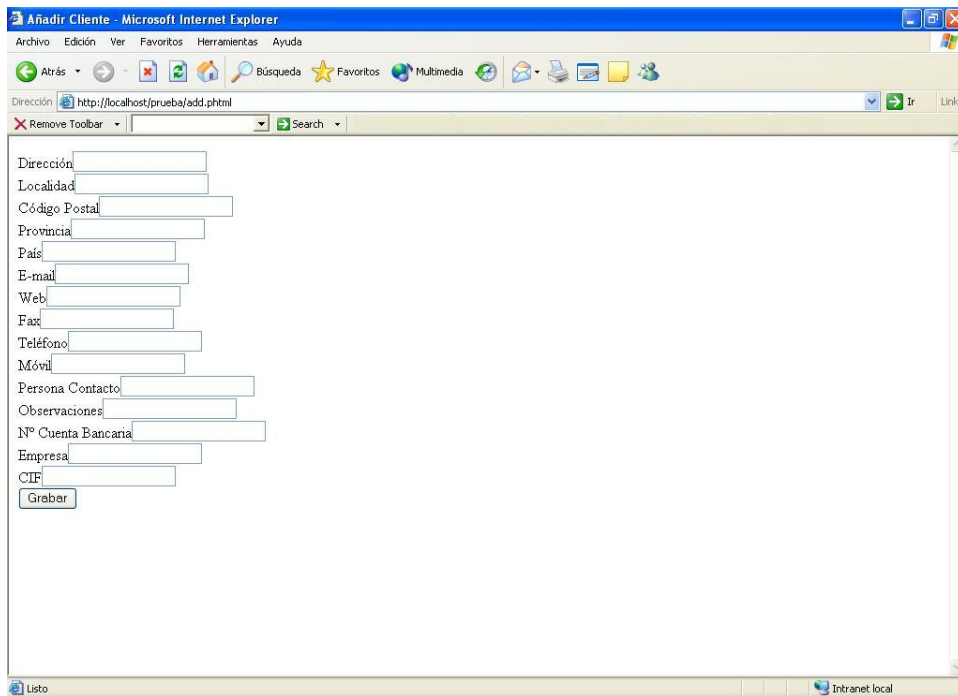


Figura 2.4 Segunda captura del modelo desarrollado en PHP.



Se puede observar que la información se muestra de una manera poco ortodoxa, ya que HTML presenta ciertos problemas con los espacios. Aun habiéndose mejorado la presentación de la información, se pensó en estudiar otras tecnologías que mejoraran la interfaz gráfica, para favorecer la claridad respecto al uso de la aplicación, incluso algún tipo de lenguaje orientado a objetos podría ayudar de manera notable al desarrollo del programa, puesto que se podría implementar con mayor eficiencia las funciones que proporcionan los servicios, y contar con las ventajas que este tipo de lenguajes ofrecen.

Existen numerosos lenguajes que mejoran la funcionalidad limitada de HTML, como JavaScript, sin embargo no se podía dejar pasar por alto la posibilidad de incorporar la tendencia actual en lo que a diseño y desarrollo de páginas Web se refiere, es decir, la aplicación de diseño Web: Flash.

Se consideró la herramienta Macromedia Flash MX 2004, para el desarrollo de la interfaz gráfica a usar en este proyecto, obteniendo mejoras muy notables con respecto a la versión anterior implementada únicamente con HTML y PHP. Sin embargo, apareció una nueva problemática, ¿cómo conectar una aplicación desarrollada en Flash con la información almacenada en la base de datos MySQL?

Tras un estudio previo se tuvieron en cuenta las siguientes soluciones:

- Usar Objetos XML
- Utilizar el método LoadVars() de ActionScript
- Utilizar Flash Remoting & AMFPHP

La problemática era conocida, y las dos primeras opciones eran soluciones propuestas para salir del paso, puesto que en principio Macromedia no había implementado una herramienta dedicada a la comunicación con las bases de datos. Sin embargo, con la expansión de la tecnología Flash, Macromedia proporcionó una actualización que prometía solventar esta cuestión, nos referimos al plug-in “Flash Remoting”. A continuación se expone una breve explicación sobre esta herramienta proporcionada [3]:

*“Macromedia Flash Remoting MX provee la conexión entre Macromedia Flash y su servidor de aplicaciones web, lo cual hace de la creación de aplicaciones dinámicas de Internet una tarea fácil y rápida. Con su modelo de programación potente pero sencillo, se puede integrar fácilmente contenido dinámico de Macromedia Flash con aplicaciones creadas con Macromedia ColdFusion MX, Microsoft .NET, Java, así como servicios web basados en SOAP.”*

Sin embargo Flash Remoting usa objetos Flash específicos que no son capaces de comunicarse compatiblemente en MySQL, por lo tanto no nos ofrece una solución válida a nuestro problema. No obstante, recientemente surgió la herramienta AMFPHP que parece ser la forma de poder implementar correctamente Flash y MySQL.

A continuación se muestra una breve descripción de AMFPHP traducida de la página [4]: *“AMFPHP es un gateway de Flash Remoting de código abierto, rápido y fiable. Flash Remoting es una tecnología construida en el núcleo del reproductor Flash Player*

que permite la comunicación entre el servidor y el cliente indistintamente. Los sistemas basados en objetos XML presentan mayores problemas a la hora de manejar los datos, la manera de integrar estos objetos en la aplicación, así como en la debuggación. Con Flash Remoting se pueden realizar llamadas a métodos remotos desde el cliente Flash consiguiendo que los argumentos acaben en el lenguaje remoto nativo, y regresarán a Flash correctamente tipeados sin que haya confusión con la serialización de la información.

También ofrece un 'NetConnection debugger' que muestra exactamente que se está enviando entre el cliente y el servidor. Remoting usa AMF, un sencillo formato binario que limita el exceso de volumen de los paquetes, por lo que logra que el intercambio de información sea mucho más rápido que usando objetos XML. Remoting es el camino idóneo para construir aplicaciones distribuidas robustas, rápidas y seguras. AMFPHP es una completa implementación de Remoting para PHP con multitud de características enfocadas a la creación de aplicaciones remotas."

Tras un tiempo migrando el boceto inicial a la tecnología Flash, se pudo comprobar la mejoría evidente entre las diferentes versiones desarrolladas. Siendo Flash una herramienta mucho más potente, tanto a la hora de diseñar la interfaz gráfica, como a la hora de manipular la información. Además Flash cuenta con un lenguaje orientado a objetos denominado 'ActionScript' que favorece enormemente la implementación de los servicios y funcionalidades que requiere este proyecto.

Se puede comprobar la mejora obtenida, en la siguiente captura de pantalla:

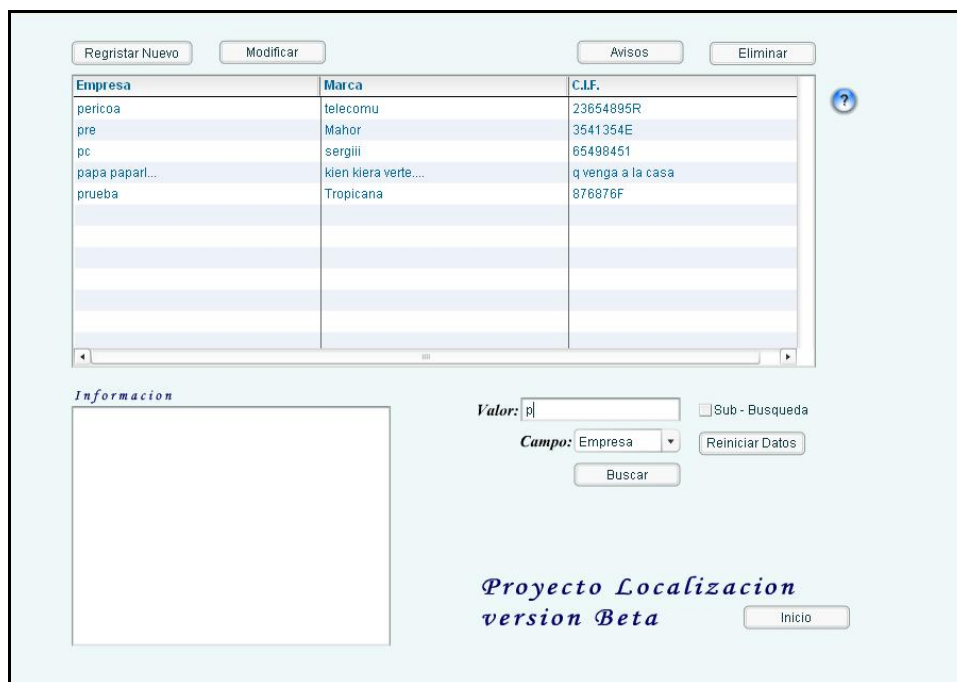


Figura 2.5 Captura del modelo diseñado en Flash.

Así pues, ésta será la tecnología que se usará para el desarrollo del proyecto final:

- Servidor Web : Apache
- Base de Datos: MySQL
- Interfaz entre Web y base de datos: Lenguaje del lado del servidor, PHP
- Aplicación Web: Flash integrada en HTML
- Interfaz entre Flash y PHP: AMFPHP

La visualización grafica del sistema de comunicación entre los diferentes módulos queda representada en el siguiente esquema:

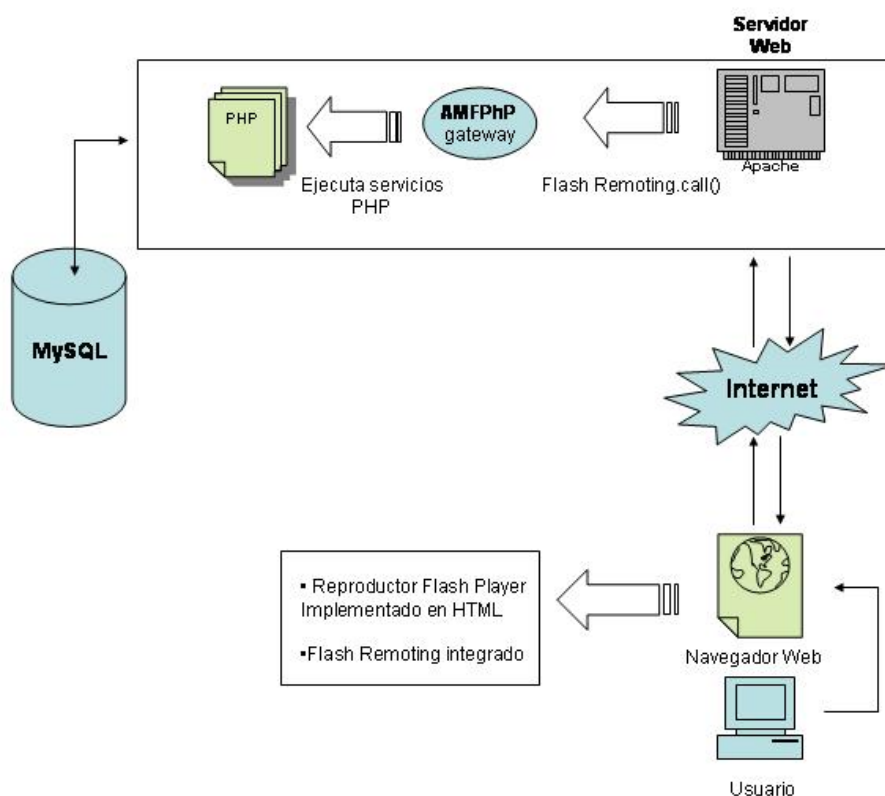


Figura 2.6 Decisión de la arquitectura

El funcionamiento de nuestra aplicación se lleva a cabo de la siguiente manera:

El usuario accede a través de una dirección de Internet al servidor, descargando así la aplicación en formato 'swf'. Para este formato es necesario el reproductor gratuito de Macromedia, Flash Player.

A través de la aplicación descargada el usuario gestiona la información de forma que cuando requiere interactuar con la base de datos, Flash Remoting realiza una llamada al gateway de AMFPHP, que alberga las distintas funciones definidas en PHP.

Estas funciones son denominadas ‘services’ y realizan las operaciones oportunas a la base de datos, capturando el resultado, y enviándolo directamente al objeto Flash Remoting, que capturarán la aplicación desarrollada.

## 2.2. Estructura de la base de datos

A continuación se representará la estructura realizada en la base de datos para almacenar de la forma más conveniente posible la información que se pueda recoger en PYMES de logística, basándose en criterios dados por una empresa del sector.

Se contemplará a continuación un esquema en el que se muestran las tablas que forman la base de datos utilizada. Se especifican únicamente aquellos campos que interfieren en la relación entre dichas tablas:

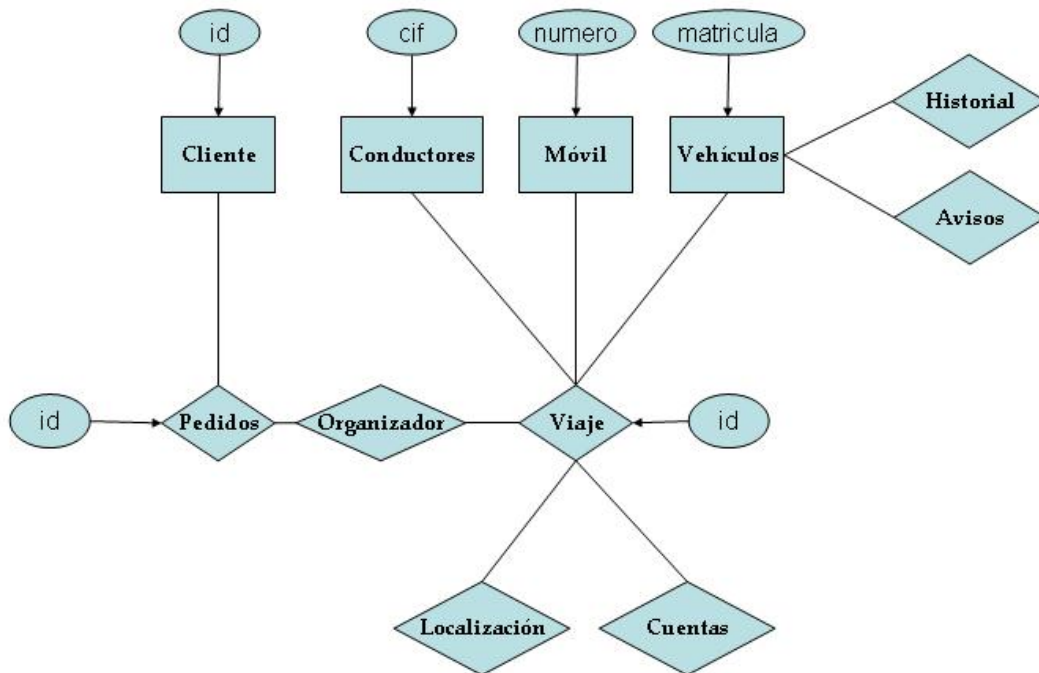


Figura 2.7 Esquema de la base de datos.

Leyenda:

- Elipses: Campos
- Rectángulos: Tablas que no incorporan campos externos.
- Rombos: Tablas que incorporan campos de procedencia externa

Los campos resaltados son los que forman parte en las relaciones entre tablas, así por ejemplo, la tabla organizador, esta compuesta por identificadores de 'Viaje' y de 'Pedidos'; la tabla 'Viaje' integra los campos: 'cif' -> 'conductores', 'numero' -> 'movil', 'matricula' -> 'vehiculos'. Estos campos, identifican además cada registro como un dato único, de manera que no pueda haber dos registros en la misma tabla con el mismo valor, en este caso MySQL impedirá que se realice una operación que viole este principio.

Aparte de estos campos, cada tabla posee otros propios dependiendo del tipo de información que se maneje. A continuación se detallaran los campos de una tabla en concreto pero no se indicarán todos los campos creados puesto que la flexibilidad y escalabilidad del programa con respecto a este apartado es una propiedad que se explicará mas adelante en el siguiente apartado: "Particularidades de la implementación en Flash con respecto a la base de datos".

Se mostrarán a continuación como ejemplo los campos que forman parte de la tabla pedidos de forma esquemática:

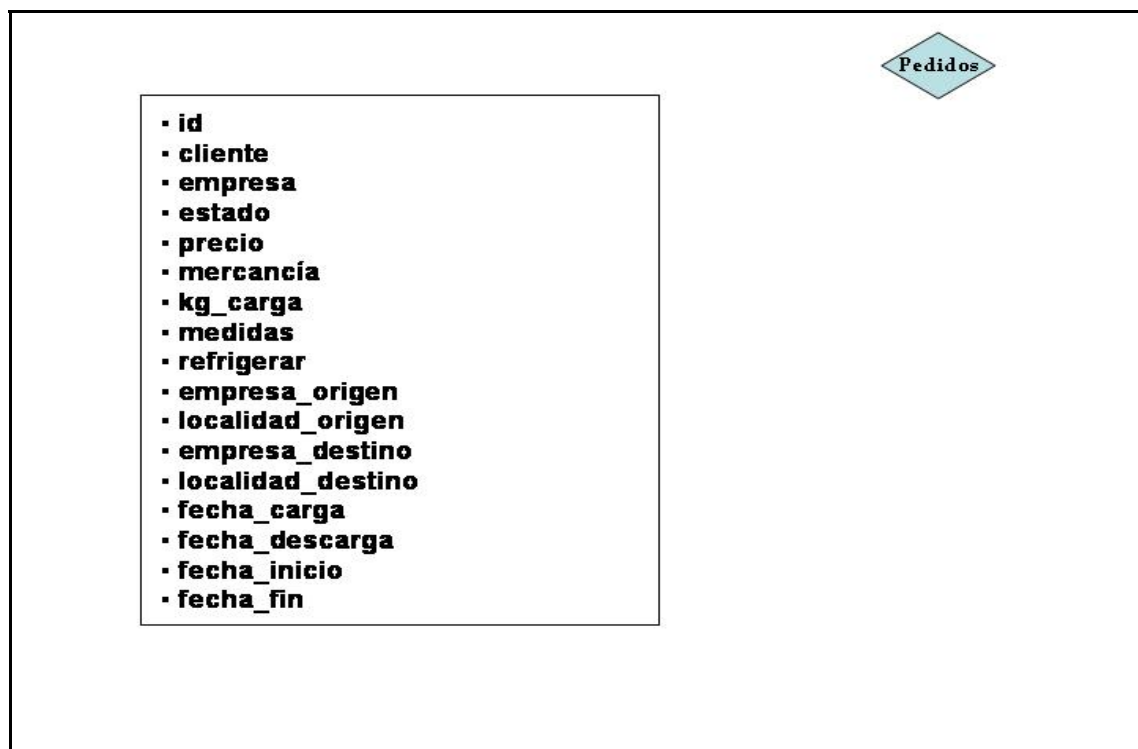


Figura 2.8 Estructura ejemplo de la tabla 'Pedidos'.

En este caso el campo 'id' tiene la particularidad de contar con la característica 'autoincrement', para ello es necesario como norma en MySQL que sea un campo con el atributo "PRIMARY", esto identifica a un registro en la tabla pedidos como elemento único en la tabla. El resto de campos almacenan datos del tipo 'int', 'varchar' o 'date'.

### 2.2.1. Particularidades de la implementación en Flash con respecto a la base de datos

Para el correcto funcionamiento de la aplicación desarrollada se deben tener una serie de campos característicos que forman parte de la estructura de la base de datos utilizada. A continuación se indicarán aquellos atributos necesarios, especificando la función especial de cada uno de ellos:

- Las tablas 'cliente', 'conductor', 'móvil' y 'vehículos', aparte de contener un atributo con la característica "UNIQUE" que asegure la unicidad de cada registro, deberá contar con un atributo denominado 'id' con la propiedad de "auto\_increment", para no violar las reglas establecidas en MySQL se debe marcar la propiedad "PRIMARY" para este campo.
- Con respecto a la tabla 'conductor' debe integrar el campo 'movil\_empresa' para que a la hora de crear nuevos viajes vayan asociados los valores conductor - móvil de empresa.
- En referencia a 'Historial' y 'Avisos', deberán incluir el campo 'matricula', ya que este es el campo relacional utilizado para asociar cada registro de estas tablas con cada uno de los vehículos. Sin embargo no hay que marcar la propiedad "UNIQUE" puesto que en este caso si queremos tener la posibilidad de introducir diferentes registros asociados al mismo vehículo.
- En cuanto a 'Pedidos' y 'Viajes', ambos deben contener el atributo 'id' que será el que identifique de manera unívoca a cada elemento de la tabla. En este caso, y a diferencia del anteriormente indicado, no existirá otro campo con la propiedad "UNIQUE", aunque si fuera necesario se podría establecer sin que afectara en teoría al funcionamiento del programa. Además del campo 'id', estas tablas deben incluir el campo "estado", "fecha\_inicio", y "fecha\_fin" que especifican:
  - Estado: estado en que se encuentra el pedido/viaje: espera, en curso, realizado...
  - Fecha inicio: fecha en la que se inicia el pedido/viaje
  - Fecha\_fin: fecha en la que finalizó el pedido/viaje
- En la tabla 'Viajes' hay que incluir también el campo 'localiza', que especifica si un viaje está siendo localizado, así como el intervalo de tiempo que seguirá el algoritmo de localización.

- En “Cuentas”, es necesario incluir los atributos:
  - ‘id\_viaje’: en donde almacena de forma única el identificador de un viaje.
  - ‘ingresos’ / ‘gastos’ / ‘rentabilidad’: para almacenar los ingresos/gastos/rentabilidades relacionados/as con el viaje.
- ‘Localiza’ debe tener el campo ‘id\_viaje’ y ‘movil’ que hacen referencia al viaje localizado, así como el campo ‘id’ con la propiedad “auto\_increment” y “PRIMARY” para que se registren las diferentes coordenadas de posición de manera secuencial según el tiempo en que se produzcan.
- La tabla ‘organizador’ estará compuesta únicamente por los campos ‘id\_viaje’ e ‘id\_pedido’, con lo que se asocian uno o más pedidos a un solo viaje.

**Nota:**

La aplicación diseñada registra valores automáticamente para la mayoría de estos parámetros, siendo transparente al usuario.

Además de estos parámetros característicos de la base de datos que garantice el funcionamiento y automatización de tareas de la aplicación desarrollada, es necesario introducir tres nuevas tablas que atenderán al uso de parámetros de configuración del programa. La funcionalidad que estas desempeñan son:

- ‘conf\_vista’: Esta tabla almacena una serie de datos de configuración en relación a las demás tablas. Cada campo lleva asociado un “nombre a mostrar” con el que se representará en el programa. Además, a la hora de previsualizar registros no se mostrarán todos los datos, puesto que no todos los campos representan información con la misma relevancia, así pues, en ‘conf\_vista’ se identifican aquellos campos que serán mostrados en la previsualización.
- ‘conf\_perfiles’: Aquí se almacena la información referente a los usuarios y los passwords correspondientes, asignados por el administrador.

A continuación se detalla esquemáticamente la estructura de estas tablas indicando el nombre de las mismas, así como los campos que han de ser implementados:

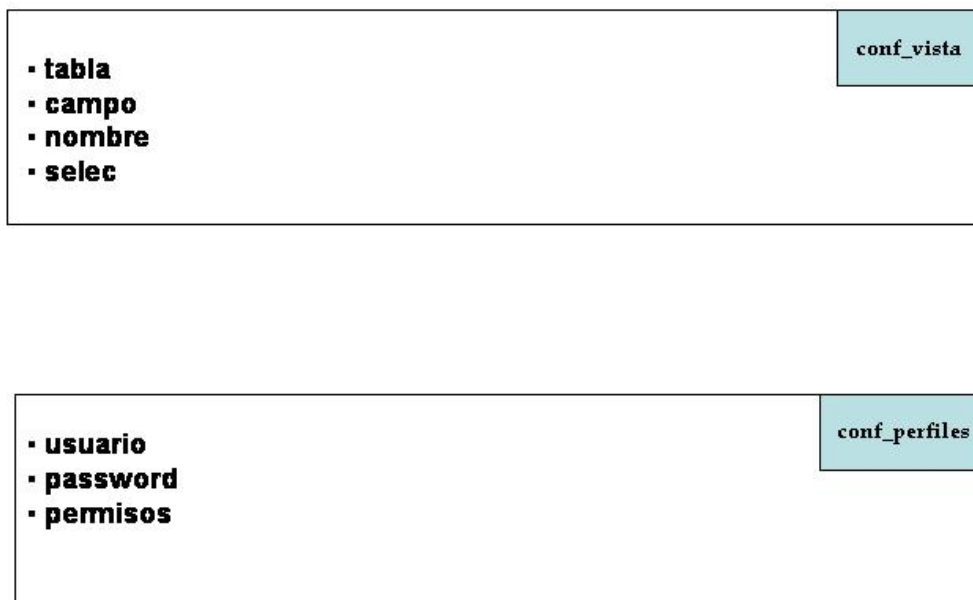


Figura 2.9 Esquema de 'conf\_vista' y 'conf\_perfiles'.

- La última necesaria en la estructura es la de 'gastos', con un único campo 'tipo', en donde se almacenan aquellos parámetros que forman parte de los gastos. Esto es utilizado para calcular los gastos totales de cada viaje y el cálculo de la rentabilidad.

Por último, cabe destacar que en la base de datos deben configurarse ciertos parámetros que dependerán de cada PYME, tales como usuario y password, permisos de cada usuario en relación a las funciones permitidas en la base de datos, conexiones permitidas.

En este proyecto se ha instalado el servidor de la base de datos en el mismo servidor Web Apache, por lo que la base de datos se ha configurado para permitir las conexiones del 'localhost', a un usuario con su password establecido y los permisos necesarios.

Con todo esto, ya se obtiene un correcto funcionamiento de la aplicación que ha sido desarrollado para permitir la máxima escalabilidad posible, ya que el programa interpreta el número de campos con los que trabajar de forma dinámica. Es decir, si tras un tiempo la PYME decide incluir un nuevo campo para almacenar nuevos valores en la base de datos, solo es necesario configurarlos mediante la herramienta proporcionada, en donde se le



asignará un nombre, y se activará o desactivará la opción de previsualización. No obstante, ésta y otras funcionalidades del programa estarán completamente detalladas en su manual correspondiente.

Tras esto el programa se adaptará al cambio sin que afecte a su rendimiento, mostrando tanto los datos registrados en el pasado, como valores introducidos en el nuevo atributo. Con ello conseguimos un programa flexible y con capacidad de adaptación según las necesidades de cada PYME.

## 2.3. Integración del sistema de localización

Uno de los requisitos para este proyecto era implementar un servicio de localización de flotas para contribuir al control de los vehículos. En la ‘solución básica’ se eligió la tecnología GSM a integrar en la aplicación desarrollada. Como se indicó en el primer apartado, son operadoras de telefonía móvil, Movistar y Vodafone, quienes ofrecen dicho servicio. Vodafone ofrece una API de Java, para implementar en un programa de diseño propio. Se usó esta herramienta para realizar un programa en Java, compatible con el sistema creado. La arquitectura básica de este servicio está detallada en el anexo correspondiente. Ésta y más información sobre la API está detallada en un documento pdf facilitado en [5].

El sistema de funcionamiento del servicio ofrecido por Vodafone se basa en la aplicación ‘Plug InRed’. La funcionalidad de aplicación se resume a continuación de forma breve junto con un esquema gráfico de comunicación:

*“Gracias al Plug InRed de Vodafone usted como desarrollador podrá añadir valor a sus aplicaciones de PC añadiéndoles la capacidad de comunicarse con el mundo de la telefonía móvil, de forma que amplíe y mejore su propia oferta de servicios.*

*Sus aplicaciones podrán adaptarse a un gran catálogo de servicios de red móvil accesibles desde un ordenador conectado a Internet:*

- *Envío y recepción de SMS tanto a móviles Vodafone como a otros operadores.*
- *Localización de terminales Vodafone con coordenadas (x,y), mapas y rutas.*
- *Envío de mensaje multimedia (MMS) a móviles Vodafone y otros operadores.*
- *Envío de mensajes WAP-Push.”*

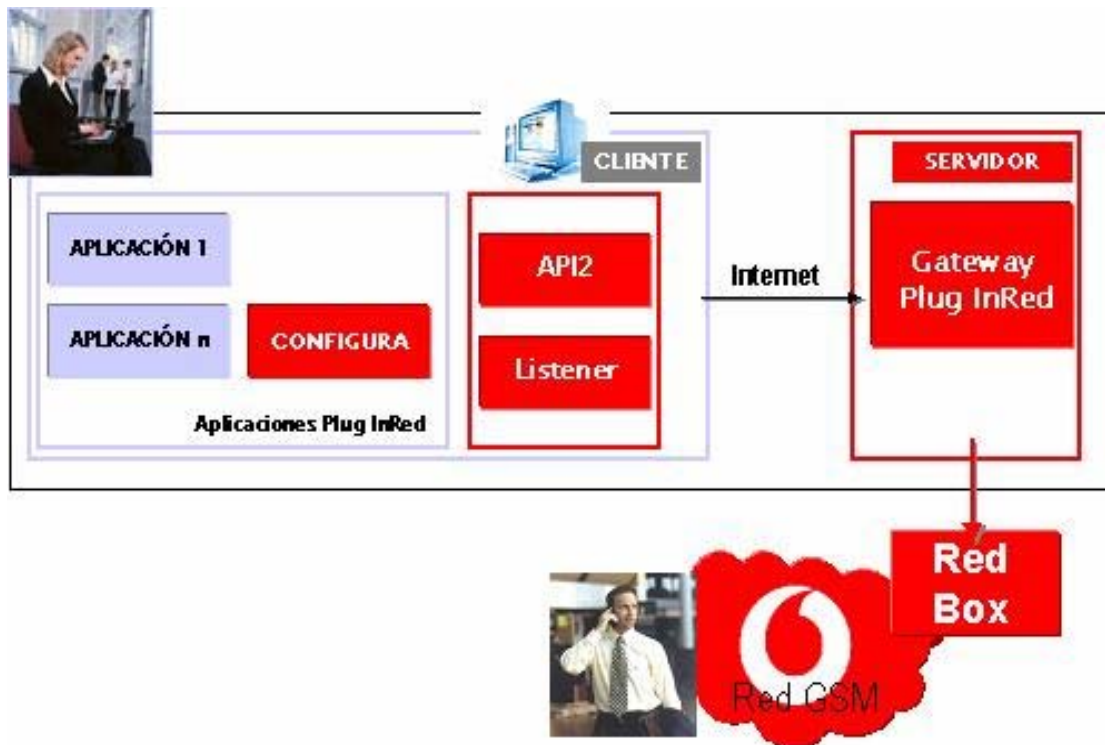


Figura 2.10 Esquema de comunicación de Plug InRed.

Información recogida de [5].

Por tanto, se cuenta con Vodafone como proveedor del sistema de localización. Es necesario ahora, para la integración del servicio, desarrollar un programa en java que incluya las funciones del API de Vodafone. Surge aquí la cuestión de ¿Cómo conectar la tecnología implementada hasta ahora, con este nuevo anexo?

Puesto que se deben almacenar las coordenadas de los vehículos de cada una de las llamadas realizadas al sistema de localización, que mejor que hacerlo en la base de datos, cuya estructura ya ha sido implementada pensando en esta problemática. Por lo tanto MySQL es la interfaz que permite la integración del sistema localización en la aplicación diseñada para gestionar la información de la base de datos.

Se muestra a continuación un esquema gráfico con la comunicación entre los módulos actuales:

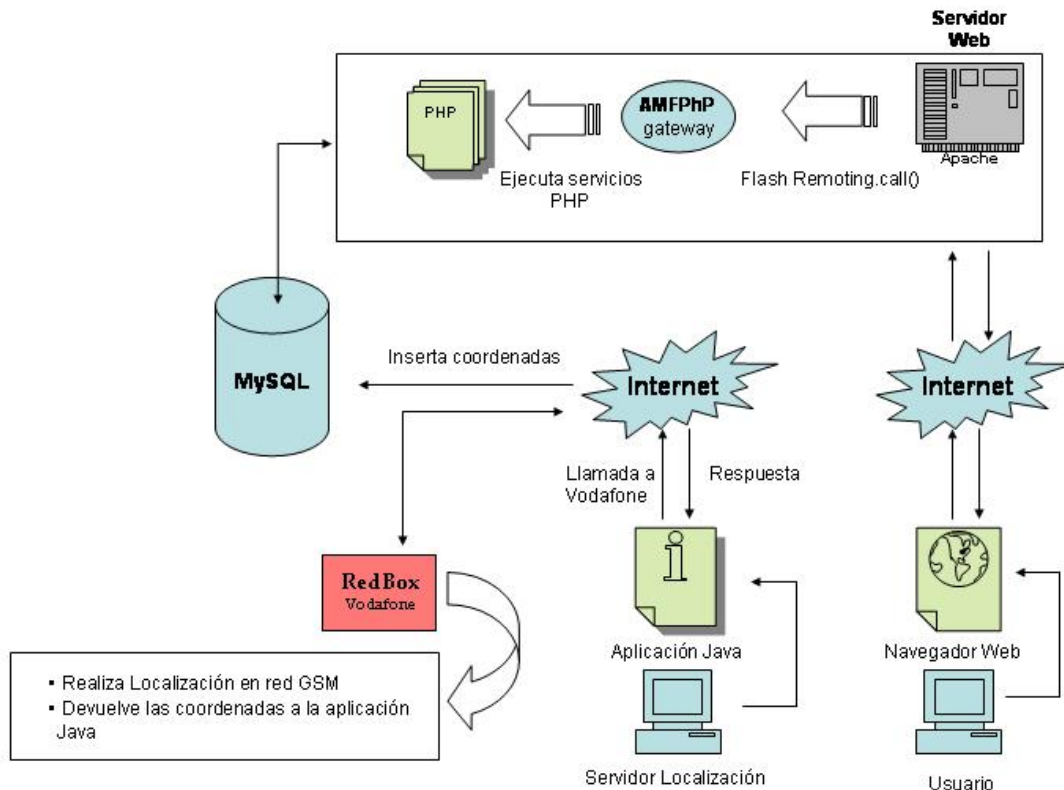


Figura 2.11 Esquema de comunicación con sistema de localización implementado.

Se necesita que la aplicación java sea capaz de realizar llamadas a las funciones proporcionadas en la API de Vodafone, y de insertar en la base de datos las coordenadas recogidas.

La implementación de la API se puede realizar sin problemas aparentes, puesto que Vodafone facilita el código java necesario para realizar las funcionalidades básicas como la de autenticación y localización. Para poder usar el servicio de localización es necesaria una autenticación previa.

Solo quedaría por implementar la forma de comunicar java con MySQL. Esta es una problemática conocida que promete resolver la tecnología JDBC. Esta es una breve descripción hallada en [6]:

*“JDBC es el acrónimo de Java Database Connectivity, un API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java*

*independientemente del sistema de operación donde se ejecute o de la base de datos a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice.*

*El API JDBC se presenta como una colección de interfaces Java y métodos de gestión de manejadores de conexión hacia cada modelo específico de base de datos. Un manejador de conexiones hacia un modelo de base de datos en particular es un conjunto de clases que implementan las interfaces Java y que utilizan los métodos de registro para declarar los tipos de localizadores a base de datos (URL) que pueden manejar.*

*Para utilizar una base de datos particular, el usuario ejecuta su programa junto con la librería de conexión apropiada al modelo de su base de datos, y accede a ella estableciendo una conexión, para ello provee en localizador a la base de datos y los parámetros de conexión específicos. A partir de allí puede realizar con cualquier tipo de tareas con la base de datos a las que tenga permiso: consultas, actualizaciones, creado modificado y borrado de tablas, ejecución de procedimientos almacenados en la base de datos, etc.”*

La forma en que actúa esta aplicación depende del parámetro ‘localiza’ implementado en la estructura de la base de datos, de esta forma el usuario podrá establecer el periodo de tiempo en el que se produzca la localización de los vehículos seleccionados con el programa. De esta forma, la aplicación java únicamente debe leer estos periodos o ‘timers’ que marcarán el intervalo de tiempo entre las llamadas al servicio de localización. Actúa pues como un “servidor de localización”, que trabaja de forma independiente al servidor Web, es decir pueden correr en la misma maquina o en maquinas diferentes.

Ahora sólo queda saber como se interpretan los resultados obtenidos. Se necesita representar cartográficamente las coordenadas de cada vehículo. Para esto se pensó en un primer momento en crear un método visual apoyado en las ventajas gráficas de usar la tecnología Flash. Sin embargo, resultó de gran ayuda la nueva aplicación de la compañía Google, conocida como ‘Google Earth’, que permite “navegar” visualmente por todo el globo terráqueo construido mediante imágenes de satélite, alcanzando cotas de detalle sorprendentes.

Tras unas pruebas se comprobó que se podían localizar puntos especificados según las coordenadas de longitud y latitud, las mismas coordenadas proporcionadas por la función de Vodafone. Solo quedaba resolver como representar un conjunto de puntos puesto que es interesante conocer, no solo la situación actual de los vehículos, sino las rutas que van describiendo. Esto es posible en ‘Google Earth’, puesto que procesa un tipo de archivos llamados ‘KML’.

KML ha sido desarrollado por Google y es una notación de marcas basada en XML cuyo contenido especifica varios parámetros de visualización en Google Earth, tales como el número de puntos que componen la ruta; la altura y color del polígono formado, el punto inicial a visualizar..., etc. La manera de configurar todas estas opciones está definida en el manual de KML proporcionado por Google.

Mediante PHP podemos generar la estructura KML, en la que introducir las coordenadas almacenadas de cada vehículo y así generar el archivo correspondiente que

recibirá el usuario en su navegador. Sin embargo en la práctica se observa que al ser KML una estructura tipo XML, el navegador lo interpreta como tal, y muestra por pantalla texto, clasificado como un lenguaje de marcas tipo HTML. La solución a esto se encuentra en el manual de XML donde se especifican los Mime-Types necesarios para que Google Earth ejecute de manera automática los archivos KML. Estos Mime-Types han de implementarse en la configuración del servidor Web.

La representación gráfica de la comunicación entre módulos para proporcionar el servicio de localización quedaría:

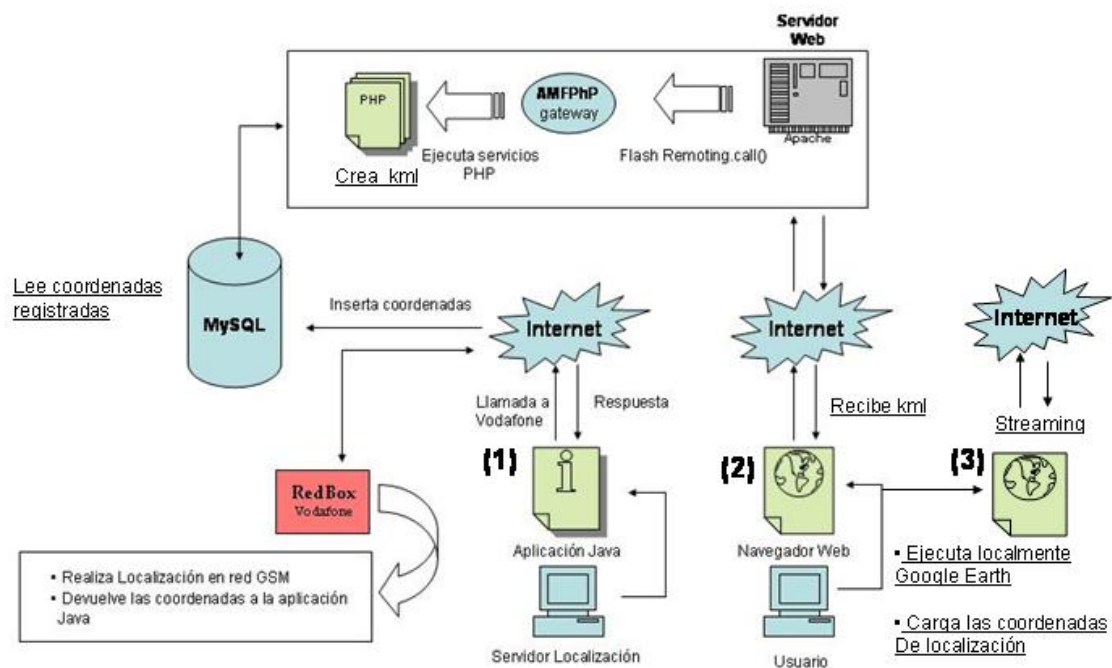


Figura 2.12 Esquema de comunicación con interpretación de coordenadas.

## Capítulo 3

# Manual de usuario

### 3.1. Introducción

Este software permite el almacenamiento y gestión de la información necesaria para el control de su empresa de logística. Además integra un servicio de localización de flotas con el que podrá visualizar la posición de sus vehículos así como las rutas que realizan en cada trayecto.

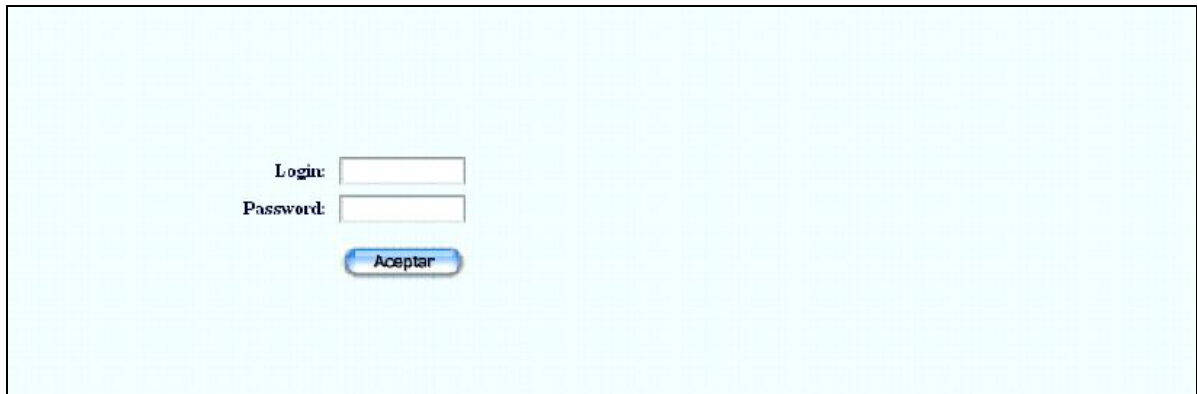
Para comenzar a utilizar este programa únicamente abra una nueva ventana de su navegador de Internet preferido e introduzca en la barra de direcciones la dirección correspondiente, proporcionada por el administrador.

**Advertencia:** Es necesario tener instalada la aplicación Macromedia Flash Player en su navegador para el correcto funcionamiento de este software.

A continuación se explicarán las distintas funcionalidades que ofrece el programa.

## 3.2. Acceso

Para poder acceder a las funciones principales del programa es necesario autenticarse como usuario registrado.



El formulario de acceso se muestra en un recuadro con fondo de cuadrícula. Incluye los siguientes elementos:

- Etiqueta "Login:" seguida de un campo de entrada de texto.
- Etiqueta "Password:" seguida de un campo de entrada de texto.
- Un botón de "Aceptar" con un efecto de sombra y un borde azul.

Figura 3.1 Acceso

- Login: debe introducir su nombre de usuario.
- Password: debe introducir su password.

Haga clic en aceptar o presione la tecla "Enter" para continuar.

**Advertencia:** en caso de desconocer esta información, o disponer de "login o password incorrectos"; por favor, póngase en contacto con el administrador.

### 3.3. Menú Inicio

Tras haber introducido el login y password correctamente, pasará directamente al menú “INICIO”. Desde aquí podrá acceder a las diferentes funciones del programa.



Figura 3.2 Menú inicio.

Para pasar al apartado deseado, haga clic en el botón correspondiente

El botón “Avisos” aparece únicamente cuando hay eventos programados que finalizaran dentro de 15 días o menos.

#### Ejemplo

- *Clientes: pasa al apartado de clientes.*
- *Pedidos: pasa al menú de pedidos.*



### 3.4. Gestión de clientes, conductores y móviles.

En esta sección usted podrá realizar las funciones necesarias para la gestión de la información privada de su empresa.

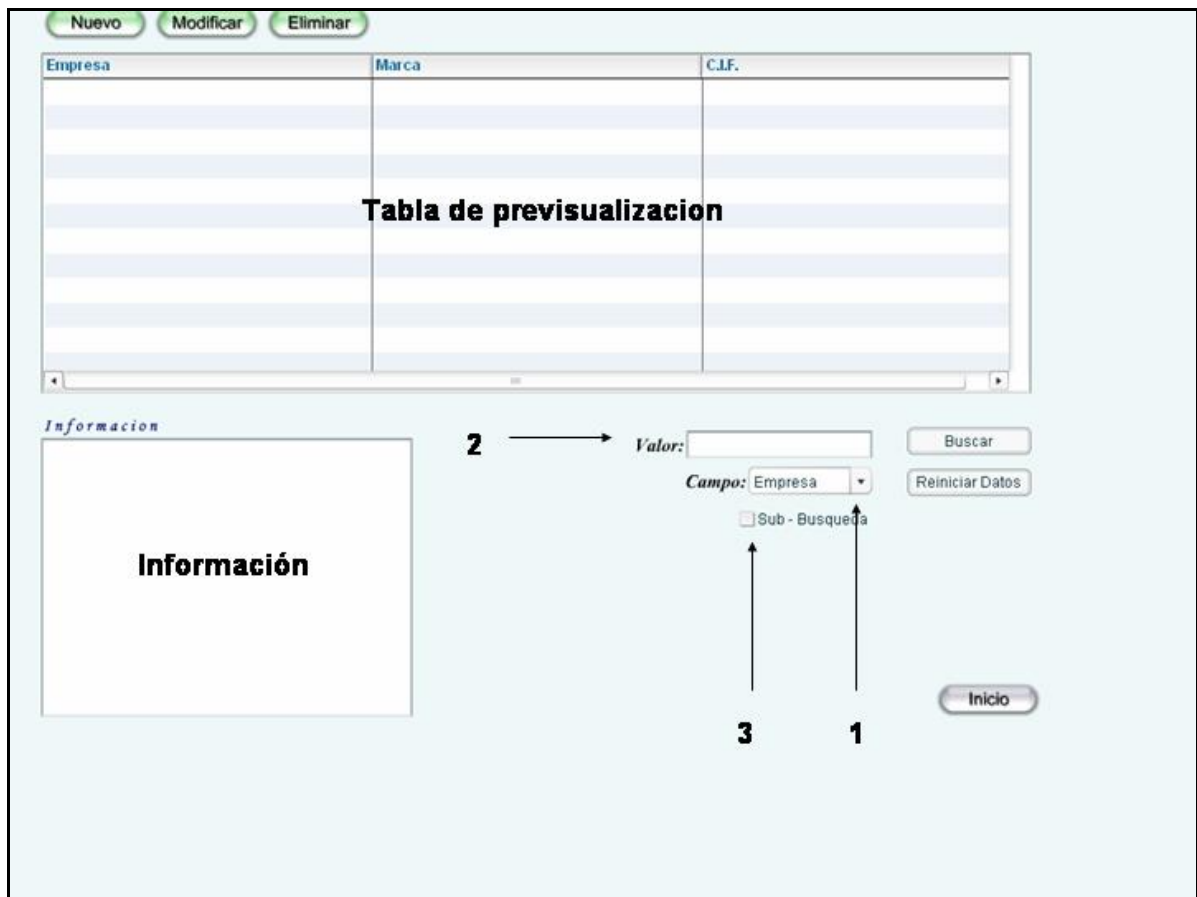


Figura 3.3 Gestión de datos: clientes, conductores y móviles.

Aquí podrá visualizar, almacenar, modificar y eliminar los datos que precise.

#### **FUNCIONES COMUNES PARA CLIENTES, CONDUCTORES Y MOVILES:**

- Usted podrá buscar el grupo de datos que más le interese. Para ello especifique un patrón de búsqueda pinchando en el (1) y escriba el valor correspondiente en (2). A continuación haga clic en “Buscar”
- Tras realizar una búsqueda aparecerán dos nueva opciones. Estas son, el botón “Reiniciar Datos” y la casilla de “Sub-Búsqueda” (3). Marque esta casilla cuando quiera realizar una segunda búsqueda sobre los resultados obtenidos en

la primera. Pulse el botón “Reiniciar Datos” para resetear los datos previsualizados.

- Una ventana de advertencia le avisará en caso de que la búsqueda no produzca resultados.
- En la “Tabla de previsualización” usted podrá ver ciertos parámetros de los registros deseados, configurados por el administrador. Si desea obtener toda la información referente a un registro específico, seleccione en la tabla dicho elemento y haga clic la fila iluminada. Se mostrará el resultado de esta operación en el cuadro “Información”.
- Para almacenar un nuevo registro haga clic en “Nuevo”.
- Para modificar / eliminar un registro deber seleccionar previamente un dato en la “tabla de previsualización”. A continuación haga clic en “Modificar” / “Eliminar”.
- Para regresar al menú “INICIO” haga clic en el botón “Inicio”.

#### **PARTICULARIDAD CLIENTES:**

- Al acceder al apartado “Clientes” la tabla de previsualización no mostrará ningún registro. Debe realizar una búsqueda para ver la información deseada.

#### **PARTICULARIDAD CONDUCTOR, MOVILES:**

- Al acceder a estos apartados se mostrarán por defecto toda la información almacenada correspondiente.

### 3.5. Vehículos

Este apartado presenta la misma funcionalidad que la descrita para “Conductores” y “Móviles” pero añade una serie de opciones que serán explicadas a continuación.

Modelo	Tipo	Matricula	Marca

*Informacion*

Valor:

Campo:

Sub - Busqueda

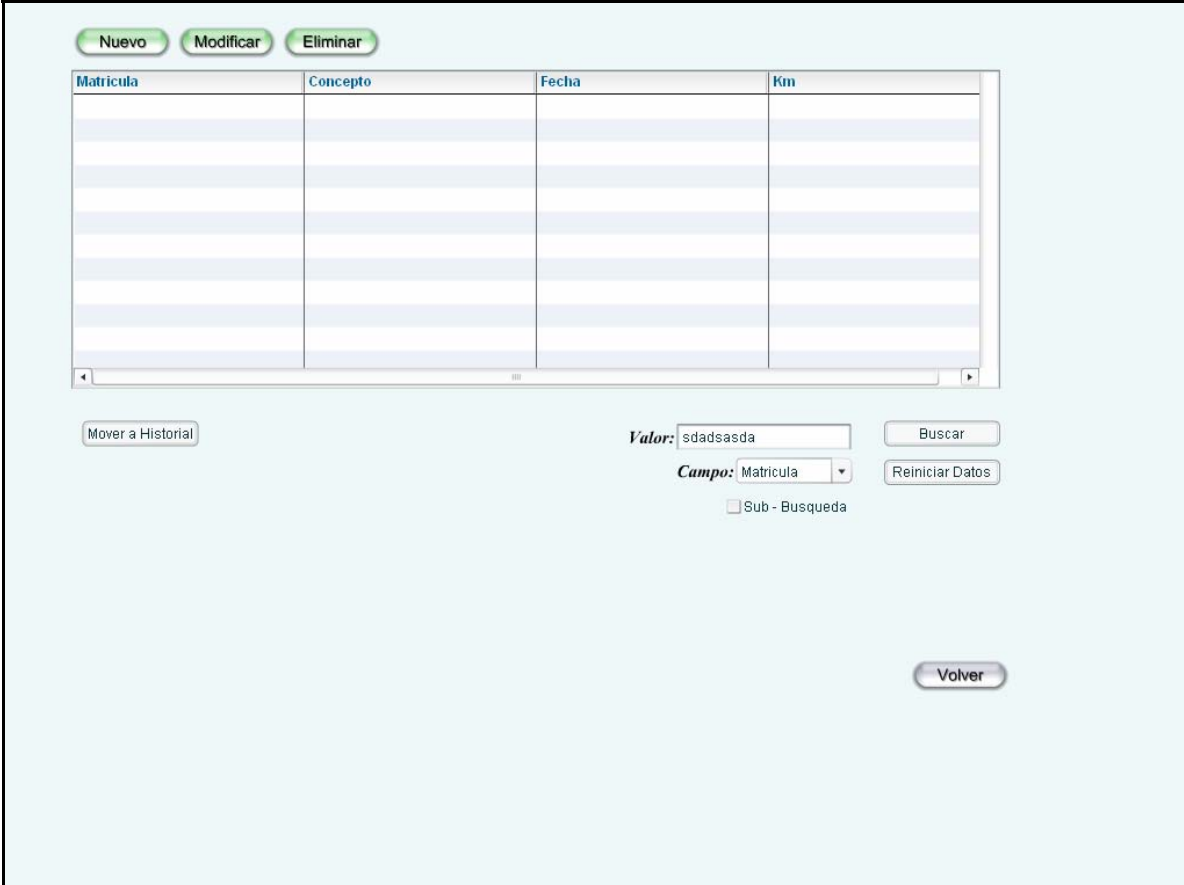
Figura 3.4 Gestión de vehículos.

- Para acceder al historial del vehículo haga clic en “Historial”.
- Para acceder a los avisos del vehículo haga clic en “Avisos”.
- Para acceder a la rentabilidad del vehículo haga clic en “Rentabilidad”.

**Advertencia:** Debe haber seleccionado un dato en la tabla para poder acceder a estas opciones.

### 3.5.1. Historial y avisos

Aquí podrá registrar sucesos que formen parte del historial del vehículo seleccionado o programar avisos para que el programa le avise de eventos futuros programados. El sistema de uso es el mismo que el descrito para “Conductor”, “Móvil” o “Vehículos”, con la única particularidad de que en este caso no existe el cuadro “Información” puesto que todos los parámetros a ofrecer están incluidos en la “Tabla de Previsualización”.



The screenshot displays a web interface for managing vehicle alerts and history. At the top, there are three buttons: "Nuevo", "Modificar", and "Eliminar". Below these is a table with four columns: "Matricula", "Concepto", "Fecha", and "Km". The table is currently empty. Below the table, there is a "Mover a Historial" button. To the right, there is a search section with a "Valor:" input field containing "sdadsasda", a "Buscar" button, a "Campo:" dropdown menu set to "Matricula", and a "Reiniciar Datos" button. Below the search section is a checkbox labeled "Sub - Búsqueda". At the bottom right, there is a "Volver" button.

Figura 3.5 Gestión de avisos e historial de los vehículos.

#### **PARTICULARIDAD AVISOS:**

- Esta opción incorpora el botón “Mover a Historial” que permite automáticamente trasladar un evento ‘aviso’ al apartado “Historial”. Seleccione el dato que desee incluir en el historial del vehículo y haga clic en este botón.

### 3.5.2. Rentabilidad

En esta opción podrá visualizar la rentabilidad generada por el vehículo seleccionado a lo largo del año.

- Únicamente debe especificar un año en el selector, y hacer clic en “Dibujar Gráfica”.
- Para regresar al apartado anterior haga clic en “Cancelar”.

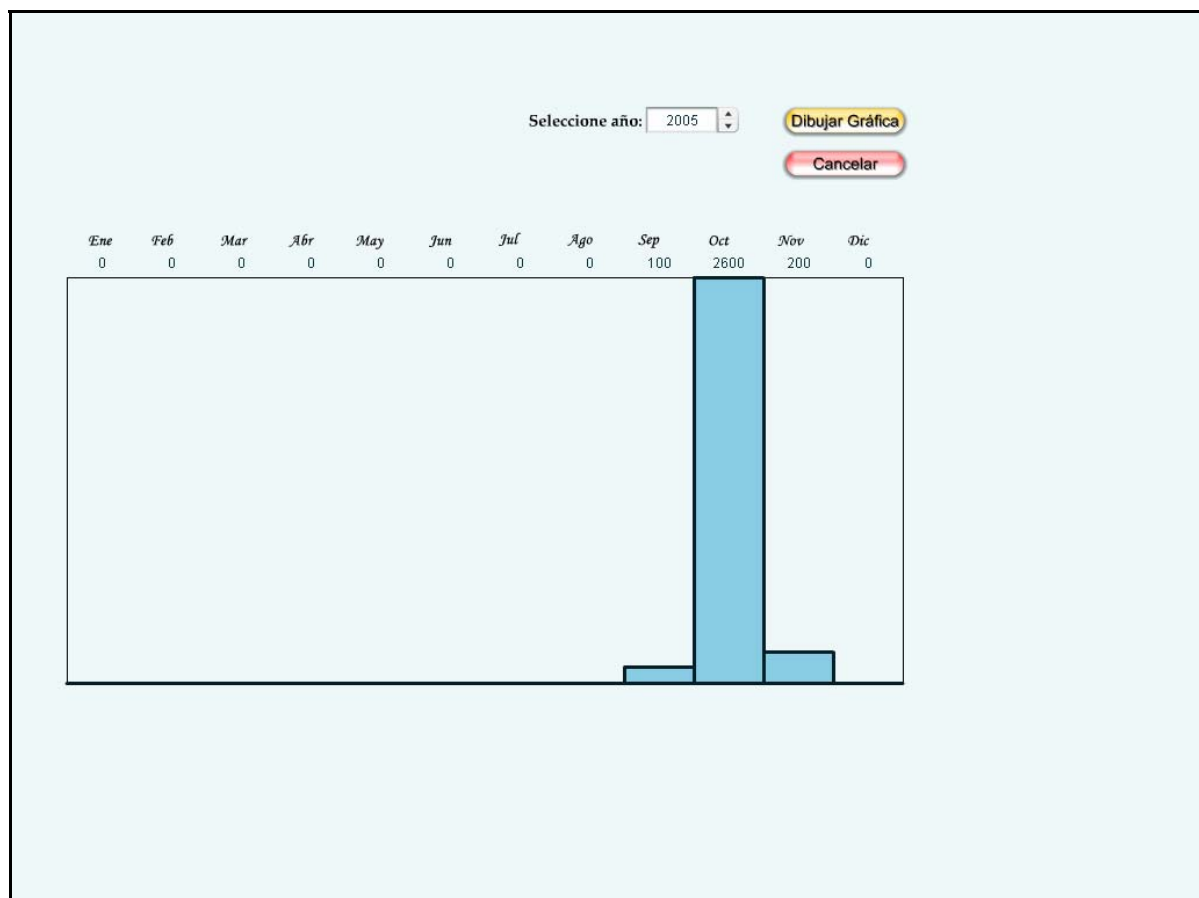


Figura 3.6 Visualización de las rentabilidades.

### 3.6. Inserción y modificación de registros

En este apartado se realiza la inserción o modificación de registros. Los distintos parámetros están referenciados en la columna “Campos”, únicamente debe introducir en la columna “Valor” y la fila correspondiente, el valor deseado.

Una vez rellenados los parámetros, haga clic en “Aceptar” para almacenar la información. En caso contrario, pulse “Cancelar” para volver.

Ejemplo de nuevo conductor:

Campo	Valor
Nombre	
Apellidos	
CIF	
Direccion	
Localidad	
Provincia	
CP	
Telefono Fijo	
Telefono Movil	
movil_empresa	
Carnet	
fechanac	
observaciones	

Figura 3.7 Inserción y modificación de registros.

Para el caso de modificar registros, el sistema de funcionamiento es el mismo, excepto que en la columna “Valor” se cargará la información correspondiente del registro a modificar.

**Advertencia:** a la hora de introducir una fecha debe cumplir este formato

(YYYY-MM-DD): año, mes, día.

### 3.7. Eliminación de registros

Una vez seleccionado el dato que desea borrar, se le pedirá una confirmación. En caso de querer proceder a eliminarlo haga clic en “Aceptar” en caso contrario pulse “Cancelar”.

Ejemplo de eliminar un móvil:

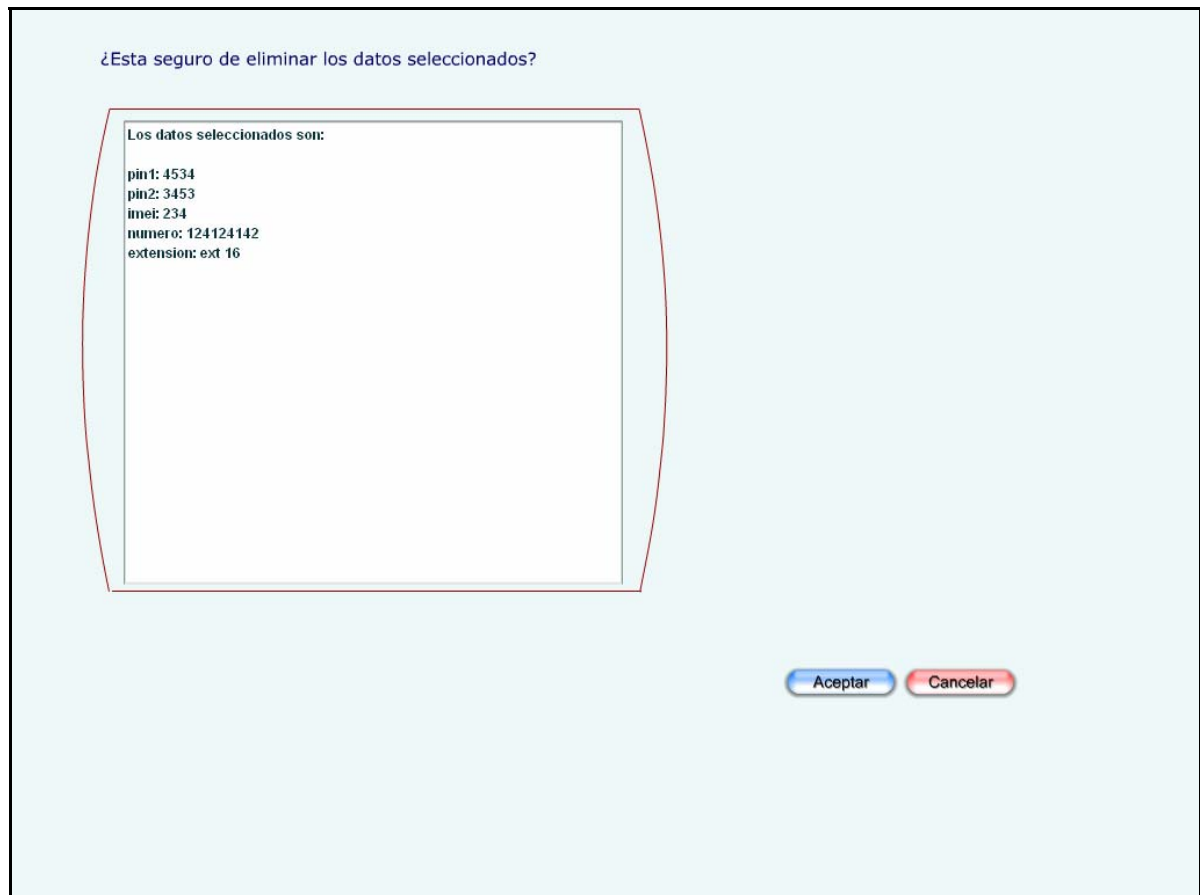


Figura 3.8 Eliminación de registros.

### 3.8. Pedidos

Puede realizar un nuevo pedido, ver los pedidos pendientes, o aquellos que han sido realizados. Para pasar al apartado deseado, haga clic en el botón correspondiente

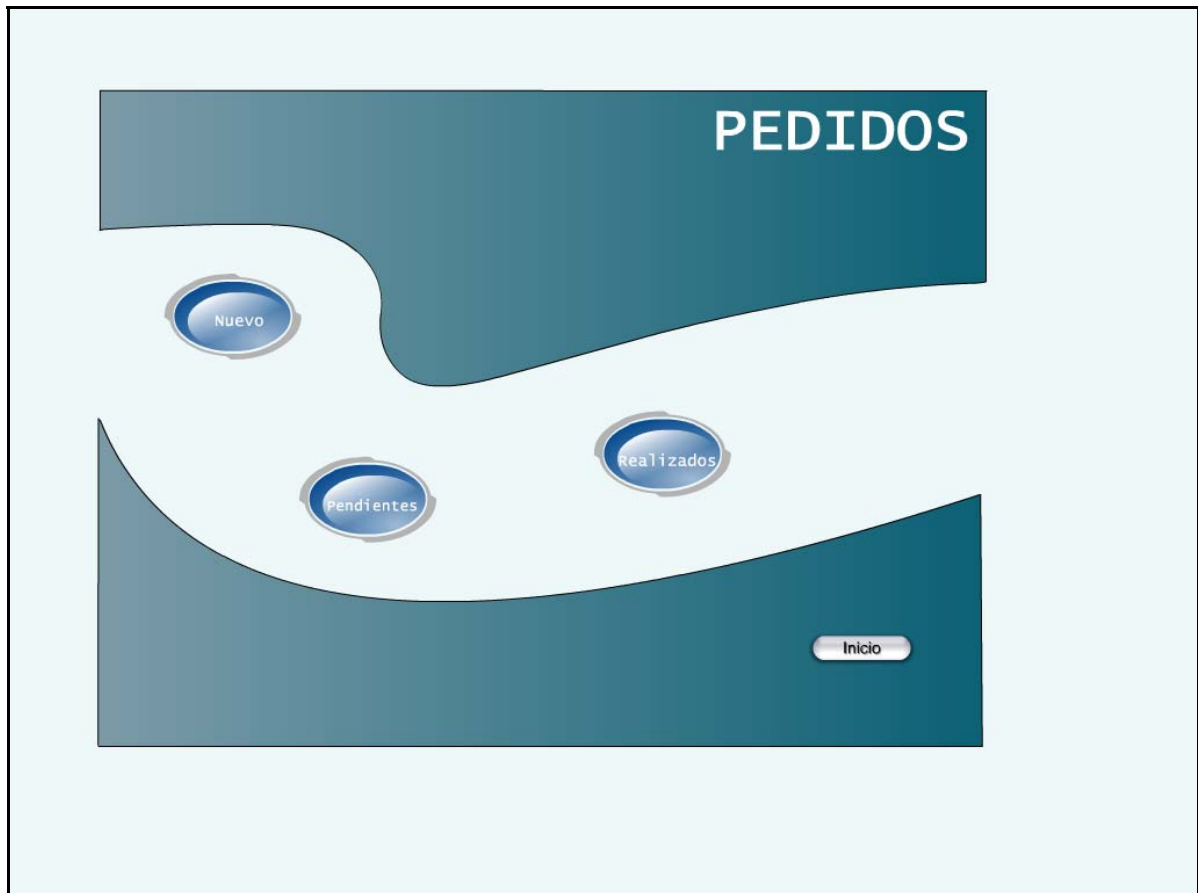


Figura 3.9 Menú Pedidos.

Para regresar al menú "INICIO" haga clic en el botón "Inicio".



### 3.8.1. Nuevo pedido

En esta sección podrá organizar un nuevo pedido. Para ello debe buscar y seleccionar el cliente que realiza el pedido, e introducir los valores de los parámetros especificados en la columna “Campos”. El sistema de búsqueda es el mismo que el se detalló en los apartados anteriores.

Empresa, Teléfono y CIF hacen referencia a parámetros que diferencian de manera única cada empresa, estos se representan únicamente de forma informativa.

Empresa	Marca	C.I.F.

Valor:

Campo: Empresa

Sub - Búsqueda

Campo	Valor
Precio	
Mercancia	
Kg Carga	
Medidas Especiales	
Refrigeracion	
Origen Empresa	
Origen Localidad	
Destino Empresa	
Destino Localidad	
Fecha Carga	
Fecha Descarga	

Empresa

Telefono

cif

Figura 3.10 Gestión de un nuevo pedido.

Una vez esté seguro de querer organizar y almacenar el pedido presione “Aceptar”, en caso contrario presione “Cancelar”.

El pedido quedará en estado “pendiente”. Por lo que podrá gestionarlo en el apartado “Pedidos” -> “Pendientes”.

**Advertencia:** a la hora de introducir una fecha debe cumplir este formato

(YYYY-MM-DD): año, mes, día.

## 3.8.2. Pedidos Pendientes

Aquí podrá visualizar todos aquellos pedidos que aun no se han realizado.

The screenshot shows a web interface for managing pending orders. At the top left, there are two buttons: 'Modificar' and 'Eliminar'. The main title is 'Pedidos Pendientes'. Below it is a table with the following columns: 'id', 'Cliente', 'empresa', 'Precio', 'Mercancia', 'Kg Carga', and 'Medidas E'. The table is currently empty. Below the table, there is a section titled 'Información del cliente/pedido'. On the left, there are two buttons: 'Info. Pedido' and 'Info. Cliente'. On the right, there is a search section with a 'Valor:' input field, a 'Campo:' dropdown menu (set to 'Cliente'), and a 'Sub - Búsqueda' checkbox. There are also buttons for 'Buscar', 'Reiniciar Datos', and 'Volver'.

Figura 3.11 Gestión de los pedidos pendientes.

En la “Tabla de previsualización” aparecerán por defecto todos los pedidos que permanezcan pendientes. Sin embargo, usted podrá realizar búsquedas selectivas, con el mismo sistema de uso descrito en apartados anteriores.

Una vez seleccionado un pedido de la tabla, usted podrá ver en el cuadro de “Información” la información del pedido así como la del cliente asociado. Para ello únicamente pulse “Info. Pedido” o “Info. Cliente”.

Usted podrá modificar o eliminar un pedido pulsando el botón correspondiente, “Modificar” o “Eliminar”.

La forma de modificar un registro es la misma que el descrito en el apartado “Nuevo Pedido” sólo que únicamente esta vez se cargarán los datos asociados al pedido, de manera que usted podrá modificarlos, o bien seleccionando un cliente distinto, o modificando el valor de los parámetros en la columna “Valor” y fila correspondiente.

### 3.8.3. Pedidos Realizados

Aquí accederá a aquellos pedidos que se han realizado clasificados por la fecha de fin.

Modificar Eliminar

Seleccione Fecha: Año: 2006  
Cargar Mes: 1 Pedidos: Enero del 2006

id	Cliente	empresa	Precio	Mercancia	Kg Carga	Medidas E

Información del cliente/pedido

Info. Pedido Info. Cliente

Valor: Buscar

Campo: Cliente Reiniciar Datos

Sub - Busqueda

Volver

Figura 3.12 Gestión de los pedidos realizados.

El funcionamiento de este apartado es el mismo que el descrito para “Pedidos Pendientes”, salvo la opción de escoger fecha. Usted puede especificar el año y el mes para visualizar los pedidos realizados correspondientes, puesto que están organizados temporalmente. Una vez seleccionado la fecha, pulse el botón “Cargar” para realizar la acción.

### 3.9. Viajes

Puede organizar un nuevo viaje, ver los viajes pendientes, los que se encuentran en curso o aquellos que han sido realizados. El apartado de localización será detallado aparte. Para pasar al apartado deseado, haga clic en el botón correspondiente

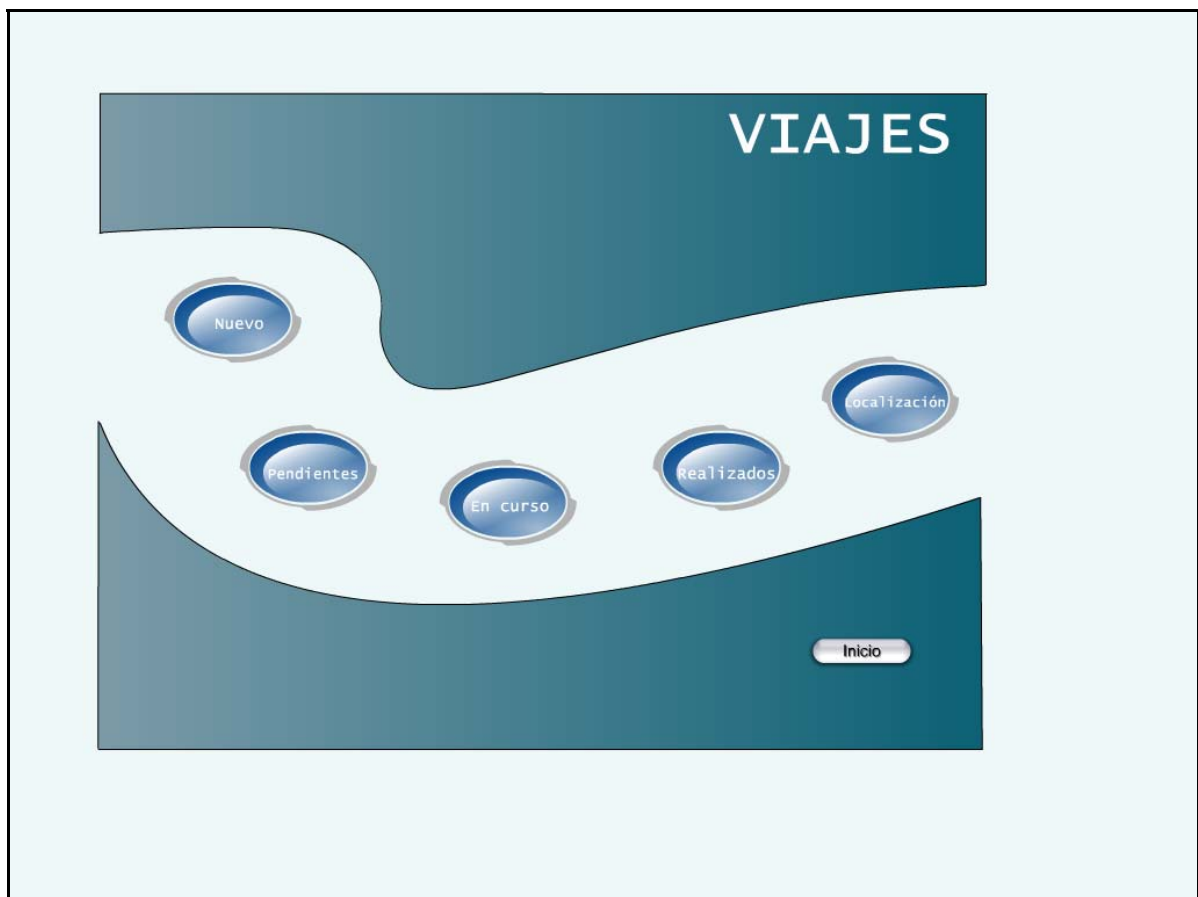


Figura 3.13 Menú de viajes.

Para regresar al menú "INICIO" haga clic en el botón "Inicio".

### 3.9.1. Nuevo viaje

En este apartado podrá organizar nuevos viajes. Un viaje está formado por un conductor, un móvil a localizar, un vehículo que actúe de cabeza y otro de remolque. Además cada viaje englobará uno o más pedidos según lo requiera.

Los viajes se organizan en dos pasos:

1. Debe seleccionar el conductor, móvil de la empresa y vehículos que lo realizarán.

pin1	imei	numero
4534	234	124124142
1231	123	346363
1254		668452368
111	0908	99999

**Datos Seleccionados:**

**Conductor**

Nombre:  Telefono:

Apellidos:  d.n.i.:

**Cabeza**

Modelo:  Matricula:

**Remolque**

Modelo:  Matricula:

Movil:

Informacion detallada

Figura 3.14 Primer paso en la creación de un nuevo viaje. Selección de requisitos.

Para la selección de estos elementos haga clic en el menú desplegable, y elija un tipo. A continuación se cargarán los datos en la “Tabla de visualización”. Seleccione el registro deseado haciendo clic en la fila correspondiente.

Cuando selecciona un conductor se cargará automáticamente el móvil de la empresa, sin embargo puede cambiarlo eligiendo “Móvil” en el menú desplegable y seleccionando un nuevo elemento.

Una vez elegido conductor, móvil, cabeza y remolque, si está seguro de continuar al paso 2, haga clic en “Aceptar”, en caso contrario haga clic en “Cancelar”.

2. En el segundo paso se asignarán los pedidos que se realizarán en el viaje.

Datos Seleccionados

id	Cliente	empresa	Precio	Mercancia	Kg Carga	Medidas E
30	3541354E	Mahor	1000			

Agregar Eliminar

id	Cliente	empresa	Precio	Mercancia	Kg Carga	Medidas E

Crear Cancelar

Figura 3.15 Segundo paso en la creación de un nuevo viaje. Elección de los pedidos.

En la tabla superior se cargarán aquellos viajes que estén pendientes por realizar.

Si desea recordar los elementos escogidos en el paso 1, haga clic en “Datos Seleccionados”.

Para agregar un pedido al viaje únicamente seleccione la fila correspondiente y a continuación haga clic en “Agregar”. En caso de querer desagregar un pedido agregado, selecciónelo en la tabla inferior y haga clic en “Eliminar”.

Una vez estén elegidos los pedidos deseados, haga clic en “Crear”, en caso de querer volver al menú de viajes, y cancelar la organización del viaje pulse “Cancelar”.

### 3.9.2. Viajes pendientes

Aquí visualizará los viajes pendientes.

**Eliminar**

**Viajes Pendientes**

id	Conductor	Cabeza	Remolque	Movil	fecha_inicio	fecha_fin
39	2303	mu 6669 DC	8897 DCF	637566644	2006-01-18	0000-00-0

**Iniciar Viaje**

**Pedidos**

id	Cliente	empresa	Precio	Mercancia	Kg Carga	Medidas E

**Ver Pedido**

**Volver**

Figura 3.16 Gestión de viajes pendientes.

En la tabla superior (amarillo) se cargan los viajes pendientes. Al seleccionar un viaje, pinchando en la fila correspondiente, se cargarán en la tabla inferior (naranja) los pedidos asociados. Usted podrá ver la información detallada de estos pedidos, seleccionando uno de ellos y pulsado el botón “Ver Pedido”.

Usted puede eliminar el viaje seleccionado, clicando en “Eliminar”.

Si desea iniciar el viaje seleccionado, haga clic en “Iniciar Viaje”. Con esto, pasará automáticamente a “Viajes en curso”.

Para volver al menú presione “Volver”.

### 3.9.3. Viajes en curso

En esta sección podrá gestionar los viajes que están en curso.

The screenshot displays a web interface for managing active trips. It features two main data tables and several control buttons.

**Viajes**

id	Conductor	Cabeza	Remolque	Movil	fecha_inicio	fecha_fin
25	yo	67	98	99889	2006-01-11	0000-00-0
42	234234	mu 6669 DC	8897 DCF	0	2006-01-19	0000-00-0

Below the table, there is a control area with a button labeled "Iniciar Localizacion", a numeric input field containing the value "0", and a button labeled "Terminar Viaje".

**Pedidos**

id	Cliente	empresa	Precio	Mercancia	Kg Carga	Medidas E
30	3541354E	Mahor	1000			

Below the table, there is a button labeled "Ver Pedido".

At the bottom right of the interface, there is a button labeled "Volver".

Figura 3.17 Gestión de viajes en curso.

Puede visualizar los viajes y los pedidos de la misma manera que en “Viajes Pendientes”.

Si desea finalizar un viaje seleccionado haga clic en “Terminar Viaje”. Al terminar un viaje termina también la localización del mismo.

Si desea iniciar la localización de un vehículo, seleccione el viaje correspondiente, y marque en el selector de valores, un número superior a 0. Este número indica el intervalo en horas en que se localizará un vehículo. Por ejemplo si usted quiere saber donde se encuentra su vehículo cada 3 horas, marque en esta casilla el valor ‘3’. El botón “Iniciar Localización” estará activo solo cuando el viaje seleccionado no haya iniciado anteriormente la localización.

Si desea regresar pulse “Volver”.



### 3.9.4. Viajes realizados

En esta sección visualizará los viajes finalizados que están agrupados por fechas.

The screenshot displays a web interface for managing completed trips. At the top, there is a date selection section with the text 'Seleccione Fecha:'. It includes a dropdown menu for the year set to '2005', a 'Cargar' button, a dropdown menu for the month set to '12', and the text 'Viajes: Diciembre del 2005'. Below this is a table of trips with the following data:

id	Conductor	Cabeza	Remolque	Movil	fecha_inicio	fecha_fin
12	2303	mu 6669 DC	8897 DCF	99999	0000-00-00	2005-12-1

Below the table are buttons for 'Ver Ruta' and 'Cuentas'. Underneath is a section for 'Pedidos' with a table of orders:

id	Cliente	empresa	Precio	Mercancia	Kg Carga	Medidas E
13			500	Melocotones		

Below the table are buttons for 'Ver Pedido' and 'Volver'.

Figura 3.18 Gestión de viajes realizados.

Por defecto se cargarán los viajes correspondientes a la fecha actual. Usted puede visualizar los viajes realizados en otra fecha seleccionando el mes y el año deseados, y a continuación pulsado “Cargar”. Puede visualizar los viajes y los pedidos de la misma manera que en “Viajes Pendientes”.

Puede ver la ruta de localización que realizó el vehículo correspondiente al viaje seleccionado, para ello haga clic en “Ver Ruta”. Este botón solo está activo en caso de que el viaje haya sido previamente localizado. Recibirá un archivo .kml que deberá abrir, para iniciar la carga de Google Earth donde verá gráficamente la situación geográfica del vehículo.

Para acceder a las cuentas del viaje seleccionado presione “Cuentas”.

Si desea regresar pulse “Volver”.

### 3.9.5. Localización

En este apartado controlará la localización de los vehículos que formen parte de los viajes en curso.

**Viajes**

id	Conductor	Cabeza	Remolque	Movil	fecha_inicio	fecha_fin
25	yo	67	98	99889	2006-01-11	0000-00-0

Modificar Timer: 1 | Ver Ruta | Terminar Viaje

**Pedidos**

id	Cliente	empresa	Precio	Mercancia	Kg Carga	Medidas E

Ver Pedido

Volver

Figura 3.19 Gestión de viajes localizables.

Puede visualizar los viajes y los pedidos de la misma manera que en “Viajes Pendientes”. Al seleccionar un viaje se cargará el valor del timer que le corresponda.

Si desea finalizar un viaje seleccionado haga clic en “Terminar Viaje”. Al terminar un viaje termina también la localización del mismo.

Si desea modificar el intervalo temporal de búsqueda, introduzca en el selector el nuevo valor y haga clic en “Modificar Timer”.

Puede ver la ruta de localización que realiza el vehículo correspondiente al viaje seleccionado, para ello haga clic en “Ver Ruta”. Este botón solo está activo en caso de que el viaje haya sido previamente localizado. Recibirá un archivo .kml que deberá abrir, para iniciar la carga de Google Earth donde verá gráficamente la situación geográfica del vehículo.

### 3.9.6. Cuentas

Aquí podrá gestionar las cuentas asociadas a cada viaje.

Campo	Valor
Gasoil (L)	0
Gasoil E	200
Km	25
Consumo (E)	0
Consumo (L)	0
Km llegada	0
Km salida	0
Gastos Autopista	100
Gasto Adicional	0

**Ingresos**

**Gastos**

**Rentabilidad**

Figura 3.20 Gestión de cuentas.

Si es la primera vez que accede a las cuentas del viaje seleccionado, aparecerán todos los parámetros con valor 0 excepto “ingresos” que calcula la suma total del precio de los pedidos realizados en dicho viaje. Usted debe introducir los valores adecuados a cada parámetro en la columna “Valor”, clicando en la fila correspondiente.

Una vez insertados los valores deseados, si usted desea calcular los gastos y rentabilidad generada por el viaje haga clic en “Calcular”

A continuación para salvar los valores actualizados presione “Aceptar”, en caso contrario pulse “Cancelar”.

## 3.10. Anexo

### 3.10.1. Consideraciones especiales

- En las “Tablas de previsualización” los parámetros ofrecidos son definidos por el administrador.
- El formato de las fechas debe seguir el siguiente patrón: YYYY-MM-DD.
- Los nombres de usuario y contraseñas son proporcionadas por el administrador.
- La visualización óptima del programa es para una resolución de 1024x768 y el navegador configurado a pantalla completa.
- Diversos mensajes de alerta pueden aparecer según diversas situaciones para pedir al usuario la confirmación de acciones o para impedir acciones ilegales.
- Los requisitos mínimos son los correspondientes a los establecidos para la última versión de Macromedia Flash Player, versión 7 o superior.
- El sistema operativo recomendado es Windows, ya que en él se han realizado los testeos de funcionamiento oportunos, sin embargo, si Macromedia proporciona el reproductor de Flash para otro sistema operativo, teóricamente funcionará correctamente.

### 3.10.2. Leyenda de colores

- Botoneras Verdes: Creación, modificación o eliminación de datos.
- Botones Standard: Procesamiento de datos.
- Botones Amarillos: Acceso a opciones.
- Botones Azul / Rojo: Aceptar / Cancelar.
- Botones Gris: Volver a menú correspondiente.
- Tablas de previsualización:
  - Azul: Datos propios
  - Naranja: Pedidos
  - Amarillo: Viajes

### 3.10.3. Definiciones

- Viajes pendientes: Son aquellos viajes creados que aun no se han iniciado y por lo tanto permanecen en espera.
- Viajes en curso: Hace referencia a viajes que han pasado del estado “pendientes” a estar realizándose. Mientras no se indique el fin del viaje, éste permanecerá en curso.
- Viajes realizados: Aquellos viajes que ya han sido ejecutados, y por tanto, ya han pasado los estados “pendiente” y “en curso”.

## Capítulo 4

# Manual de administrador

### 4.1. Introducción

En este capítulo se detallan:

- Las indicaciones oportunas para la instalación de:
  - El software de gestión
  - El sistema de localización
- El manual de usuario de la herramienta de configuración
- Como usar el servidor de localización (sólo Vodafone)

El administrador debe facilitar a los usuarios la dirección Web donde sea instalada la aplicación para su uso.

## 4.2. Instalación de los módulos

Es necesario instalar:

- Servidor Web con módulos para PHP MySQL
- MySQL
- AMFPHP
- Google Earth

### 4.2.1. Servidor Web + Php + MySQL

Puede realizar la instalación de cada uno de estos módulos por separado aunque se recomienda el uso del paquete WinLAMP, distribuido libremente en Internet, que incluye el servidor Web Apache, la base de datos MySQL, y los módulos necesarios PHP para el servidor.

En caso de querer instalar servidor Web y base de datos de forma independiente debe seguir los manuales de instalación de cada uno de los programas, que podrá encontrar en las páginas oficiales correspondientes.

**Advertencia:** Existen numerosas configuraciones para el servidor de su aplicación, tales como el puerto en el que trabaja, los permisos asignados a los archivos, la configuración de conexiones seguras https, etc. Estos parámetros debe configurarlos según sus necesidades. Para ello consulte el manual del servidor instalado.

Se recomienda que sólo tengan permisos de lectura el archivo “programa.swf” y la carpeta “flashservices” que contiene los módulos AMFPhP (incluida en el CD).

### 4.2.2. AMFPHP

Con su debida configuración pueden instalarse AMFPHP y MySQL en servidores diferentes, sin embargo se asumirá que ambos módulos están ubicados en el mismo servidor. En caso contrario póngase en contacto con el distribuidor del software.

Se proporciona en el CD de instalación una carpeta llamada AMFPHP donde se indican los pasos necesarios para su instalación.

### 4.2.3. Google Earth

En cuanto a la aplicación Google Earth, es de distribución libre y podrá encontrarla en la dirección [7].

Debe configurar los mime-types en el servidor, para ello siga las instrucciones que encontrará en la carpeta “Google Earth” del CD proporcionado.

### 4.2.4. Instalación de la aplicación

En el CD proporcionado encontrará la carpeta ‘aplicacion’, debe copiar el contenido de ésta a la carpeta ‘htdocs’ del servidor.

### 4.2.5. Anexo de instalación de la base de datos MySQL + AMFPHP

Una vez esté todo instalado, debe especificar un nombre de usuario y una contraseña en la base de datos. Puede hacerlo mediante comandos, cuya explicación está detallada en el manual de usuario de MySQL. Sin embargo podemos usar el usuario ‘root’ que viene por defecto, para ello debemos establecerle un nuevo password escribiendo en una ventana de Ms-Dos:

```
ruta_de_instalacion_de_mysql\bin\> mysqladmin -u root -p password “nuevo password”
```

A continuación presione Enter. El usuario y password especificados serán los mismos que deberá escribir en el fichero ‘usuario\_AMFPHP.txt’ y ‘password\_AMFPHP.txt’ para que los módulos Php puedan conectar con la base de datos.

El siguiente paso será importar la estructura de la base de datos proporcionada en el CD, para ello siga las instrucciones proporcionadas en la carpeta ‘BBDD’.



## 4.3. Herramienta de configuración

Esta herramienta servirá para configurar diversos aspectos de la aplicación principal. Abra una ventana de navegador y vaya a la siguiente dirección: <http://localhost/config>

### 4.3.1. Acceso

El primer paso será introducir nuestro nombre de usuario y contraseña de administrador proporcionados por el distribuidor del software. A continuación haga clic en “Aceptar”.

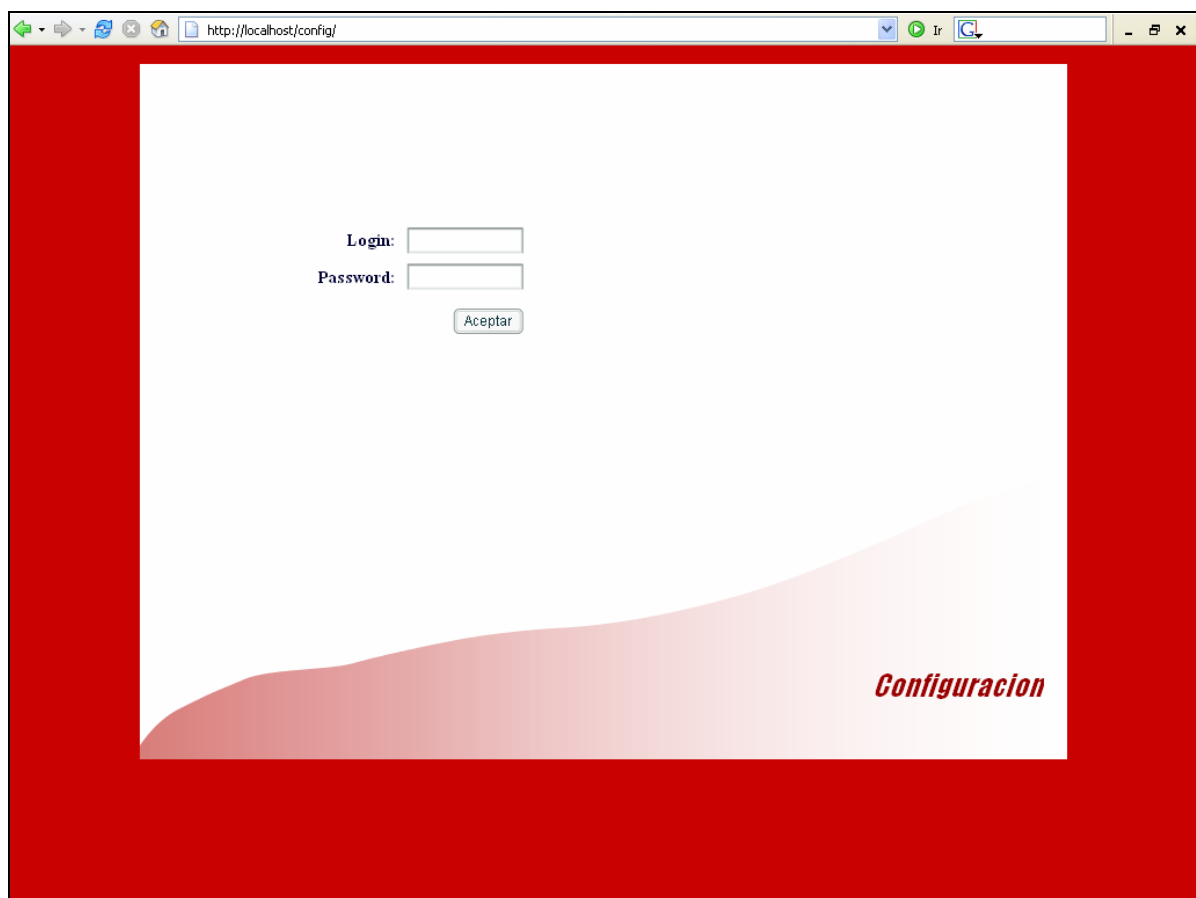


Figura 4.1 Acceso a configuración.

### 4.3.2. Menú

Desde el menú podremos acceder a las diferentes opciones. Estas son “Configurar Tablas” y “Configurar Perfiles”. Pulse el botón según la función deseada.

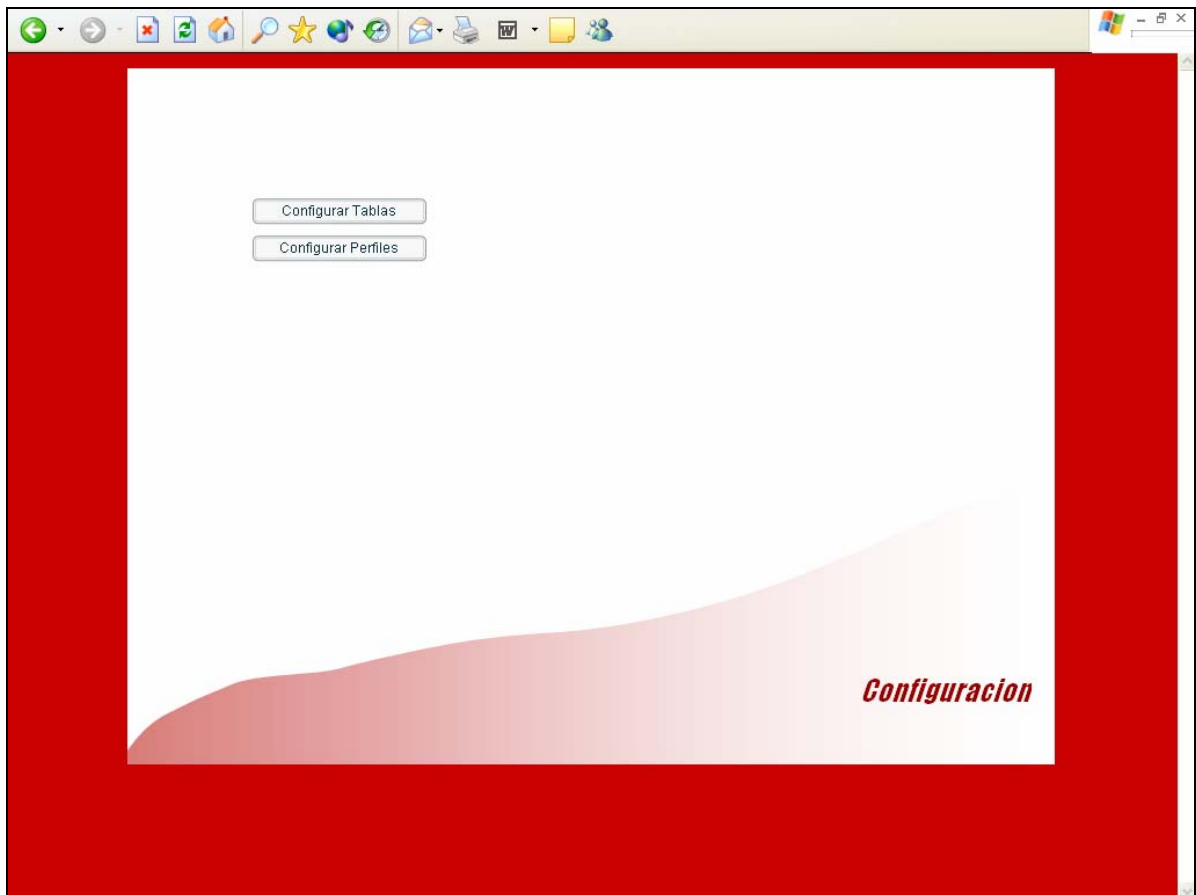


Figura 4.2 Menú inicial.

### 4.3.3. Configurar Tablas

En este apartado podrá configurar los parámetros que se cargarán en las tablas de previsualización de la aplicación principal.

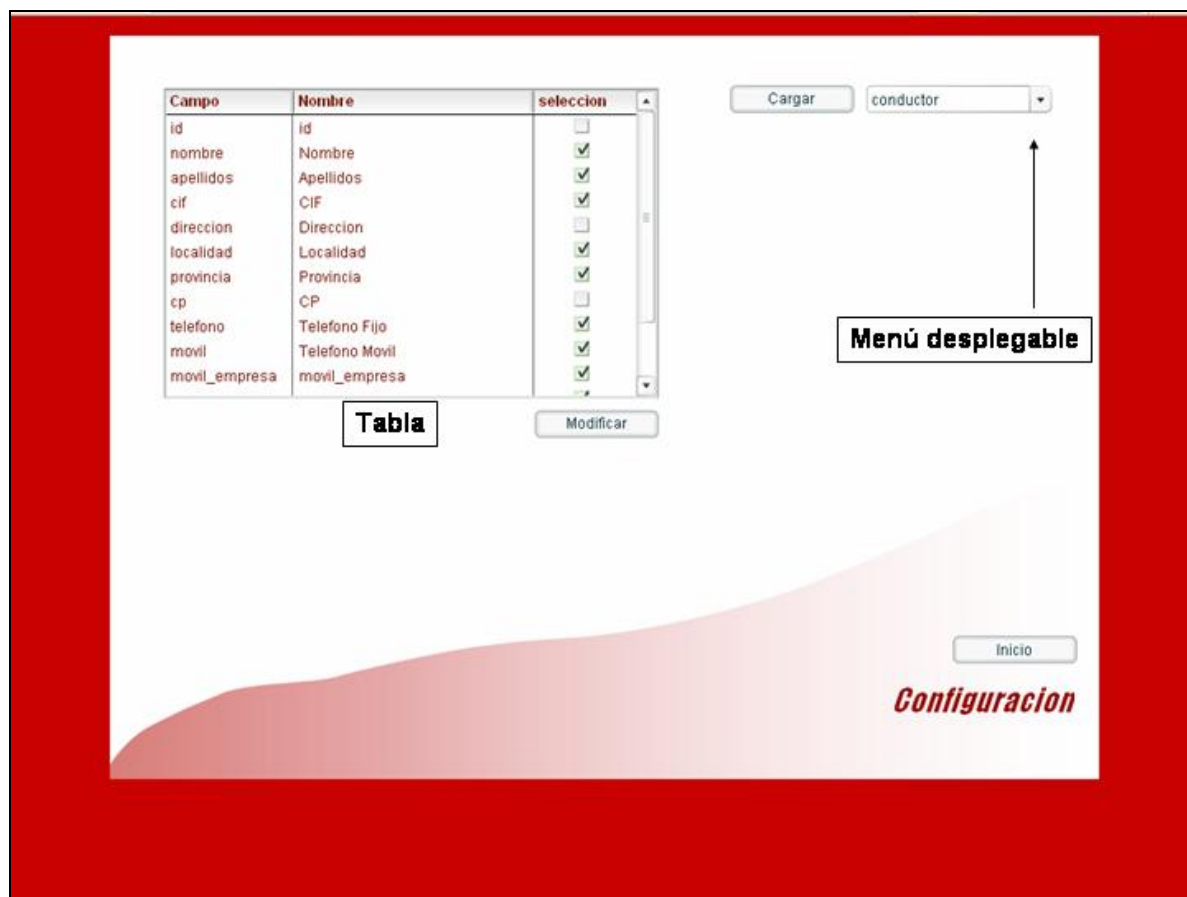


Figura 4.3 Configuración de tablas.

Pinche en el menú desplegable para seleccionar el tipo de información deseado. A continuación pulse “cargar”. Se cargarán todos los parámetros de información que constituye el tipo de datos elegido. En la tabla podrá observar para cada campo, el nombre que tiene en la base de datos, el nombre con el que se representará en la aplicación y una casilla que indica si forma parte o no en la tabla de previsualización.

Puede cambiar el nombre a mostrar, y la selección de previsualización pinchando en la fila correspondiente del campo deseado.

Una vez esté configurado a su elección haga clic en “Modificar”, recibirá un mensaje de confirmación.

Para volver pulse en “Inicio”.

### 4.3.4. Configurar Perfiles

En esta sección podrá gestionar los usuarios de la aplicación, asignándoles nombres de usuario, contraseñas y permisos.

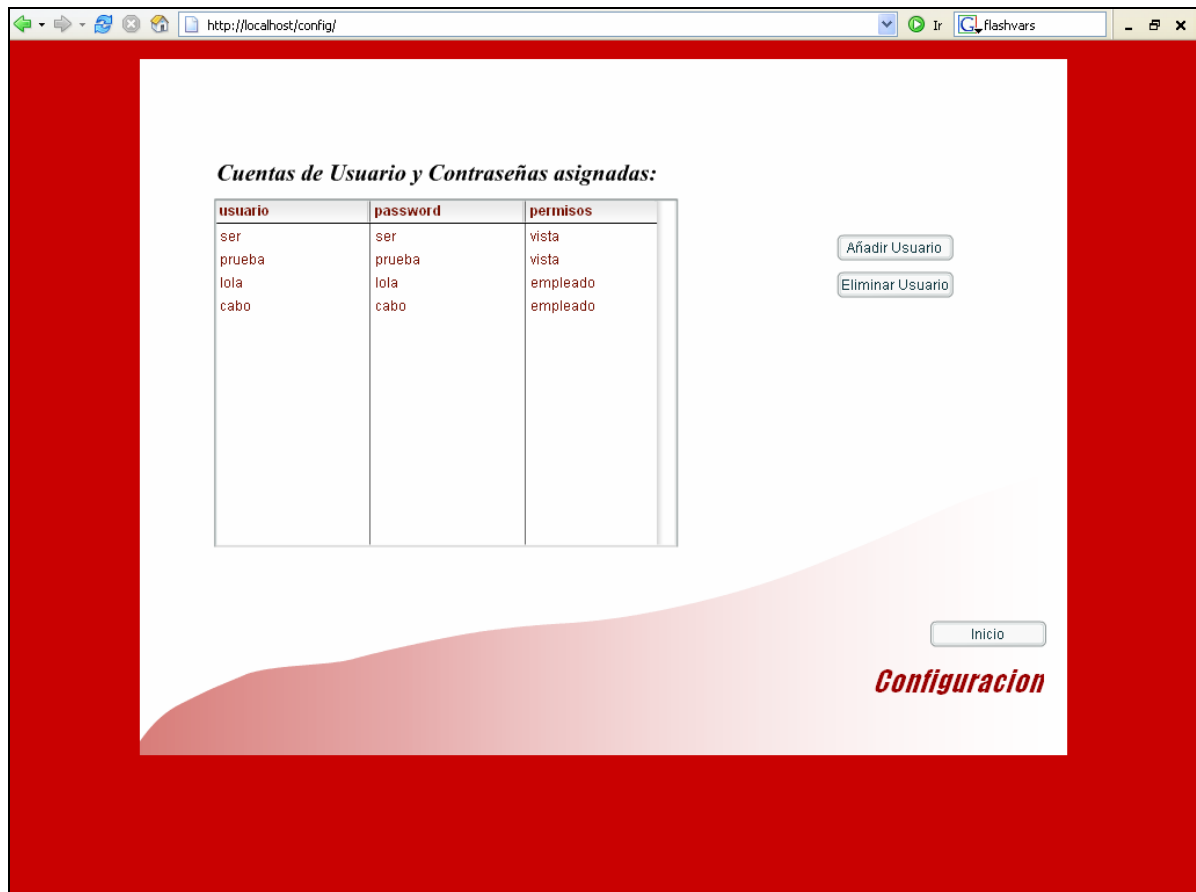


Figura 4.4 Menú de configuración de perfiles.

Al acceder a esta opción verá en la tabla los usuarios actualmente registrados. Si desea añadir o eliminar un usuario pulse el botón correspondiente.

Si desea volver haga clic en “Inicio”.

### 4.3.5. Añadir usuarios

Tras pulsar el respectivo botón accederá a esta opción donde realizará la adición del usuario.

*Introducir usuario, contraseña y permisos para asignar.*

Usuario:

Password:

Permisos: Vista

Vista

Empleado

Aceptar

Volver

**Configuracion**

Figura 4.5 Creación de nuevos usuarios.

Establezca un nombre de usuario y un password. Pinche en el menú desplegable para asignar los permisos deseados. A continuación haga clic en “Aceptar”.

Si desea volver pulse en el botón “Volver”.

### 4.3.6. Eliminar usuarios

Tras pulsar el respectivo botón accederá a esta opción donde realizará la eliminación del usuario.

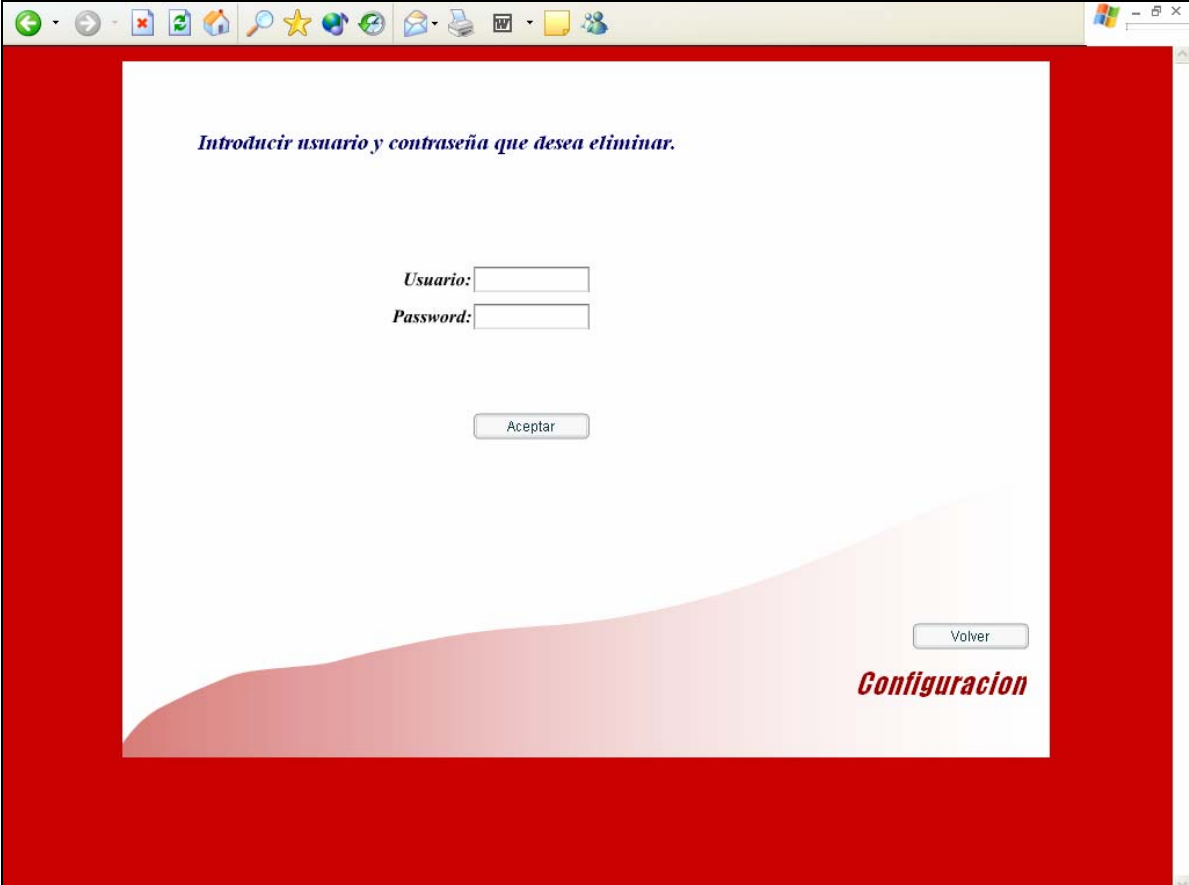
A screenshot of a web browser window showing a user deletion form. The browser's address bar and toolbar are visible at the top. The main content area has a white background with a red border. At the top, the text "Introducir usuario y contraseña que desea eliminar." is displayed. Below this, there are two input fields: "Usuario:" and "Password:". A button labeled "Aceptar" is positioned below the password field. In the bottom right corner, there is a button labeled "Volver" and the word "Configuracion" in a bold, italicized font. A large, semi-transparent red watermark is overlaid on the bottom right portion of the page.

Figura 4.6 Eliminación de usuarios.

Únicamente introduzca el usuario y password y pulse en “Aceptar” para proceder a eliminar el usuario.

Si desea volver pulse en el botón “Volver”.

## 4.4 Sistema de localización

***ADVERTENCIA: SOLO DISPONIBLE PARA EL OPERADOR VODAFONE.***

Previamente al uso del sistema de localización, debe ponerse en contacto con Vodafone y registrarse como usuario del servicio Plug InRed. Se le facilitarán los datos y claves oportunas. Siga las instrucciones necesarias para la instalación de la aplicación “Configura” y los pasos requeridos para permitir que un teléfono móvil sea localizado.

El sistema de localización está basado en un programa desarrollado en Java que actúa como “servidor de localización”. Para su instalación únicamente copie el archivo ejecutable en el equipo donde esté instalado MySQL.

Ejécute cuando desee iniciar el servicio de localización, una ventana emergente pedirá que introduzca el número y contraseña de su servicio Plug InRed. Una vez introducido correctamente mostrará información relativa a la localización. Para detenerlo únicamente cierre esta ventana. Dispone de todas las instrucciones oportunas en el CD proporcionado.

El sistema de localización solo funcionará cuando el servidor de localización esté activado, de lo contrario no se registrarán las coordenadas solicitadas.

## Capítulo 5

# Conclusión

Se ha desarrollado una aplicación que consigue solucionar la problemática propuesta. No obstante esta primera versión puede sufrir modificaciones y mejoras en versiones futuras que optimicen o añadan nuevas funcionalidades. Se pensó para una siguiente versión la inclusión de nuevas herramientas tales como:

- Nuevos estadísticos
- Facturación
- Proveedores
- Algoritmo de cálculo de ruta óptima

Sin embargo no se ha usado el programa un tiempo suficiente como para la “depuración óptima” es decir, tras un periodo relativamente largo de uso (6 o 12 meses) podrían aparecer nuevas funciones a implementar, o la necesidad de modificar alguna ya existente. Entre los nuevos servicios que podrían ser de interés cabe destacar, el envío de SMS, MMS desde la aplicación, o “alertas” que previeran a los conductores de la situación de las carreteras incluidas en sus rutas.

Aun así se cree que esta primera versión cumplirá con margen las necesidades básicas ya que el desarrollo ha sido guiado por una PYME del sector.



En cuanto al tema de localización, no es estrictamente necesario el uso de la tecnología GSM, es decir, se podría implementar el uso de equipos GPS preinstalados en los vehículos. Esto abriría las puertas a la inclusión de otros servicios variables según las necesidades del sector que podrían gestionarse desde la aplicación desarrollada. Incluso, teóricamente, la aplicación es compatible con el futuro sistema de localización europeo Galileo. Los distintos sistemas de localización podrían estar implementados y ser la PYME la que decidiera el uso de una u otra tecnología.

Todas estas incertidumbres se irán resolviendo con el paso del tiempo.

Apéndice

**A**

## **API de Vodafone. Estructura.**

## 1 Introducción

Este documento describe el uso del API Plug InRed por parte de los desarrolladores que programen aplicaciones haciendo uso del mismo.

## 2 Arquitectura del API

El API tiene una arquitectura basada en un único usuario que tiene distintas aplicaciones. Dichas aplicaciones pueden usar el API para:

- Envío de:
  - SMS
  - MMS
  - LBS
  - WAP-PUSH
- Recepción de notificación de mensajes nuevos
- Lectura de la bandeja de entrada de SMSs nuevos / existentes
- Gestión de autorespuestas

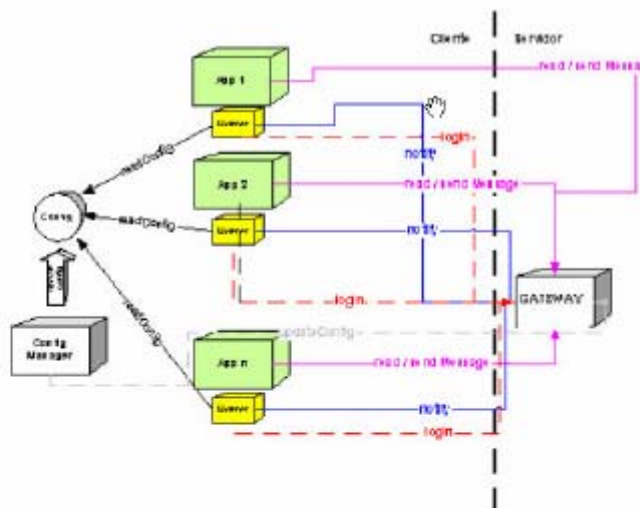


Fig 1. Arquitectura de la aplicación

## Contenido del CD-ROM

El CD-ROM que acompaña al proyecto contiene el código fuente de la aplicación desarrollada, y del servidor de localización, así como la estructura de la base de datos y las aplicaciones necesarias para su uso, así como instrucciones que facilitan la instalación.

Directorio	Fichero	Contenido
/	Leeme.txt	Instrucciones y contenido.
/AMFPHP	Amfphp.zip Instalación.txt	Módulos PHP y AMFPHP. Instrucciones de instalación.
/BBDD	Bbdd.zip Instalación.txt	Estructura de la base de datos. Instrucciones de instalación.
/Aplicación	Aplicación.zip Instalación.txt	Código fuente y aplicación. Instrucciones de instalación.
/Herramienta de configuración	Configura.zip Instalación.txt	Código fuente y herramienta de configuración. Instrucciones de instalación.
/Programas	WinLAMP.exe GoogleEarh.exe	Aplicaciones extra.

# Bibliografía

- [1] [www.arsys.es](http://www.arsys.es)
- [2] <http://sourceforge.net/projects/winlamp/>
- [3] [www.macromedia.com](http://www.macromedia.com)
- [4] [www.amfphp.org](http://www.amfphp.org)
- [5] [www.vodafone.es](http://www.vodafone.es)
- [6] <http://es.wikipedia.org/wiki/JDBC>
- [7] <http://earth.google.com/>