

**ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN
UNIVERSIDAD POLITÉCNICA DE CARTAGENA**



Proyecto Fin de Carrera

Desarrollo de una aplicación distribuida de control y gestión logística



AUTOR: Justo Alcón Cazorla
DIRECTOR: Esteban Egea López
CODIRECTOR: Alejandro Martínez Sala

Enero / 2.008



Autor	Justo Alcón Cazorla
E-mail del Autor	justo.alcon@gmail.com
Director(es)	Esteban Egea López
E-mail del Director	esteban.egea@upct.es
Codirector(es)	Alejandro Martínez Sala
Título del PFC	Desarrollo de una aplicación distribuida de control y gestión logística.
Descriptores	C#, localización, aplicaciones distribuidas.
<p>Resumen</p> <p>El objetivo de este Proyecto Fin de Carrera es el desarrollo de una aplicación distribuida de control y gestión logística. Esta aplicación permitirá la localización de vehículos, no sólo como medida de control de una empresa, sino también para poder ofertar mayores servicios a los clientes.</p> <p>Una vez finalizado este Proyecto Fin de Carrera se podrá realizar una gestión de todos los datos relacionados con la logística en una empresa y localizar sus vehículos desde una misma aplicación. La localización de los vehículos se realizará gracias a unos dispositivos, asociados a un vehículo o un conductor, que irán enviando sus coordenadas a una base datos, de forma periódica.</p> <p>Todos los datos referentes a la logística y los servicios adicionales ofrecidos se encontrarán en un servidor, al que se tendrá acceso remotamente desde los clientes. El control y la gestión de los datos se realizarán desde un interfaz de usuario situado en los clientes, el cual permitirá, además, la representación mediante Google Earth de las últimas ubicaciones conocidas de los vehículos y las rutas seguidas por éstos.</p>	
Titulación	Ingeniería de Telecomunicación
Intensificación	
Departamento	Tecnología de la Información y las Comunicaciones
Fecha de Presentación	Enero – 2.008

Índice de Contenidos

Capítulo 1 - Introducción	6
1.1 Introducción.	6
1.2 Soluciones existentes.	6
1.3 Solución propuesta.	8
1.3.1 Solución básica a los objetivos fijados para el Proyecto Fin de Carrera.....	9
1.3.2 ¿Qué base de datos?	10
1.3.2.1 Base de datos Access.	10
1.3.2.2 Base de datos Microsoft SQL Server.	10
1.3.2.3 Base de datos MySQL.....	11
1.3.2.4 Base de datos PostgreSQL.	11
1.3.2.5 Base de datos elegida.	12
1.3.3 ¿Qué lenguaje de programación usar?	12
1.3.4 ¿Cómo se comunican Cliente y Servidor?	13
1.3.4.1 Protocolos y estándares utilizados por los servicios Web.....	14
1.3.4.2 Ventajas de los servicios Web.	14
1.3.4.3 Inconvenientes de los servicios Web.	15
1.4 Resumen de contenidos del presente documento.	15
Capítulo 2 - Arquitectura del sistema	16
2.1 Introducción a la arquitectura del sistema.	16
2.2 Arquitectura de la Base de Datos.	16
2.2.1 Introducción.	16
2.2.2 Requisitos para Microsoft SQL Server Express.....	17
2.2.3 Principales Características de Microsoft SQL Server Express.	17
2.2.4 Nomenclatura utilizada para las tablas.....	18
2.2.5 Modelo Entidad – Relación de la Base de Datos.	18
2.2.5.1 Diagrama E-R del sistema global.....	18
2.2.5.2 E-R de cada tabla.	20
2.2.5.2.1 E-R ItemDatos.	20
2.2.5.2.2 E-R PropietarioDatos.....	20
2.2.5.2.3 E-R CoordenadasDatos.....	21
2.2.5.2.4 E-R ContextoDatos.....	22
2.2.5.2.5 E-R OrigenDestinoDatos.....	22
2.2.5.2.6 E-R ConductorDatos.....	23
2.2.5.2.7 E-R CamionDatos.....	24
2.2.5.2.8 E-R RemolqueDatos.	24
2.2.5.2.9 E-R CargaDatos.	25
2.2.5.2.10 E-R ItemContextoCoordenadas.	25
2.2.5.2.11 E-R ItemContexto.....	26
2.2.5.2.12 E-R ContextoCargas.	26
2.2.6 Uso de SQL Server 2005 Express Edition.	27
2.2.6.1 Habilitar conexiones remotas con SQL Server Express Edition.....	27
2.2.6.2 Protocolo de seguridad utilizado en Microsoft SQL Server.	28
2.2.6.3 Permisos para acceder al servidor Microsoft SQL Server Express Edition.	28
.....	28
2.3 Arquitectura de la aplicación.	28
2.3.1 Introducción a la arquitectura de la aplicación.	28
2.3.2 Requisitos para Microsoft Visual Studio C# Express.	28

2.3.3 Características de Microsoft Visual Studio C# Express.	29
2.3.4 Arquitectura del Servidor.	29
2.3.4.1 Introducción.	29
2.3.4.2 Diagrama de la estructura del Servidor.	30
2.3.4.3 Arquitectura de la capa DataAccess.	31
2.3.4.3.1 Subcapa SQLProvider.	31
2.3.4.3.2 Subcapa Provider.	32
2.3.4.3.3 Subcapa DataAccess.	32
2.3.4.4 Arquitectura de la capa Business.	32
2.3.4.5 Arquitectura de la capa Facade.	33
2.3.4.6 Arquitectura de la capa Service.	33
2.4 Arquitectura del Cliente.	33
2.4.1 Introducción.	33
2.4.2 Diagrama de la estructura del Cliente.	34
2.4.3 Estructura del Cliente.	34
2.5 Arquitectura del Sistema Global.	35
2.5.1 Introducción.	35
2.5.2 Objetivos del Sistema Global.	35
2.5.3 Ventajas del Sistema desarrollado.	36
2.5.4 Diagrama del Sistema Global.	36
2.5.5 Descripción de la integración de los diferentes componentes principales.	37
Capítulo 3 - Manual del usuario.	38
3.1 Introducción al manual de usuario.	38
3.2 Requisitos del Sistema de Localización de Camiones.	38
3.3 Requisitos del Servidor.	38
3.4 Requisitos de los Clientes.	39
3.5 Instalación y configuración de .NET Framework 2.0.	39
3.6 Instalación y configuración de los elementos específicos del Servidor.	39
3.6.1 Instalación y configuración de Microsoft SQL Server Express.	39
3.6.2 Instalación y configuración del Servidor Web de IIS de Microsoft.	41
3.7 Instalación y configuración de la aplicación Servidor.	42
3.8 Instalación y configuración de la aplicación Cliente.	44
3.9 Manual del usuario del Cliente.	44
3.9.1 Introducción.	44
3.9.2 Ejecución y entorno principal de la aplicación Cliente.	45
3.9.3 Descripción del entorno gráfico principal.	45
3.9.4 Descripción de la información representada en cada pestaña.	47
3.9.5 Descripción de los formularios disponibles.	56
3.9.6 Descripción de las herramientas para mostrar puntos y rutas en Google Earth.	73
.....	73
Capítulo 4 - Conclusiones y líneas futuras.	75
4.1 Conclusiones.	75
4.1.1 Conclusiones de Microsoft SQL Server.	75
4.1.2 Conclusiones de Microsoft Visual Studio C#.	75
4.1.3 Conclusiones del Sistema de Localización.	76
4.2 Líneas futuras.	76
Capítulo 5 - Bibliografía.	77

Capítulo 1

Introducción

1.1 Introducción.

El presente documento se encargará de describir una herramienta software desarrollada como Proyecto Fin de Carrera. El software en cuestión será un *Sistema de Localización de Camiones* para aplicarlo en el ámbito logístico de las empresas.

Este *Sistema de Localización de Camiones* ofrece:

- Una base de datos diseñada para contener todos los datos necesarios para la logística y los servicios ofrecidos.
- Un software para la gestión remota de los datos de logística y la representación de puntos y rutas en Google Earth.
- Una documentación explicando la estructura del sistema para futuras implementaciones y el manual del usuario necesario para la correcta instalación, configuración y utilización de la aplicación.

Las herramientas software de este tipo ofrecen un método para el control de la situación geográfica de los vehículos de una empresa, permitiendo una mejor optimización en el transporte. Estas herramientas permiten ofrecer una serie de servicios y facilidades, tanto para el cliente como para la empresa, permitiendo reducir costes, tiempo y esfuerzos.

Cabe destacar que en este Proyecto Fin de Carrera no se incluye el sistema necesario para que las coordenadas, mandadas por los diferentes dispositivos de localización desde los vehículos, lleguen a la base de datos y sean registradas. Atendiendo a esto, se dará por hecho que ya existe un sistema implementado para tal fin.

La finalidad de este *Sistema de Localización de Camiones* no será ofrecer una herramienta final, totalmente funcional y libre de errores, sino la de mostrar una versión de prueba de las posibilidades que se pueden alcanzar con su desarrollo en futuros proyectos. El fin último que deberá perseguir esta aplicación será el de unificar todas las funcionalidades y servicios necesarios para poder realizar un control avanzado de la actividad empresarial y de la flota de vehículos. Para no olvidar al usuario, se deberá perseguir la claridad y la sencillez sin perder la funcionalidad necesaria.

1.2 Soluciones existentes.

Dentro del ámbito de este Proyecto Fin de Carrera se pueden encontrar diversas soluciones comerciales como pueden ser las siguientes:

1. Seguimiento y Localización de Flotas de NetProfider.

Está destinada al seguimiento, posicionamiento y localización de flotas de vehículos y sus incidencias en tiempo real. Las comunicaciones se realizan con un servidor (Server) a través de Internet usando tecnología GPRS. A este servidor pueden estar conectados uno o más clientes

(Pc's, PDA, Teléfono móvil,...) para realizar el seguimiento y control en tiempo real de los vehículos.

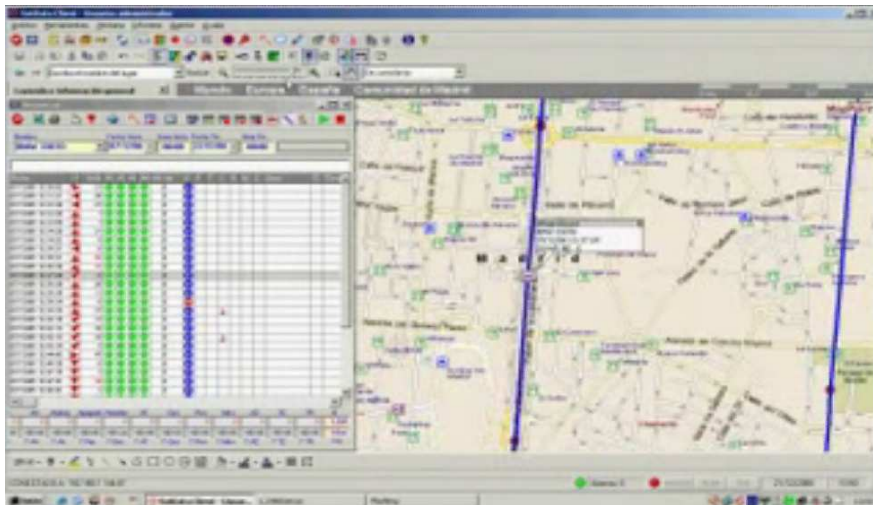


Ilustración 1. Captura del sistema de localización de NetProvider.

2. OnlineAVL de Crambo Wireless.

Es, junto con el dispositivo instalado, el corazón del sistema de gestión de flotas. Proporciona una visibilidad total de la actividad de la empresa gracias a lo que se puede conseguir una mayor productividad y aumentar la satisfacción de los clientes. OnlineAVL le permite gestionar su flota de forma eficaz desde casi cualquier PC. El programa sólo requiere una conexión a Internet, de este modo, evita los gastos que puede suponer mantener un centro de control y reduce los riesgos de caídas del sistema.



Ilustración 2. Captura del sistema OnlineAVL de Crambo Wireless.

3. TransCar.

Transcar es una aplicación para la gestión integral de empresas que se dediquen al transporte de mercancías, agencias de transporte o logística. Este programa de transportes además de la gestión propia, incluye módulos de tesorería, CRM, Remuneración, Exportación a los principales programas de contabilidad...

Con TransCar se podrá llevar una gestión más exacta y ágil de una empresa, y de esta manera podrá reducir costes y aumentar la productividad. Igualmente, una vez se comience a trabajar con la aplicación, el aprendizaje será rápido, con lo que conseguirá además de ahorrar una gran cantidad de trabajo, aumentar de forma significativa la calidad de éste.

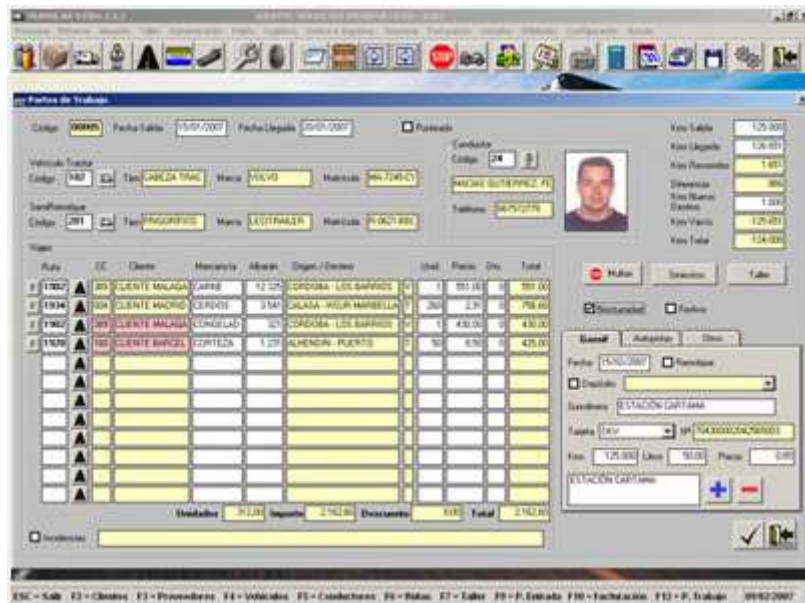


Ilustración 3. Captura del sistema de localización de TransCar.

1.3 Solución propuesta.

Para este Proyecto Fin de Carrera se planteó el crear una aplicación que gestionase una base de datos diseñada específicamente para el ámbito logístico de las empresas.

En el desarrollo de la aplicación se necesitaron tomar las siguientes decisiones:

- Base de datos a utilizar en el sistema, con la que debería comunicarse la aplicación *Servidor*.
- Lenguaje de programación a utilizar tanto en la aplicación *Cliente* como en la aplicación *Servidor*.
- Modo de comunicación entre los *Clientes* y el *Servidor*.

En los próximos subapartados se discutirá y razonará las posibles soluciones que se barajaron y cuáles fueron las adoptadas finalmente.

1.3.1 Solución básica a los objetivos fijados para el Proyecto Fin de Carrera.

Antes de decidir las tecnologías a usar, se va a mostrar un boceto de lo que sería la solución para este Proyecto Fin de Carrera.

El *Sistema de Localización* estará compuesto por:

- Una base de datos desarrollada específicamente para el sistema. En ella se almacenará toda la información necesaria.
- Una aplicación *Servidor* a la que se conectará los *Cientes* remotamente y que gestionará la información de la base de datos.
- Un servidor Web, gracias al cual se podrá tener acceso a la aplicación *Servidor*.
- Una aplicación *Cliente*, ubicada en cada una de las computadoras de los usuarios, que se conectará con el *Servidor*.
- Un conjunto de dispositivos de localización instalados en los vehículos. Estos dispositivos, que estarán basados en GPS o GPRS, mandarán al *Servidor* periódicamente su identificador y las coordenadas en las que se encuentran en ese instante.

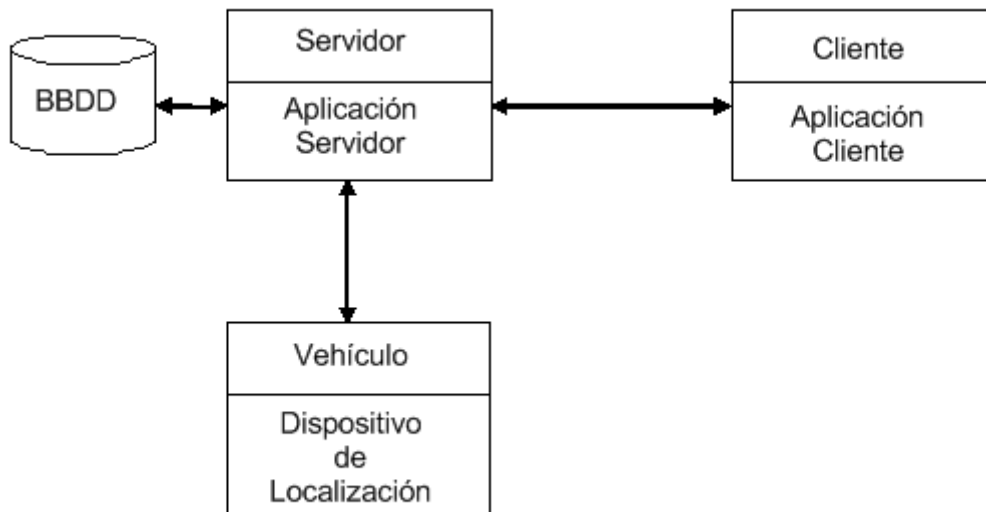








Ilustración 4. Diagrama de la solución básica.

1.3.2 ¿Qué base de datos?

Dentro de las bases de datos disponibles podemos encontrar diferentes alternativas:

Criterios	Bases de datos			
	Access	SQL Server	MySQL	PostgreSQL
Plataforma			 / 	 / 
Velocidad	-	✓	✓	-
Volumen Datos	-	✓	✓	✓
Integridad	-	✓	-	✓
Potencia	-	✓	✓	✓
Coste/MB	✓	-	✓	✓

✓ Positivo - Negativo

Tabla 1. Comparativa de bases de datos obtenida de Internet.

Como se puede observar en la tabla anterior, se estudiaron cuatro posibilidades y se analizaron sus pros y contras en sus características más destacables.

1.3.2.1 Base de datos Access.

Disponibles sólo bajo 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.

1.3.2.2 Base de datos Microsoft SQL Server.

Disponibles sólo bajo sistema operativo Windows.

SQL Server es el sistema de bases de datos más completo y potente, y resulta 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).

El principal punto negativo que tiene esta base de datos es su relación "Coste/MB". Si el volumen de datos a manejar no excede de 4 GB se puede optar por elegir la versión Express, versión gratuita que tiene muchas de las ventajas y las prestaciones de la versión comercial.

1.3.2.3 Base de datos MySQL.

Disponibles bajo sistema operativo Windows o Linux.

MySQL tiene como principales características:

- Su gran velocidad y su precio reducido. Es el servidor de bases de datos más rápido de todos los que ofrecemos y el de menor precio por MB.
- 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 su coste por MB es menor y además 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.

1.3.2.4 Base de datos PostgreSQL.

Disponibles bajo sistema operativo Windows o Linux.

Características:

- 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.
- 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, pues su coste por MB es menor 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 de los estudiados.

1.3.2.5 Base de datos elegida.

Finalmente, la base de datos que se eligió para el PFC fue Microsoft SQL Server Express Edition. Esta base de datos, además de ser gratuita, cumplía ampliamente con las necesidades requeridas para el *Sistema de Localización de Camiones*. Sus principales limitaciones estaban en el volumen máximo de datos y en la plataforma a usar, pero ninguna de estas limitaciones suponía un problema.

Además de esto, una de las principales ventajas de usar SQL Server Express de Microsoft era poder obtener un montón de facilidades en la implementación si se usaba alguna de las versiones de Visual Studio.

1.3.3 ¿Qué lenguaje de programación usar?

Una vez se seleccionó la base de datos a utilizar se tuvo que elegir cuál sería el lenguaje de programación empleado en el desarrollo de la aplicación. De entre todas las posibles elecciones, se decidió la programación en C# con Microsoft Visual C# Studio 2005 Express Edition (versión gratuita de Microsoft Visual Studio 2005 Professional).

Las razones por las que se eligió C#, frente a los demás, fueron las siguientes:

- Ventajas frente a C y C++:
 - Compila a código intermedio (CIL) independiente del lenguaje en que haya sido escrita la aplicación e independiente de la máquina donde vaya a ejecutarse.
 - Recolección de basura automática.
 - Eliminación del uso punteros, en C# no se necesitan.
 - Capacidades de reflexión.
 - No hay que preocuparse por archivos de cabecera ".h".
 - No importa el orden en que hayan sido definidas las clases ni las funciones.
 - No hay necesidad de declarar funciones y clases antes de definirlos.
 - No existen las dependencias circulares.
 - Soporta definición de clases dentro de otras.
 - No existen funciones, ni variables globales, todo pertenece a una clase.
 - Todos los valores son inicializados antes de ser usados (automáticamente por defecto, o manualmente desde constructores estáticos).
 - No se pueden utilizar valores no booleanos (enteros, coma flotante...) para condicionales. Es mucho más limpio y menos propenso a errores.
 - Puede ejecutarse en una sandbox restringida.

- Ventajas Frente a C++ y Java:
 - Concepto formalizado de los métodos get y set, con lo que se consigue código mucho más legible.
 - Gestión de eventos (usando delegados) mucho más limpia.
- Ventajas frente a Java:
 - El rendimiento es, por lo general, mucho mejor.
 - CLI está estandarizado, mientras que los bytecodes de java no lo están.
 - Soporta bastantes más tipos primitivos (*value types*), incluyendo tipos numéricos sin signo.
 - Indizadores que permiten acceder a cualquier objeto como si se tratase de un array.
 - Compilación condicional.
 - Aplicaciones multi-hilo simplificadas.
 - Soporta la sobrecarga de operadores, que aunque pueden complicar el desarrollo son opcionales y algunas veces muy útiles.
 - Permite el uso (limitado) de punteros cuando realmente se necesiten, como al acceder a librerías que no se ejecuten sobre la máquina virtual.

1.3.4 ¿Cómo se comunican Cliente y Servidor?

Para la comunicación entre los *Clientes* y el *Servidor* se eligió usar los servicios Web de Microsoft, integrados en sus herramientas de desarrollo y que resuelven el problema de interoperabilidad.

Un servicio Web es una colección de protocolos y estándares que sirven para intercambiar datos, a través de una red o de Internet, entre distintas aplicaciones. Estas aplicaciones pueden estar desarrolladas en lenguajes de programación diferentes y pueden ser ejecutadas sobre cualquier plataforma. Gracias a la adopción de estándares abiertos se consigue resolver los problemas de interoperabilidad.

Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Por otro lado, el organismo WS-I es el encargado de mejorar la interoperabilidad entre distintas implementaciones de servicios Web y desarrollar diversos perfiles para definir de manera más exhaustiva los estándares a usar.

Las principales razones de elegir los servicios Web, como solución a los problemas de interoperabilidad, han sido las siguientes:

- Las redes de las organizaciones protegidas con firewalls normalmente cierran casi todos los puertos TCP salvo el 80, que es precisamente el usado por los navegadores Web. Como los servicios Web están basados en HTTP sobre TCP en el puerto 80, no son bloqueados por los firewalls
- Gracias a SOAP se dispone de una buena interfaz para acceder a las funcionalidades de otros ordenadores en red.
- Los servicios Web pueden aportar gran independencia entre la aplicación que usa el servicio Web y el propio servicio. De este modo, los cambios efectuados en uno no afectarán al otro.

1.3.4.1 Protocolos y estándares utilizados por los servicios Web.

A continuación se listarán la colección de protocolos y estándares utilizados por los servicios Web para más tarde explicarlos resumidamente.

- Web Services Protocol Stack (pila de protocolos para los servicios Web): Así se denomina al conjunto de servicios y protocolos de los servicios Web.
- XML (Extensible Markup Language): Es el formato estándar para los datos que se vayan a intercambiar.
- SOAP (Simple Object Access Protocol) o XML-RPC (XML Remote Producer Call): Protocolos sobre los que se establece el intercambio.
- Otros protocolos: los datos en XML también pueden enviarse de una aplicación a otra mediante protocolos normales como HTTP (Hypertext Transfer Protocol), FTP (File Transfer Protocol), o SMTP (Simple Mail Transfer Protocol).
- WSDL (Web Services Description Languages): Es el lenguaje de la interfaz pública para los servicios Web. Es una descripción basada en XML de los requisitos funcionales necesarios para establecer una comunicación con los servicios Web.
- UDDI (Universal Description, Discovery and Integration): Protocolo para publicar la información de los servicios Web. Permite comprobar qué servicios Web están disponibles.
- WS-Security (Web Service Security): Protocolo de seguridad aceptado como estándar por OASIS (Organization for the Advancement of Structured Information Standards). Garantiza la autenticación de los actores y la confidencialidad de los mensajes enviados.

1.3.4.2 Ventajas de los servicios Web.

Las ventajas de usar los servicios Web, para la interoperabilidad entre distintas aplicaciones, son las siguientes:

- Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
- Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.
- Al apoyarse en HTTP, los servicios Web pueden aprovecharse de los sistemas de seguridad firewall sin necesidad de cambiar las reglas de filtrado.
- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.
- Permiten la interoperabilidad entre plataformas de distintos fabricantes por medio de protocolos estándar.

1.3.4.3 Inconvenientes de los servicios Web.

Los inconvenientes de usar los servicios Web son los siguientes:

- Para realizar transacciones no pueden compararse en su grado de desarrollo con los estándares abiertos de computación distribuida como CORBA (Common Object Request Broker Architecture).
- Su rendimiento es bajo si se compara con otros modelos de computación distribuida, tales como RMI (Remote Method Invocation), CORBA, o DCOM (Distributed Component Object Model). Es uno de los inconvenientes derivados de adoptar un formato basado en texto. Y es que entre los objetivos de XML no se encuentra la concisión ni la eficacia de procesamiento.
- Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en firewall cuyas reglas tratan de bloquear o auditar la comunicación entre programas a ambos lados de la barrera.

1.4 Resumen de contenidos del presente documento.

Una vez introducido el presente Proyecto Fin de Carrera se explicará, de modo resumido, el contenido de los próximos capítulos.

- Capítulo 2: Arquitectura del sistema. Se proporcionará la documentación necesaria para la comprensión de la estructura del sistema para que todo aquel que desee modificar o implementar nuevas funcionalidades pueda hacerlo. En este capítulo se incluirán diagramas y tablas explicativas de la arquitectura.
- Capítulo 3: Manual del usuario. Se proporcionará la documentación necesaria para que cualquier usuario pueda instalar, configurar y manejar adecuadamente la aplicación del *Sistema de Localización de Camiones*. Se mostrarán diversas capturas junto a las instrucciones oportunas para facilitar el aprendizaje.
- Capítulo 4: Conclusiones y líneas futuras. Se explicarán brevemente las conclusiones, indicando los conocimientos obtenidos con este Proyecto Fin de Carrera y la experiencia tenida tanto con SQL Server como con Visual Studio C#. Además, se plantearán ideas para futuras implementaciones.

Capítulo 2

Arquitectura del sistema

2.1 Introducción a la arquitectura del sistema.

En este capítulo se explicará detenidamente la arquitectura y el funcionamiento del sistema. Para ello se describirán los diferentes componentes, su funcionalidad y la forma en que se comunican éstos dentro del sistema.

La arquitectura básica del sistema está compuesta por cuatro elementos principales:

- Base de Datos: utilizando Microsoft SQL Server Express y gestionada con Microsoft SQL Server Management Studio Express.
- Aplicación Servidor: programado en C# con Microsoft Visual Studio C# 2005 Express.
- Servidor Web de Microsoft: Permite el acceso por red o Internet desde cada aplicación *Cliente*, mediante SOAP, a la aplicación *Servidor*.
- Aplicación Cliente: programado en C# con “Microsoft Visual Studio C# 2005 Express”.

Cada uno de estos elementos principales será desarrollado en apartados de forma independiente, mostrando los diagramas y explicando su funcionalidad. Finalmente se dedicará un apartado para tratar la forma en que se integrarán los diferentes componentes para crear el conjunto.

2.2 Arquitectura de la Base de Datos.

2.2.1 Introducción.

La base de datos utilizada para la aplicación ha sido Microsoft SQL Server Express, versión gratuita de Microsoft SQL Server. Esta versión, por ser gratuita, tiene una serie de limitaciones en las características de memoria, base de datos y esquemas. A pesar de estas limitaciones, las prestaciones ofrecidas por la versión gratuita son suficientes para las necesidades requeridas en el sistema.

Entre las ventajas que tiene usar Microsoft SQL Server Express, se puede destacar la posibilidad de usar SQL Server Management Studio Express para gestionar la base de datos. Esta herramienta gratuita facilita las tareas necesarias para la gestión de Microsoft SQL Server Express, permitiendo modificar las configuraciones de la base de datos y crear, modificar y eliminar tablas de forma gráfica.

Para poder usar otras bases de datos, como pueden ser Oracle Database o MySQL, será necesario implementar en el servidor la clase abstracta “Provider” de la capa “DataAccess”. La implementación de la clase abstracta “Provider” (permitiendo el polimorfismo con dicha clase) deberá tener la misma funcionalidad que la implementación ya realizada para Microsoft SQL Server y que se encuentra en la clase

“SQLProvider”. En cualquier caso, este tema se tratará en el apartado dedicado al *Servidor* del sistema.

2.2.2 Requisitos para Microsoft SQL Server Express.

Los requisitos para Microsoft SQL Server Express son los siguientes:

- Procesador:
 - Mínimo: 600 MHz o superior, compatible con Pentium III.
 - Recomendado: 1 GHz o superior.
- Framework:
 - .NET Framework 2.0.
- Sistema Operativo:
 - Microsoft Windows XP con SP2.
 - Microsoft Windows 2000 Profesional con SP4.
 - Microsoft Windows Server 2003 Standard con SP1 o posterior.
 - Microsoft Windows Server 2003 Enterprise con SP1 o posterior.
 - Microsoft Windows Server 2003 Datacenter con SP1 o posterior.
 - Microsoft Windows Server 2003 Web Edition con SP1 o posterior.
 - Microsoft Windows Small Business Server 2003 con SP1 o posterior.
 - Microsoft Windows Vista Home Basic o superior (añadiendo SQL Express Advanced SP2).
 - Microsoft Windows XP Embedded SP2 Feature Pack 2007.
 - Microsoft Windows Embedded for Point of Service SP2.

2.2.3 Principales Características de Microsoft SQL Server Express.

Microsoft SQL Server 2005 Express se basa en Microsoft SQL Server 2005 y admite la mayoría de las características del motor de base de datos de ese producto y de las versiones anteriores.

A continuación se muestran algunas de las principales características y componentes que admite:

- Procedimientos almacenados.
- Administrador de configuración de SQL Server.
- Vistas.
- Réplica (sólo como suscriptor).
- Desencadenadores.
- Optimizador de consultas avanzado.
- Cursores.

- SMO/RMO.
- Utilidades sqlcmd y osql.
- Integración con “Microsoft Visual Studio 2005”.
- Niveles de aislamiento de instantánea.
- Service Broker (únicamente como cliente).
- Compatibilidad con XML nativo, incluidos Xquery y esquemas XML.
- CLR de SQL.
- Compatibilidad con lenguajes Transact-SQL.
- Conjunto de resultados activos múltiples (MARS).
- Conexión de administrador dedicada.

2.2.4 Nomenclatura utilizada para las tablas.

Las tablas representativas de las entidades principales se nombrarán usando: “*Nombre entidad + Datos*” (Ejemplo: “CargaDatos”). En los casos en los que se han usado tablas para relacionar diferentes entidades principales, la nomenclatura es la suma de los nombres de las entidades (Ejemplos: “ItemContextoCoordenadas” es una tabla donde se indica las coordenadas que tenía un ítem y el contexto que estaba cursando en esas coordenadas; “ContextoCargas” es una tabla donde se le asocia a cada contexto una o varias cargas).

En las tablas, las claves serán nombradas con “id + *Nombre entidad*” para las entidades principales y “id + *Nombre tabla*” para las demás tablas. Así, la clave de “CargaDatos” será “idCarga”, y la de “ContextoCargas” será “idContextoCargas”.

2.2.5 Modelo Entidad – Relación de la Base de Datos.

2.2.5.1 Diagrama E-R del sistema global.

El sistema está modelado por las siguientes entidades:

- Ítem.
- Propietario.
- Coordenadas.
- Contexto.
- Origen/Destino.
- Conductor.
- Camión.
- Remolque.
- Carga.

Además de las tablas mencionadas, han sido necesarias otras tablas para complementar la información y así formar la estructura global de la base de datos.

A continuación se muestra el diagrama global de la base de datos:

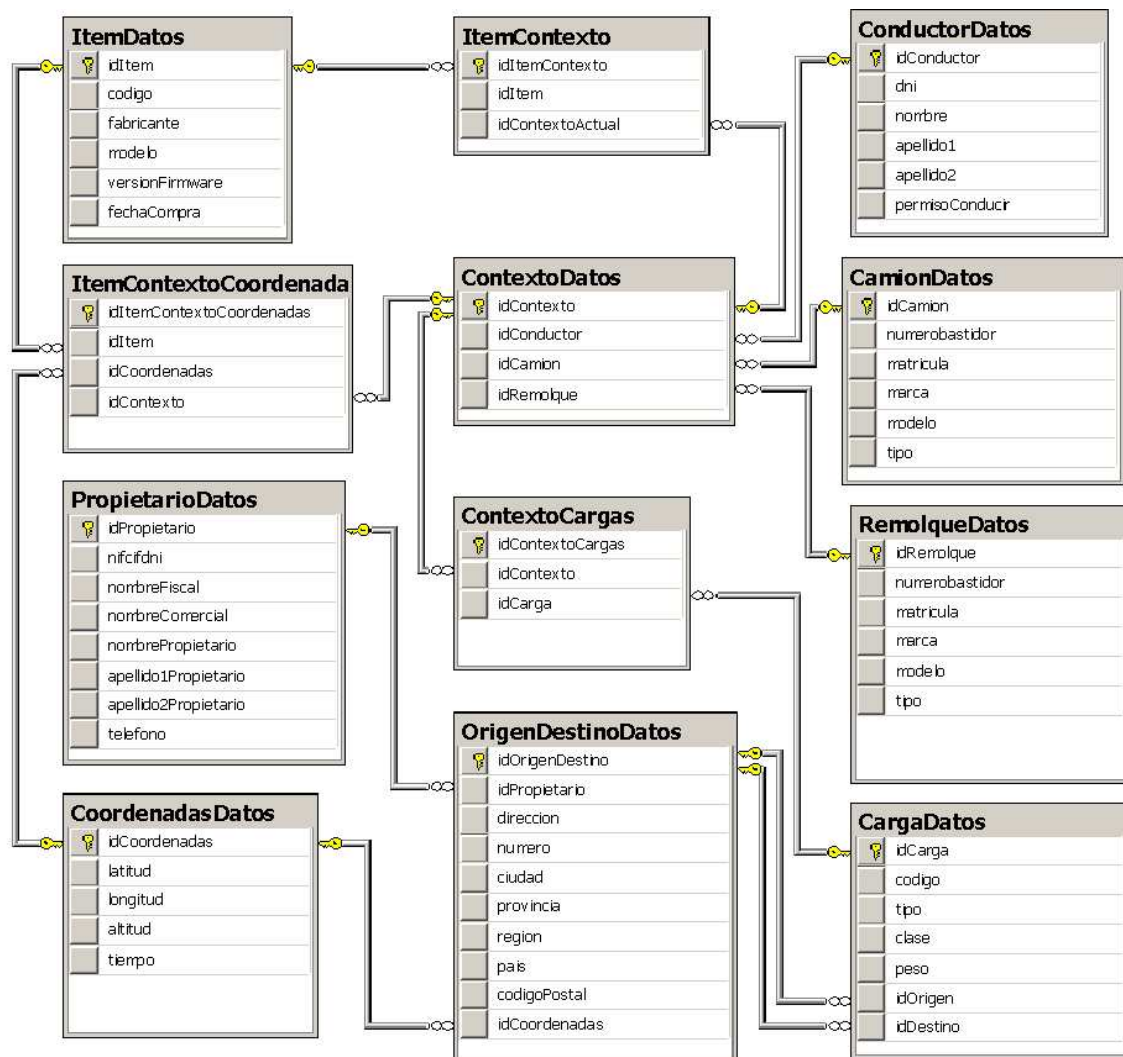


Ilustración 5. Diagrama de la estructura de la base de datos.

En este diagrama, un contexto hará referencia a un viaje, al que asociaremos un conductor, un camión, un remolque, un ítem y una o más cargas (ya que en cada viaje se podrá llevar más de una carga). Para poder asociar a cada contexto una o más cargas se utilizará la tabla “ContextoCargas”, que tendrá una entrada por cada carga registrada en la tabla “CargaDatos”.

Una carga podrá tener tanto un origen como un destino. Cada origen/destino podrá tener, además de una serie de datos, unas coordenadas y un propietario. Este propietario podrá ser una empresa o un individuo.

Para registrar las coordenadas de un ítem en cada momento junto con el contexto que tiene asociado en ese momento, la base de datos dispondrá de la tabla “ItemContextoCoordenadas”. En la tabla “ItemContexto” tendremos un campo llamado “idContextoActual” que será el que indique qué contexto está cursando cada ítem y así, tener toda la información necesaria para crear registros en la tabla “ItemContextoCoordenadas”.

2.2.5.2 E-R de cada tabla.

2.2.5.2.1 E-R ItemDatos.

ItemDatos			
Clave	Atributos	Tipo de datos	Permite Nulos
Primaria	idItem	Int	No
	codigo	varchar(50)	No
	fabricante	varchar(50)	No
	modelo	varchar(50)	No
	versionFirmware	varchar(50)	Sí
	fechaCompra	varchar(50)	Sí

Tabla 2. Tabla “ItemDatos” de la base de datos.

- “idItem”: Es un identificador autoincremental usado como clave principal.
- “codigo”: Código o número de serie del dispositivo de localización.
- “fabricante”: Empresa que ha fabricado el dispositivo de localización.
- “modelo”: Nombre del modelo del dispositivo.
- “versionFirmware”: Versión de Firmware del dispositivo.
- “fechaCompra”: Fecha en la que la empresa compró el dispositivo.

2.2.5.2.2 E-R PropietarioDatos.

PropietarioDatos			
Clave	Atributos	Tipo de datos	Permite Nulos
Primaria	idPropietario	int	No
	nificfdni	varchar(50)	No
	nombreFiscal	varchar(50)	Sí

	nombreComercial	varchar(50)	Sí
	nombrePropietario	varchar(50)	Sí
	apellido1Propietario	varchar(50)	Sí
	apellido2Propietario	varchar(50)	Sí
	telefono	varchar(50)	Sí

Tabla 3. Tabla “PropietarioDatos” de la base de datos.

- “idPropietario”: Es un identificador autoincremental usado como clave principal en esta tabla.
- “nifcidni”: CIF/DNI/NIF del propietario. Será una cosa u otra dependiendo de si el propietario es una persona o es una empresa.
- “nombreFiscal”: Nombre fiscal de la empresa.
- “nombreComercial”: Nombre comercial de la empresa.
- “nombrePropietario”: En caso de que el propietario sea una persona, el nombre de esa persona.
- “apellido1Propietario”: Primer apellido del propietario.
- “apellido2Propietario”: Segundo apellido del propietario.
- “telefono”: Teléfono de contacto.

2.2.5.2.3 E-R CoordenadasDatos.

CoordenadasDatos			
Clave	Atributos	Tipo de datos	Permite Nulos
Primaria	idCoordenadas	Int	No
	latitud	float	No
	longitud	float	No
	altitud	float	Sí
	tiempo	datetime	Sí

Tabla 4. tabla “CoordenadasDatos” de la base de datos.

- “idCoordenadas”: Es un identificador autoincremental usado como clave principal en esta tabla.

- “latitud”: Latitud de las coordenadas.
- “longitud”: Longitud de las coordenadas.
- “altitud”: Altitud de las coordenadas.
- “tiempo”: Marca temporal de la toma de las coordenadas.

2.2.5.2.4 E-R ContextoDatos.

ContextoDatos			
Clave	Atributos	Tipo de datos	Permite Nulos
Primaria	idContexto	Int	No
	idConductor	Int	Sí
	idCamion	Int	Sí
	idRemolque	Int	Sí

Tabla 5. Tabla “ContextoDatos” de la base de datos.

- “idContexto”: Identificador autoincremental usado como clave principal de la tabla.
- “idConductor”: Identificador del conductor asociado a este contexto (o viaje).
- “idCamion”: Identificador del camión asociado a este contexto.
- “idRemolque”: Identificador del remolque asociado a este contexto.

2.2.5.2.5 E-R OrigenDestinoDatos.

OrigenDestinoDatos			
Clave	Atributos	Tipo de datos	Permite Nulos
Primaria	idOrigenDestino	Int	No
Externa	idPropietario	Int	Sí
	direccion	varchar(50)	No
	numero	varchar(50)	Sí

	ciudad	varchar(50)	No
	provincia	varchar(50)	Sí
	region	varchar(50)	No
	pais	varchar(50)	No
	codigoPostal	varchar(50)	Sí
Externa	idCoordenadas	Int	Sí

Tabla 6. Tabla “OrigenDestinoDatos” de la base de datos.

- “idOrigenDestino”: Identificador no autoincremental usado como clave principal de la tabla.
- “idPropietario”: Identificador del propietario al que pertenece este origen/destino (que en la mayoría de las ocasiones será un almacén).
- “direccion”: Dirección del origen/destino.
- “numero”: Número de la dirección.
- “ciudad”: Ciudad en la que se encuentra el origen/destino.
- “region”: Región del origen/destino.
- “pais”: País del origen/destino.
- “codigoPostal”: Código postal de origen/destino.
- “idCoordenadas”: Identificador de las coordenadas que tendrá el origen/destino.

2.2.5.2.6 E-R ConductorDatos.

ConductorDatos			
Clave	Atributos	Tipo de datos	Permite Nulos
Primaria	idConductor	Int	No
	dni	varchar(50)	No
	nombre	varchar(50)	No
	apellido1	varchar(50)	No
	apellido2	varchar(50)	No

	permisoConducir	varchar(50)	No
--	-----------------	-------------	----

Tabla 7. Tabla “ConductorDatos” de la base de datos.

2.2.5.2.7 E-R CamionDatos.

CamionDatos			
Clave	Atributos	Tipo de datos	Permite Nulos
Primaria	idCamion	Int	No
	numerobastidor	varchar(50)	No
	matricula	varchar(50)	No
	marca	varchar(50)	No
	modelo	varchar(50)	No
	tipo	varchar(50)	Sí

Tabla 8. Tabla “CamionDatos” de la base de datos.

- “idCamion”: Identificador autoincremental usado como clave principal de la tabla.
- “numerobastidor”: Número de bastidor del camión.
- “matricula”: Matrícula asociada al camión.
- “marca”: Fabricante del camión.
- “modelo”: Modelo del fabricante de camiones.
- “tipo”: Tipo de vehículo.

2.2.5.2.8 E-R RemolqueDatos.

RemolqueDatos			
Clave	Atributos	Tipo de datos	Permite Nulos
Primaria	idRemolque	Int	No
	numerobastidor	varchar(50)	No

	matricula	varchar(50)	No
	marca	varchar(50)	No
	modelo	varchar(50)	No
	tipo	varchar(50)	Sí

Tabla 9. Tabla “RemolqueDatos” de la base de datos.

- “idRemolque”: Identificador autoincremental usado como clave principal de la tabla.
- “numerobastidor”: Número de bastidor del remolque.
- “matricula”: Matrícula asociada al remolque
- “marca”: Fabricante del remolque.
- “modelo”: Modelo del fabricante de remolques.
- “tipo”: Tipo de remolque.

2.2.5.2.9 E-R CargaDatos.

CargaDatos			
Clave	Atributos	Tipo de datos	Permite Nulos
Primaria	idCarga	Int	No
	codigo	varchar(50)	No
	tipo	varchar(50)	Sí
	clase	varchar(50)	Sí
	peso	float	Sí
Externa	idOrigen	Int	Sí
Externa	idDestino	Int	Sí

Tabla 10. Tabla “CargaDatos” de la base de datos.

2.2.5.2.10 E-R ItemContextoCoordenadas.

ItemContextoCoordenadas			
Clave	Atributos	Tipo de datos	Permite Nulos
Primaria	idItemContextoCoordenadas	int	No
Externa	idItem	int	No
Externa	idCoordenadas	int	No
Externa	idContexto	int	Sí

Tabla 11. Tabla “ItemContextoCoordenadas” de la base de datos.

- “idItemContextoCoordenadas”: Identificador autoincremental usado como clave principal de la tabla.
- “idItem”: Identificador del ítem al que pertenecen las coordenadas recibidas.
- “idCoordenadas”: Identificador de las coordenadas.
- “idContexto”: Identificador del contexto que tiene asociado en ese momento el ítem.

2.2.5.2.11 E-R ItemContexto.

ItemContexto			
Clave	Atributos	Tipo de datos	Permite Nulos
Primaria	idItemContexto	Int	No
Externa	idItem	Int	No
Externa	idContextoActual	Int	Sí

Tabla 12. Tabla “ItemContexto” de la base de datos.

- “idItemContexto”: Identificador autoincremental usado como clave principal de la tabla.
- “idItem”: Identificador de un ítem.
- “idContexto”: Identificador del contexto que tiene actualmente asociado el ítem que acompaña a este contexto.

2.2.5.2.12 E-R ContextoCargas.

ContextoCargas			
Clave	Atributos	Tipo de datos	Permite Nulos
Primaria	idContextoCargas	int	No
Externa	idContexto	int	Sí
Externa	idCarga	int	No

Tabla 13. Tabla “ContextoCargas” de la base de datos.

- “idContextoCargas”: Identificador autoincremental usado como clave principal de la tabla.
- “idContexto”: Identificador de un contexto al que se asocia la carga que le acompaña en la tabla.
- “idCarga”: Identificador de la carga asociada al contexto.

2.2.6 Uso de SQL Server 2005 Express Edition.

A continuación se explicarán algunas consideraciones y configuraciones necesarias para utilizar esta base de datos.

2.2.6.1 Habilitar conexiones remotas con SQL Server Express Edition.

- Para habilitar un protocolo de red:
 1. En el menú “Inicio”, seleccione “Todos los programas”, seleccione “Microsoft SQL Server 2005” y haga clic en “Administrador de configuración de SQL Server”. Opcionalmente, puede abrir “Administración de equipos”; para ello, haga clic con el botón secundario en “Mi PC” y elija “Administrar”. En “Administración de equipos”, expanda “Servicios y Aplicaciones” y “Administrador de configuración de SQL Server”.
 2. Expanda “Configuración de red de SQL Server 2005” y, a continuación, haga clic en “Protocolos de *NombreServidor*”.
 3. En la lista de protocolos, haga clic con el botón secundario en el protocolo que desea habilitar y, a continuación, haga clic en “Habilitar”. El icono del protocolo cambiará para indicar que el protocolo está habilitado.
 4. Para deshabilitar el protocolo, realice el mismo procedimiento pero elija “Deshabilitar” en el paso 3.
- Para configurar un protocolo de red:
 1. En el menú “Inicio”, haga clic con el botón secundario en “Mi PC” y elija “Administrar”.

2. En “Administración de equipos”, expanda “Servicios y Aplicaciones”, “Administrador de configuración de SQL Server”, “Configuración del servidor de red” y “Protocolos de *NombreServidor*”, y, a continuación, haga clic en el protocolo que desea configurar. (Si selecciona TCP, tendrá la opción de habilitar direcciones IP específicas en el panel derecho.)
3. Haga clic con el botón secundario en el protocolo que desea configurar y, a continuación, elija “Propiedades”.
4. En “Propiedades”, puede establecer opciones específicas del protocolo.

Además, para que funcione “Microsoft SQL Server” en red habrá que deshabilitar el firewall y el antivirus o introducir la excepción adecuada en cada uno.

2.2.6.2 Protocolo de seguridad utilizado en Microsoft SQL Server.

Tanto en el servidor SQL como en la conexión desde la aplicación (variable “ConnectionString” del archivo de configuración en el servidor) se utiliza “Integrated Security” de Windows con el valor SSPI. SSPI (Security Support Provider Interface) permite a una aplicación usar diferentes modelos de seguridad disponibles en la computadora o en la red sin cambiar el interfaz con el sistema de seguridad.

2.2.6.3 Permisos para acceder al servidor Microsoft SQL Server Express Edition.

Usando SQL Server Management Studio Express, en el panel de la izquierda, haciendo clic con el botón derecho encima del nombre del servidor se selecciona “Propiedades -> Permisos”. En el panel de “Inicio de sesión o funciones se selecciona public” y en el panel “Permisos explícitos para public” se activan todas las casillas “WITH GRANT”.

2.3 Arquitectura de la aplicación.

2.3.1 Introducción a la arquitectura de la aplicación.

La aplicación *Servidor* y la aplicación *Cliente* del sistema han sido desarrollados en C# usando Microsoft Visual Studio C# 2005 Express, versión gratuita de Microsoft Visual Studio 2005. A pesar de tener una serie de limitaciones, esta versión es suficiente para el desarrollo de la aplicación.

2.3.2 Requisitos para Microsoft Visual Studio C# Express.

- Procesador:
 - Mínimo: Procesador de tipo Pentium a 600 MHz.
 - Recomendado: Procesador de tipo Pentium a 1 GHz.
- Sistema Operativo:

- Microsoft Windows® 2003 Server.
- Microsoft Windows XP.
- Microsoft Windows 2000.
- RAM:
 - Mínimo: 128 MB.
 - Recomendado: 256 MB.
- Disco Duro: Pueden ser necesarios hasta 1,3 GB de espacio disponible.
- Pantalla:
 - Mínimo: 800 x 600, 256 colores.
 - Recomendado: 1024 x 768, color de alta densidad de 16 bits.

2.3.3 Características de Microsoft Visual Studio C# Express.

Las principales características de Microsoft Visual Studio C# Express son las siguientes:

- Diseñadores visuales para la creación de Formularios Windows.
- Un editor de código de primera categoría con características de productividad mejoradas como IntelliSense Statement Completion, IntelliSense Code Snippets, y refactoring.
- Depurador simplificado que incluye Editar y Continuar y además otras mejoras como visualizadores de datos para DataSets, HTML, XML, y datos de texto.
- Habilidad para publicar y compartir automáticamente tus aplicaciones completas en Internet, tu red de área local o en CDs utilizando ClickOnce.
- Soporte para la creación de aplicaciones preparadas para datos utilizando SQL Server 2005 Express.
- El kit de inicio RSS Screensaver para construir protectores de pantalla personalizados para Windows.
- El kit de inicio Movie Collection: es una aplicación Windows Form que utiliza el servicio Web de “Amazon.com” para buscar dinámicamente títulos de películas. El kit demuestra tecnologías como: llamadas a servicios Web XML, databinding, almacenamiento local de datos usando SQL Server 2005 Express Edition, y más.
- Guías para programadores noveles incluyendo tutoriales paso a paso.

2.3.4 Arquitectura del Servidor.

2.3.4.1 Introducción.

El *Servidor* está compuesto por varias capas de abstracción. Estas capas facilitan la creación, modificación y eliminación de funcionalidad en las capas inferiores sin que influya a las capas superiores.

Las capas, nombradas de la inferior a la superior, son las siguientes:

- **DataAccess:** Provee un interfaz a la capa “Business” para insertar/modificar/eliminar datos en los registros de las tablas. Está dividida a su vez en subcapas.
- **Business:** Ofrece un interfaz a la capa “Facade”, abstrayendo una serie de funcionalidades implementadas en esta capa.
- **Facade:** Desacopla al usuario del *Servidor*.
- **Service:** Servicio Web usando SOAP para la comunicación entre cada uno de los *Cientes* y el *Servidor*.

En los próximos apartados se describirá la funcionalidad de cada una de las capas del *Servidor*.

2.3.4.2 Diagrama de la estructura del Servidor.

En este apartado se mostrará un diagrama UML que describe las clases que forman el *Servidor* y las relaciones que existen entre ellas. Estas clases estarán agrupadas en paquetes, los cuales representarán las diferentes capas de las que consta el *Servidor*.

Debido a que algunas clases tienen un gran número de atributos y métodos, la forma de representar el “Diagrama de Clases” será mostrando el diagrama con todas las clases contraídas (sin mostrar los atributos y los métodos, mostrando tan solo los nombres de las clases), dando una vista general de las relaciones existentes.

En este apartado no se proporcionará información sobre los métodos y los atributos ya que la funcionalidad de cada uno de ellos se puede ver en el API del *Servidor*. Este API se puede encontrar en el directorio “DocsServidor” del Proyecto, siendo el índice de dicho API el archivo “index.html”.

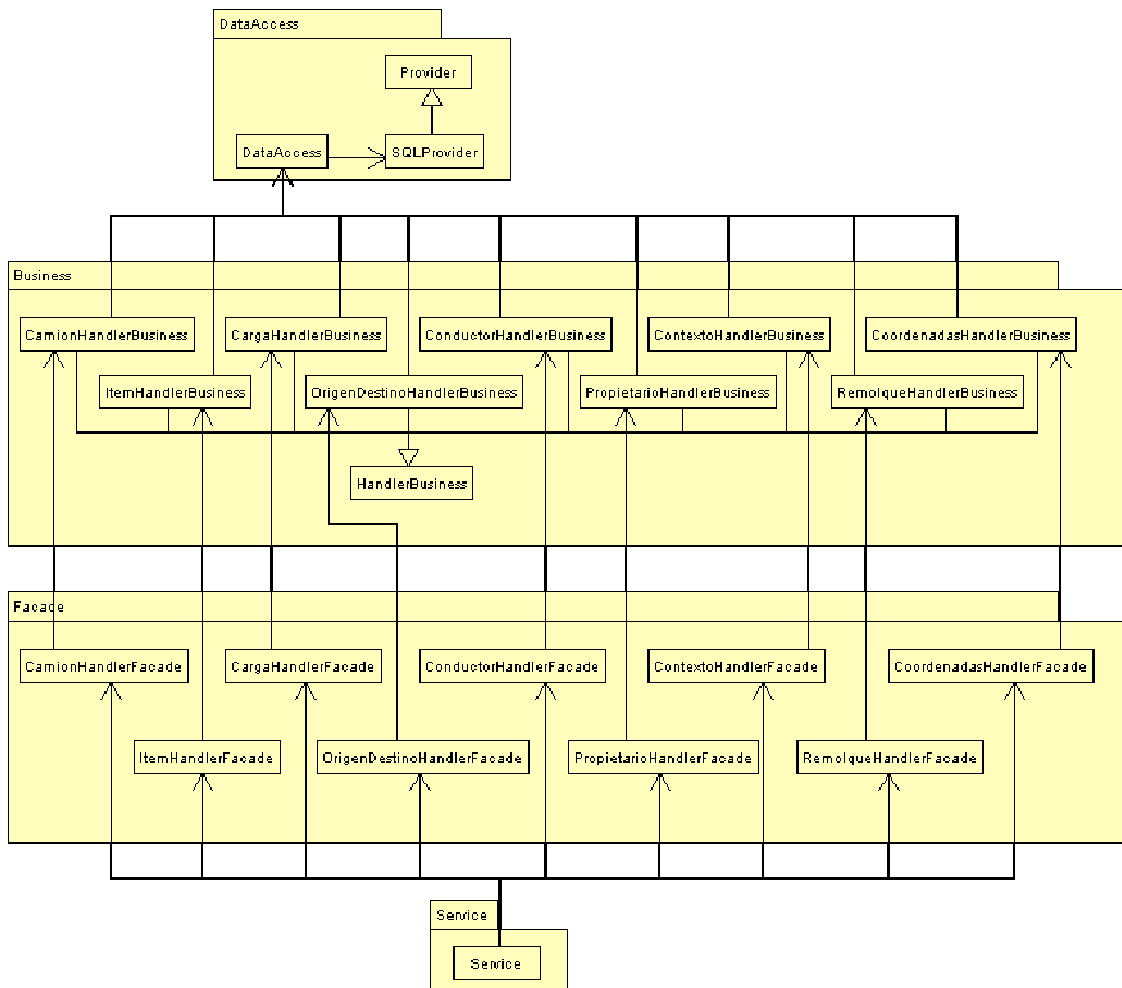


Ilustración 6. Diagrama UML contraído de la aplicación *Servidor*.

2.3.4.3 Arquitectura de la capa DataAccess.

Esta capa se comunica directamente con la base de datos y ofrece un interfaz a la capa “Business” para la creación, modificación y eliminación de información.

Esta capa está dividida a su vez en subcapas:

- **SQLProvider:** Proveedor de datos específico para Microsoft SQL Server 2005 o Microsoft SQL Server 2005 Express.
- **Provider:** Clase abstracta que funciona de clase padre para cada una de las implementaciones de los diferentes proveedores de datos.
- **DataAccess:** Proporciona un interfaz para la capa “Business”. Mediante el polimorfismo, esta subcapa tendrá instancias de la clase abstracta “Provider” y, dependiendo de la base de datos que se esté usando, las llamadas se harán a la referencia de una u otra implementación.

Debido a que esta capa tiene que comunicarse directamente con la base de datos, ambos tendrán que estar en la misma computadora.

2.3.4.3.1 Subcapa SQLProvider.

Se trata de una implementación específica para Microsoft SQL Server 2005 de la clase abstracta “Provider”. Esta implementación hace uso de ADO.NET para la comunicación con la base de datos y para los tipos de datos necesarios.

Esta subcapa es la que se comunica directamente con la base de datos y ofrece a la subcapa “DataAccess”, mediante polimorfismo a través de la clase “Provider”, una colección de métodos para la creación, modificación y eliminación de información en las diferentes tablas de la base de datos.

2.3.4.3.2 Subcapa Provider.

Clase abstracta de la que heredarán cada una de las implementaciones que se hagan de los diferentes proveedores de datos (como Microsoft SQL Server o MySQL). Todas las implementaciones que se hagan de esta clase tendrán que ofrecer la funcionalidad descrita en el API para cada uno de sus métodos (todos ellos abstractos).

Como ejemplo para la implementación correcta de nuevos proveedores de datos se recomienda tomar la implementación ya realizada para Microsoft SQL Server (clase “SQLProvider”).

2.3.4.3.3 Subcapa DataAccess.

Es un nivel de abstracción de las diferentes implementaciones de los proveedores de datos. Ofrece a la capa “Business” un interfaz para la comunicación con la base de datos, independientemente de cuál se utilice.

Esta subcapa lee de un fichero de configuración (“Web.config”) cual es la base de datos usada y los parámetros para conectarse a ella. Atendiendo a la base de datos especificada en el fichero de configuración, crea una instancia del proveedor de datos adecuado.

Una vez se ha creado una instancia del proveedor de datos adecuado, “DataAccess” hace de interfaz entre el proveedor de datos elegido y la capa “Business”. De esta forma, la base de datos usada y la forma en la que se interactúa con ella es transparente para las capas superiores.

2.3.4.4 Arquitectura de la capa Business.

También conocida como “Business Logic”. Es una capa que ofrece un interfaz a las capas superiores mediante la abstracción de “DataAccess”. Implementa funcionalidades para facilitar el acceso a la base de datos y proveer un interfaz más sencillo y claro.

Esta capa se encuentra formada por varios componentes, los cuales son agrupaciones lógicas de diferentes entidades de la base de datos. Gracias a esto se tendrá más transparencia de la estructura de la base de datos en las capas superiores, consiguiendo un interfaz más claro y sencillo.

Los componentes que forma la capa “Business” son:

- **CamionHandlerBusiness:** Gestiona los datos relacionados con los camiones.
- **CargaHandlerBusiness:** Gestiona los datos relacionados con los pedidos.
- **ConductorHandlerBusiness:** Gestiona los datos de los conductores.

- ContextoHandlerBusiness: Gestiona los datos sobre la configuración de los pedidos.
- CoordinadasHandlerBusiness: Gestiona los datos sobre las coordenadas de orígenes/destinos o de la ubicación de los dispositivos de localización o ítems.
- ItemHandlerBusiness: Gestiona los datos relacionados con los dispositivos de localización o ítems.
- OrigenDestinoHandlerBusiness: Gestiona los datos de los almacenes que puede ser usados como origen o destino de las mercancías.
- PropietarioHandlerBusiness: Gestiona los datos de los propietarios de los almacenes.
- RemolqueHandlerBusiness: Gestiona los datos relacionados con los remolques.

Esta capa se ha desarrollado para ser ejecutada en la misma computadora que la capa "DataAccess". A pesar de esto, se podría modificar para poder encontrarse en otra computadora, utilizando para ello el protocolo SOAP de Microsoft.

2.3.4.5 Arquitectura de la capa Facade.

También conocida como "Facade Pattern". Se encuentra formado por los mismos componentes que la capa inferior ("Business") y provee un interfaz sencillo, abstrayendo a las capas superiores de la implementación realizada en la capa inferior ("Business" en nuestro caso).

Esta capa permite desacoplar el interfaz de usuario del *Servidor*, de forma que sólo ofrecerá la funcionalidad a la que el usuario final pueda tener acceso.

2.3.4.6 Arquitectura de la capa Service.

Los diferentes *Cientes* se conectarán al *Servidor* haciendo uso del protocolo SOAP (basado en XML) a través de la capa "Service". Esta capa ofrecerá un interfaz para el acceso a los datos remotamente.

2.4 Arquitectura del Cliente.

2.4.1 Introducción.

El *Cliente* está diseñado para cumplir tres objetivos fundamentales:

- Comunicarse con el *Servidor* mediante SOAP. El *Cliente* envía y recibe datos utilizando este protocolo.
- Ofrecer un interfaz gráfico intuitivo, funcional y claro al usuario. Este interfaz gráfico se encargará de representar los datos y de ofrecer un conjunto de formularios para la creación/modificación/eliminación de información de la base de datos.
- Representación mediante Google Earth de puntos o rutas pertenecientes a diferentes ítems.

En los próximos apartados se explicará detenidamente la estructura y la funcionalidad del *Cliente*. No se entrará en detalle en el diseño del interfaz gráfico que será detallado en el capítulo dedicado al manual del usuario.

2.4.2 Diagrama de la estructura del Cliente.

En este apartado se mostrará un diagrama UML que describe las clases que forman el *Cliente* y las relaciones que existen entre ellas.

Debido a que algunas clases tienen un gran número de atributos y métodos, la forma de representar el “Diagrama de Clases” será mostrando el diagrama con todas las clases contraídas (sin mostrar los atributos y los métodos, mostrando tan solo los nombres de las clases), dando una vista general de las relaciones existentes.

En este apartado no se proporcionará información sobre los métodos y los atributos ya que la funcionalidad de cada uno de ellos se puede ver en el API del *Cliente*. Este API se puede encontrar en el directorio “DocsCliente” del Proyecto, siendo el índice de dicho API el archivo “index.html”.

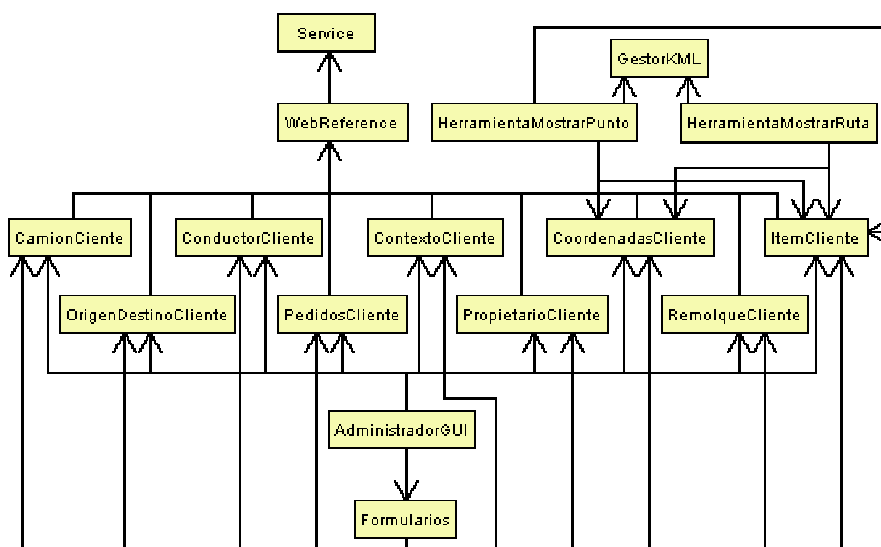


Ilustración 7. Diagrama UML contraído de la aplicación *Cliente*.

NOTA: Por motivos de espacio y de claridad no se han representado todos los formularios existentes. La clase “Formularios” representa al conjunto de todos los formularios. Mencionar que, aunque la clase “Formularios” tiene una instancia para cada clase del tipo “*Cliente” en el diagrama mostrado, en la realidad esto no será así y cada formulario contendrá sólo una instancia de las clases necesarias.

En el diagrama mostrado hay que destacar que la clase “WebReference” representa el interfaz de comunicación, por red o por Internet, con la clase “Service” del *Servidor*. Hay que recordar que la comunicación con el *Servidor* se realiza usando servicios Web de Microsoft (protocolo SOAP).

2.4.3 Estructura del Cliente.

El diseño gráfico del *Cliente* se basa en la agrupación de los datos en pestañas y en la creación, modificación, eliminación y filtrado de éstos mediante formularios. El

objetivo de este diseño es el de ofrecer al usuario un interfaz claro e intuitivo, permitiendo trabajar en los diferentes roles de forma ordenada.

Cada una de las pestañas representa los datos agrupados de forma lógica, facilitándose la representación y el entendimiento por parte del usuario. Para facilitar esta agrupación, se dispone de una clase para cada pestaña que implementa la funcionalidad relacionada con la representación, creación, modificación, eliminación y filtrado de datos.

Los formularios permitirán acceder a la funcionalidad disponible en cada pestaña. Cada formulario se muestra en forma de cuadro de diálogo, donde el usuario introducirá los datos pedidos. De esta forma, toda la funcionalidad implementada por la clase asociada a cada pestaña será accesible mediante formularios.

Además de esto, el cliente permite la representación de puntos y rutas en Google Earth. Para conseguir esto, se ha implementado una clase (llamada "GestorKML.cs") que permite generar ficheros KML que son mostrados automáticamente. Si se desea conocer con más detalle el funcionamiento de esta clase, consulte el API y el código referente a esta clase.

2.5 Arquitectura del Sistema Global.

2.5.1 Introducción.

Una vez se han comentado todos los componentes de los que consta el *Sistema de Localización de Camiones* se pasará a explicar la forma en la que se integran los diferentes elementos para formar el sistema global.

2.5.2 Objetivos del Sistema Global.

El principal objetivo que se pretendía alcanzar con el desarrollo de este *Sistema* era el obtener una herramienta que integrase los servicios más utilizados en el ámbito de la logística, así como ofrecer una serie de servicios adicionales útiles para los diferentes clientes.

Aunque éste era el objetivo final de la aplicación a desarrollar, también ha sido necesario cubrir otras necesidades o cuestiones sumamente importantes:

- El diseño de la base de datos se ha realizado de forma genérica. Esto ha sido así para que este sistema, como primera versión de pruebas, pueda ser adaptado en adelante a las empresas interesadas. Si se hubiera especializado demasiado en un caso concreto, dispondríamos de un diseño difícilmente adaptable a las necesidades de las diferentes empresas del sector. Por este motivo, el diseño de la base de datos sólo contiene aquella información que se consideró común para todas las empresas con una flota de vehículos de transporte.
- La aplicación *Servidor* consta de los suficientes niveles de abstracción para poder cambiar de base de datos sin tener que modificar toda la aplicación entera. Además, se ha procurado facilitar la extensibilidad y la modularidad de la aplicación, con vistas a futuras implementaciones para funcionalidades adicionales.
- El interfaz gráfico de la aplicación *Cliente* ha sido diseñado pensando en la comodidad y sencillez de cara al usuario, sin descuidar toda la funcionalidad que debía ofrecer. Además, en este diseño se ha

agrupado toda la información en pestañas con vistas a que, en un futuro, dependiendo del tipo de usuario, éste podrá acceder a una funcionalidad u otra.

- El modo de comunicarse entre los *Clientes* y el *Servidor* ha sido mediante servicios Web, los cuales están basados en estándares abiertos y utilizables con otras herramientas de desarrollo.
- En la implementación, tanto del *Cliente* como del *Servidor*, se ha procurado disminuir, en la medida de lo posible, la carga en la red, con vistas a que este *Sistema* se utilice vía Internet, donde es importante la cantidad de datos a enviar/recibir.
- Para la representación de ubicaciones y rutas se ha utilizado Google Earth. Esta herramienta es en la actualidad una de las más potentes, ofreciendo una enorme cantidad de servicios y funciones. Google Earth posee en la actualidad una de las mejores bases de datos de cartografía y de imágenes vía satélite.
- Durante el desarrollo del *Sistema* se han utilizado únicamente herramientas gratuitas. Esto implica que será posible desarrollar nuevas funcionalidades sin necesidad de tener que gastar en licencias.

2.5.3 Ventajas del Sistema desarrollado.

Como consecuencia de los objetivos mencionados en el apartado anterior, podemos mencionar algunas de las ventajas que tiene este *Sistema*:

- Será fácilmente ampliable la cantidad de servicios ofrecidos.
- Ofrece una solución genérica al problema de localización de vehículos que podrá ser modificada para resolver problemas más específicos.
- Consta de un interfaz gráfico funcional, cómodo y sencillo.
- Para la comunicación se han usado servicios Web, basado en protocolos y estándares abiertos, que permite compatibilidad con otros lenguajes y entornos de desarrollo, además de presentar menos problemas ante firewalls.
- La representación de puntos y rutas se ha implementado para usar Google Earth que posee una base de datos muy completa de todo el planeta.
- Basado en tecnologías de acceso gratuito, y cuyas herramientas de desarrollo básicas también lo son.

2.5.4 Diagrama del Sistema Global

A continuación se mostrará un diagrama para visualizar la estructura global del sistema y ayudar a su comprensión.

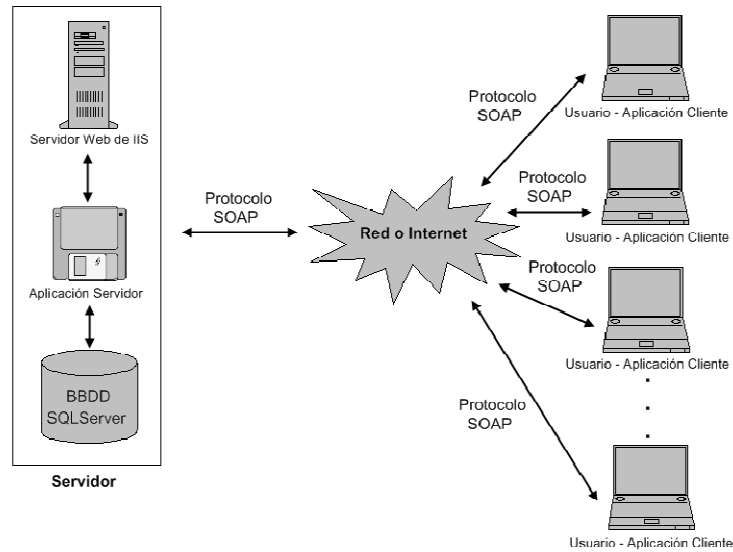


Ilustración 8. Diagrama básico del sistema global.

2.5.5 Descripción de la integración de los diferentes componentes principales

Durante los apartados anteriores, dentro de este capítulo, se han ido describiendo cada uno de los elementos que componen el *Sistema* global. Una vez comprendidos todos los componentes, es momento de explicar la forma en que se integran todos ellos.

Dentro del *Servidor* nos encontramos con tres de los cuatro componentes fundamentales del *Sistema* (*BBDD*, *Aplicación Servidor* y *Servidor Web*). La aplicación *Servidor* se comunica con la base de datos gracias a la utilización de ADO.NET (conjunto de clases que exponen servicios de acceso a datos para el programador de .NET) en su código, mientras que, gracias a los servicios Web de Microsoft, se pueden realizar llamadas a la aplicación *Servidor* desde la red o Internet.

Por último, la forma de comunicarse cada una de las aplicaciones *Cliente* con la aplicación *Servidor* es mediante servicios Web, basado en protocolos (dentro de los cuales se encuentra SOAP basado en XML) y estándares abiertos. La aplicación *Servidor* se encontrará en un servidor Web que recibirá las peticiones y enviará la información necesaria a los diferentes *Clientes*.

Capítulo 3

Manual del usuario

3.1 Introducción al manual de usuario.

La finalidad de este capítulo será proporcionar al usuario la documentación necesaria para la utilización del sistema. Se incluirán instrucciones para la instalación, configuración y ejecución del software necesario tanto para el *Servidor* como para el *Cliente*, así como el manual del interfaz gráfico del *Cliente*.

El manual del usuario está dividido en los siguientes apartados:

- Requisitos del *Sistema de Localización de Camiones*.
- Instalación y configuración de .NET Framework 2.0.
- Instalación y configuración de los requisitos específicos del *Servidor*.
- Instalación y configuración de la aplicación *Servidor*.
- Instalación y configuración de la aplicación *Cliente*.
- Manual de usuario del *Cliente*.
 - Ejecución y entorno principal de la aplicación *Cliente*.
 - Descripción del entorno gráfico principal del *Cliente*.
 - Descripción de la información representada en cada pestaña.
 - Descripción de los formularios disponibles.
 - Descripción de las herramientas para mostrar puntos y rutas en Google Earth.

3.2 Requisitos del Sistema de Localización de Camiones.

Debido a que el *Sistema* está compuesto por tres elementos principales, cada uno de ellos tendrá unos requisitos para su correcto funcionamiento.

Tanto la base de datos como la aplicación *Servidor* se encontrarán en una computadora situada en la red, mientras que las aplicaciones *Cliente* se encontrarán en las computadoras de los usuarios. Todos estos *Clientes* se conectarán con el *Servidor* utilizando SOAP.

3.3 Requisitos del Servidor.

En la computadora que funciona como *Servidor* se necesitará tener instalado el siguiente software:

- Microsoft SQL Server o Microsoft SQL Server Express para la base de datos.
- El entorno de ejecución .NET Framework 2.0 para la aplicación *Servidor*.

- El servidor Web de IIS (Internet Information Services) de Microsoft Windows.

3.4 Requisitos de los Clientes.

Cualquier usuario que desee utilizar el *Cliente* del sistema necesitará tener instalado el entorno de ejecución .NET Framework 2.0.

3.5 Instalación y configuración de .NET Framework 2.0.

El único requisito común al *Servidor* y a los *Clientes* será tener instalado .NET Framework 2.0, el cual se puede descargar gratuitamente por Internet desde Microsoft.

Una vez instalado .NET Framework 2.0 se podrán ejecutar tanto la aplicación *Cliente* como la aplicación *Servidor*.

3.6 Instalación y configuración de los elementos específicos del Servidor.

Como ya se ha comentado en los requisitos del *Servidor*, además de necesitar .NET Framework 2.0, se necesita tener instalado la base de datos y el servidor Web. En los próximos subapartados se explicará detalladamente cómo instalar y configurar estos componentes en el *Servidor*.

En el caso de la base de datos, se darán instrucciones para su instalación y configuración. Esto sólo se hará para la base de datos usada ("Microsoft SQL Server Express") y no para otra.

3.6.1 Instalación y configuración de Microsoft SQL Server Express.

Esta base de datos se puede descargar gratuitamente por Internet desde Microsoft. Su instalación es sencilla y tan sólo hay que seguir los pasos explicados durante la instalación.

Una vez descargada e instalada la base de datos con las opciones elegidas por el administrador del sistema se procederá a instalar la herramienta SQL Server Management Studio Express para la gestión y la configuración de la base de datos. Ésta herramienta se puede obtener gratuitamente por Internet desde Microsoft.

Con SQL Server Management Studio Express instalado, se procederá a añadir la base de datos del *Sistema de Localización de Camiones* y a configurar los permisos de acceso para el correcto funcionamiento.

Tras entrar en SQL Server Management Studio Express y haberse autenticado, se adjuntará la base de datos del *Sistema de Localización de Camiones*. Para ello, en el "Explorador de soluciones" haremos clic con el botón derecho del ratón y elegiremos la opción "Adjuntar", obteniendo la siguiente ventana:

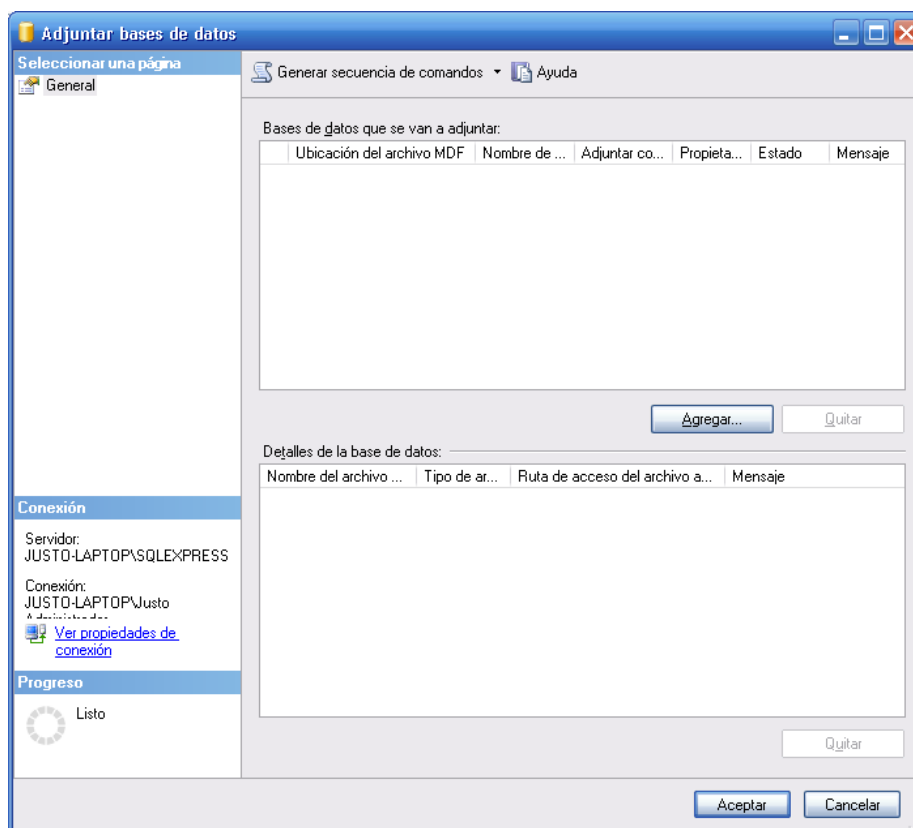


Ilustración 9. Captura de cómo adjuntar una base de datos con SSMSEE.

En esta ventana se elegirá “Agregar...” y se seleccionará el archivo “LocalizacionCamiones.mdf” que contiene la base de datos del *Sistema de Localización de Camiones*.

Una vez hecho esto, restará configurar los permisos de acceso a la base de datos. Para hacer esto, se hace clic con el botón derecho del ratón encima del nombre del servidor de la base de datos en el “Explorar de soluciones” y se elije “Propiedades”. La ventana que aparecerá será la siguiente:

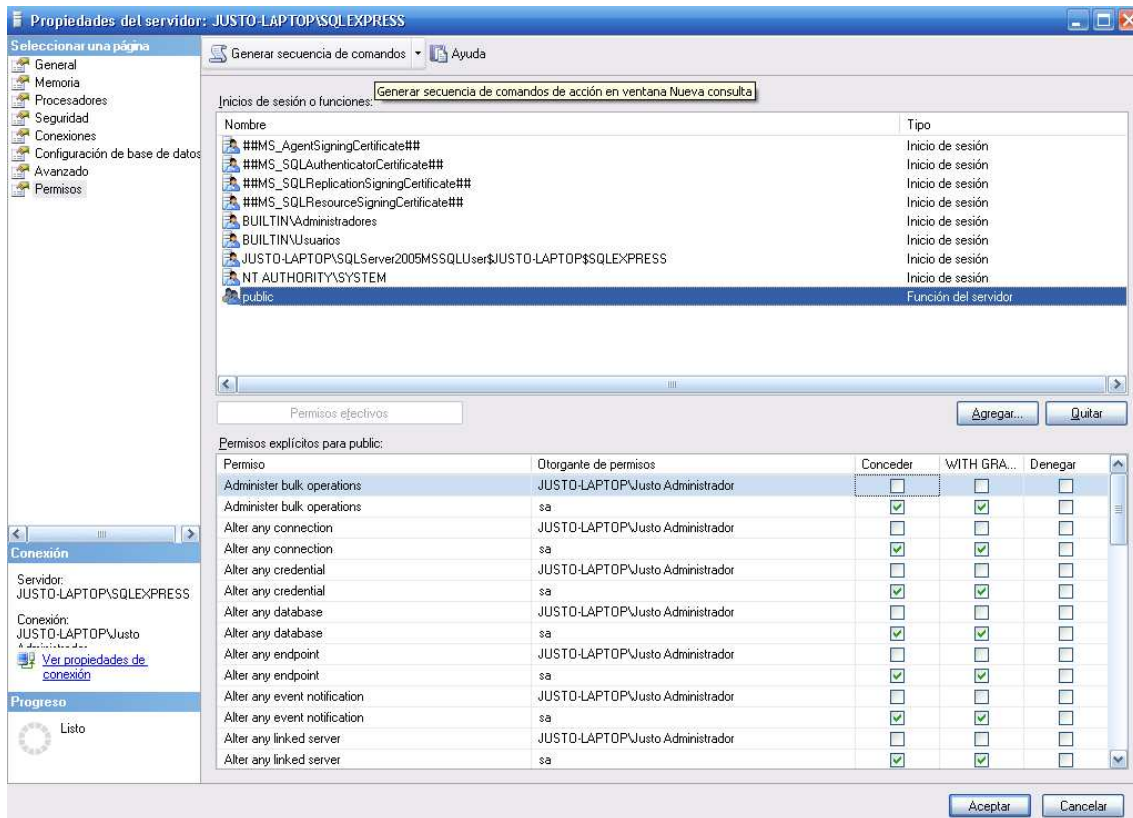


Ilustración 10. Captura de cómo configurar los permisos de acceso a SQL Server.

En “Permisos -> public” se marcarán todas las filas de la columna “WITH GRANT” y se aceptarán los cambios.

Llegado este punto, la base de datos estará correctamente configurada y será accesible por la aplicación *Servidor*.

3.6.2 Instalación y configuración del Servidor Web de IIS de Microsoft.

IIS (Internet Information Services) es una serie de servicios para los ordenadores que funcionan con Windows y a menudo viene incluido en él. Este servicio convierte un ordenador en un servidor de Internet o Intranet. En las computadoras que tienen este servicio instalado se pueden publicar páginas Web tanto local como remotamente (servidor Web).

El servidor Web de IIS se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo Microsoft incluye los de Active Server Pages (ASP) y ASP.NET. También pueden ser incluidos los de otros fabricantes, como PHP o Perl.

Para poder instalar la versión del servidor Web de IIS, necesario para el *Sistema*, hay que tener alguna de las siguientes versiones de Windows:

- IIS 5.0, Windows 2000.
- IIS 5.1, Windows XP Professional.
- IIS 6.0, Windows Server 2003 y Windows XP Professional x64 Edition.

- IIS 7.0, Windows Vista y Windows Server 2008.

Para la instalación del servidor Web se hará desde “Agregar y quitar programas” del “Panel de control”. Desde ahí se accederá a “Agregar o quitar componentes de Windows” y se marcará el elemento mostrado en la siguiente imagen:

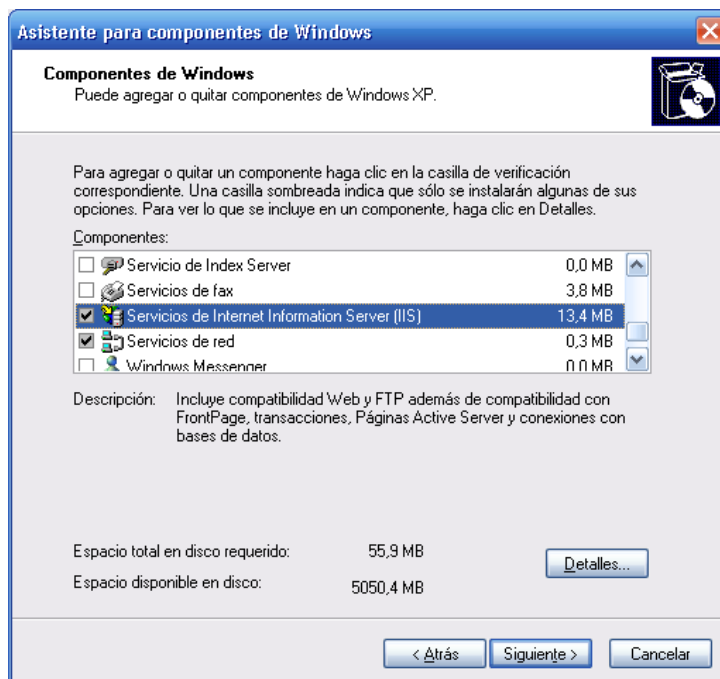


Ilustración 11. Captura de instalación de IIS.

Por defecto, al marcar esta casilla se instalará el servidor Web y el servidor FTP. Dado que el servidor FTP no es necesario para el *Sistema de Localización de Camiones* se podrá evitar su instalación desmarcándolo en los “Detalles...” de este elemento.

3.7 Instalación y configuración de la aplicación Servidor.

Para la instalación de la aplicación *Servidor* habrá que copiar el directorio “WebSite” en la ruta “C:\inetpub\wwwroot” y configurar el archivo “Web.config” que se encuentra en dicho directorio. Los directorios “C:\inetpub” y “C:\inetpub\wwwroot” serán creados automáticamente al instalar el servidor Web que viene con el IIS de Windows.

A continuación es necesario configurar las propiedades del servidor Web para que reconozca el contenido del directorio como una aplicación, ya que en caso contrario no funcionará el servicio Web. Para realizar esto, se hace clic con el botón derecho del ratón encima del icono de “Mi Pc” y se selecciona “Administrar”, obteniendo la siguiente ventana:

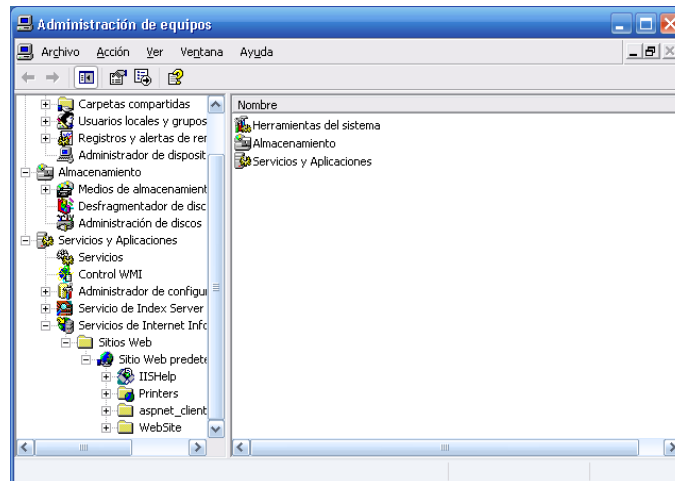


Ilustración 12. Captura “Administración de equipos”.

Dentro de “Servicios de Internet Information Server -> Sitios Web -> Sitio Web predeterminado” se encuentra ahora el directorio que anteriormente se ha copiado en “wwwroot”. Haciendo clic con el botón derecho del ratón y entrando en las “Propiedades” del directorio “WebSite” se mostrará una ventana como la siguiente:

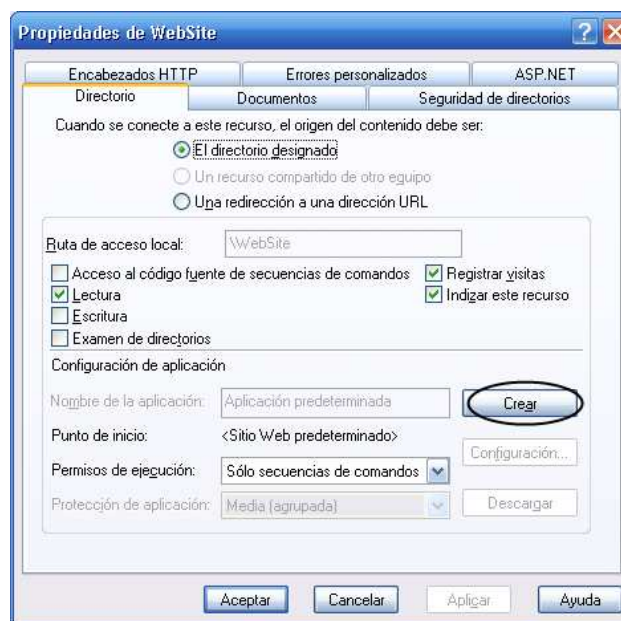


Ilustración 13. Captura de cómo configurar la aplicación en el servidor Web.

En la ventana que aparece se pulsará sobre el botón “Crear”, opción señalada en la imagen anterior. Una vez realizado esto, sin realizar ninguna acción más, se aceptarán los cambios y se saldrá de la ventana de administración de Windows.

Como último paso para la configuración del Servidor, restará modificar el archivo “Web.config” localizado en el directorio “WebSite”. En este archivo se deberá especificar el nombre del servidor SQL y el nombre de la base de datos.

La línea a cambiar en el archivo “Web.config” es la siguiente (mostrada con los campos configurados por defecto):

```
< add key="cnStr" value="server=JUSTO-LAPTOP\SQLEXPRESS;  
database=LocalizacionCamiones; Integrated Security=SSPI" />
```

En esta línea se pasa a la aplicación una serie de parámetros:

- “server”: tras el símbolo “=” se indica el nombre de la computadora y el nombre del servidor SQL, separados por el símbolo “\”.
- “database”: tras el símbolo “=” se especifica el nombre de la base de datos del *Sistema de Localización de Camiones*.
- “Integrated Security”: se especifica el modo de seguridad integrada usada.

3.8 Instalación y configuración de la aplicación Cliente

La aplicación *Cliente* no será necesario instalarla pero, antes de su ejecución, habrá que modificar los parámetros necesarios para que pueda comunicarse con el *Servidor* de forma remota.

Para indicar la dirección del *Servidor* se deberá editar el archivo “LocalizacionCamionesCliente.exe.config” situado en el directorio que contiene los ejecutables del *Cliente*. Dentro de este archivo se modificará el contenido que hay entre los tags “<value> </value>” por la dirección en la que se encuentre el archivo “Service.asmx”, en el *Servidor*.

Si se siguieron los pasos explicados en el apartado referido a la “Instalación y configuración de la aplicación *Servidor*”, el contenido que habrá entre los tags “<value> </value>” deberá tener el siguiente formato:

```
http://“IP_Servidor”/WebSite/Service.asmx
```

Una vez realizado esto, y si se ha configurado todo correctamente, el *Cliente* al ejecutarse se conectará con la base de datos ubicada en el *Servidor*.

3.9 Manual del usuario del Cliente

3.9.1 Introducción

Una vez se han configurado todos los componentes del *Sistema de Localización de Camiones*, cada uno de los usuarios puede pasar a ejecutar el *Cliente*.

Si alguna de las configuraciones no es correcta o no existe comunicación con el *Servidor* no se podrá conectar y se irá obteniendo un mensaje de error por cada una de las conexiones fallidas que ocurran.

En caso de que aparezcan problemas obteniendo los datos del *Servidor*, se deberá revisar más detenidamente la configuración realizada en busca de posibles errores.

El interfaz de usuario para el *Cliente* ha sido diseñado para ser intuitivo, fácil y claro. Su distribución está basada en la representación de los datos agrupados en pestañas. Cada pestaña agrupa datos asociados directamente y tiene una serie de formularios para realizar la creación, modificación, eliminación y filtrado de datos de forma sencilla visualmente.

A continuación se irán describiendo cada uno de los elementos del interfaz, así como los diferentes pasos necesarios para realizar cada una de las posibles tareas.

3.9.2 Ejecución y entorno principal de la aplicación Cliente.

En el directorio de la aplicación *Cliente* el ejecutable es el archivo "LocalizacionCamionesCliente.exe". Tras la ejecución de la aplicación *Cliente* se mostrará una ventana como la siguiente:

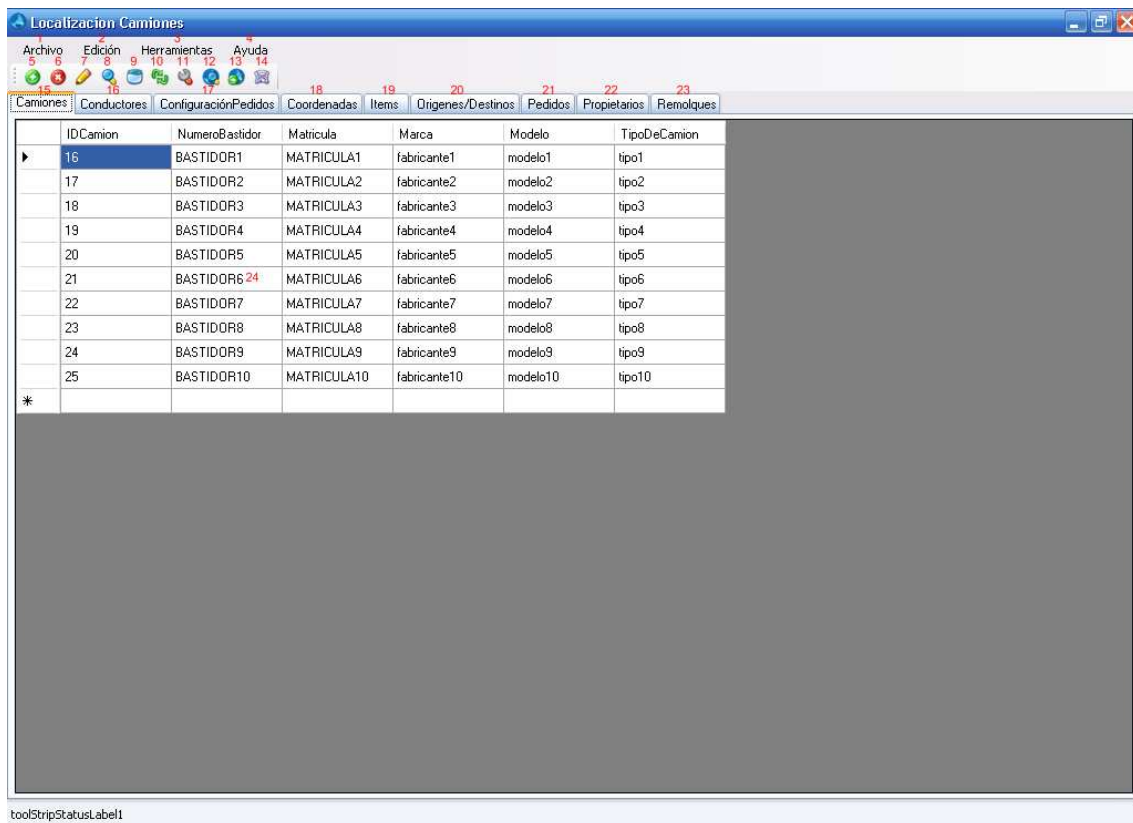


Ilustración 14. Captura entorno principal de la aplicación *Cliente*.

3.9.3 Descripción del entorno gráfico principal

Los elementos presentes en la ventana principal de la aplicación *Cliente* son los siguientes:

1. Archivo. Muestra un menú desplegable con las siguientes opciones:
 - a. Guardar tabla actual en formato XML: Permite los datos mostrados en la pestaña actual en fichero XML.
 - b. Salir: Permite salir de la aplicación.

2. Edición. Muestra un menú desplegable con las siguientes opciones:
 - a. Nuevo: Muestra un formulario para crear una nueva entrada en la tabla mostrada en la pestaña activa en ese momento.
 - b. Eliminar: Muestra un formulario para elegir la entrada a eliminar de la tabla mostrada en la pestaña activa en ese momento.
 - c. Modificar: Muestra un formulario para modificar la entrada elegida de la tabla mostrada en la pestaña activa en ese momento.
 - d. Filtrar: Muestra un formulario para elegir la columna y la cadena a buscar en la tabla mostrada en la pestaña activa en ese momento.
 - e. Limpiar Filtro: Elimina la tabla secundaria que aparece al aplicar un filtro en la pestaña activa en ese momento.
 - f. Actualizar Tabla: Actualiza los datos mostrados en la tabla activa en ese momento.
 - g. Configurar Pedido: Independientemente de la pestaña que se encuentre activa en ese momento, aparecerá un formulario que permitirá asociar a un pedido un contexto. Un contexto se interpreta como un viaje, el cual tendrá asociado un conductor, un camión, un remolque y un ítem. Si se desea modificar alguno de los contextos, estando con la pestaña "ConfiguraciónPedidos" activa se deberá acceder al formulario para modificar datos de esa pestaña.
3. Herramientas. Muestra un menú desplegable con las siguientes opciones:
 - a. Ver posición actual de un ítem: Muestra un formulario en el que se selecciona el identificador del ítem del que se quiere saber la posición actual.
 - b. Ver ruta de un pedido: Muestra un formulario en el que se selecciona el identificador del ítem del que se quiere ver la ruta. Esta ruta será la que ha seguido desde que salió de su origen, cursando el último contexto que le fue asignado.
4. Ayuda: Muestra un menú desplegable con las siguientes opciones:
 - a. Acerca de...: Muestra los créditos de la aplicación.
5. Botón "Nuevo": Muestra un formulario para crear una nueva entrada en la tabla mostrada en la pestaña activa en ese momento.
6. Botón "Eliminar": Muestra un formulario para elegir la entrada a eliminar de la tabla mostrada en la pestaña activa en ese momento.
7. Botón "Modificar": Muestra un formulario para modificar la entrada elegida de la tabla mostrada en la pestaña activa en ese momento.
8. Botón "Filtrar": Muestra un formulario para elegir la columna y la cadena a buscar en la tabla mostrada en la pestaña activa en ese momento.
9. Botón "Limpiar Filtro": Elimina la tabla secundaria que aparece al aplicar un filtro en la pestaña activa en ese momento.
10. Botón "Actualizar Tabla": Actualiza los datos mostrados en la tabla activa en ese momento.
11. Botón "Configurar Pedido": Independientemente de la pestaña que se encuentre activa en ese momento, aparecerá un formulario que permitirá asociar a un pedido un contexto. Un contexto se interpreta como un viaje,

el cual tendrá asociado un conductor, un camión, un remolque y un ítem. Si se desea modificar alguno de los contextos, estando con la pestaña “ConfiguraciónPedidos” activa se deberá acceder al formulario para modificar datos de esa pestaña.

12. Botón “Ver posición actual de un ítem”: Muestra un formulario en el que se selecciona el identificador del ítem del que se quiere saber la posición actual.
13. Botón “Ver ruta de un pedido”: Muestra un formulario en el que se selecciona el identificador del ítem del que se quiere ver la ruta. Esta ruta será la que ha seguido desde que salió de su origen, cursando el último contexto que le fue asignado.
14. Botón “Guardar”: Independientemente de la pestaña que se encuentre activa en ese momento, aparecerá un formulario que permitirá asociar a un pedido un contexto. Un contexto se interpreta como un viaje, el cual tendrá asociado un conductor, un camión, un remolque y un ítem. Si se desea modificar alguno de los contextos, estando con la pestaña “ConfiguraciónPedidos” activa se deberá acceder al formulario para modificar datos de esa pestaña.
15. Pestaña “Camiones”: Muestra la información referente a los diferentes camiones registrados en la base de datos.
16. Pestaña “Conductores”: Muestra la información referente a los diferentes conductores registrados en la base de datos.
17. Pestaña “ConfiguraciónPedidos”: Muestra información referente a los pedidos, así como la asignación que se le ha realizado.
18. Pestaña “Coordenadas”: Muestra la información referente a las coordenadas de cada uno de los ítems registrados en la base de datos.
19. Pestaña “Ítems”: Muestra la información referente a los diferentes dispositivos de localización registrados en la base de datos.
20. Pestaña “Orígenes/Destinos”: Muestra información sobre los diferentes orígenes/destinos registrados en la base de datos. Normalmente serán almacenes de proveedores y de clientes.
21. Pestaña “Pedidos”: Muestra la información referente a los pedidos registrados en la base de datos. Una vez realizado un pedido, éste deberá ser configurado.
22. Pestaña “Propietarios”: Muestra la información referente a los propietarios de los diferentes orígenes/destinos registrados en la base de datos.
23. Pestaña “Remolques”: Muestra la información referente a los diferentes remolques registrados en la base de datos.
24. Panel presente en todas las pestañas, en el que se mostrarán tablas para representar la información relacionada con cada una de dichas pestañas.

3.9.4 Descripción de la información representada en cada pestaña.

Dado que cada pestaña contendrá una tabla con una serie de campos cuyos nombres estarán abreviados, se explicará resumidamente qué es y qué representa cada uno de estos campos.

1. Descripción pestaña “Camiones”. En la siguiente imagen se pueden ver los campos mostrados en esta pestaña:

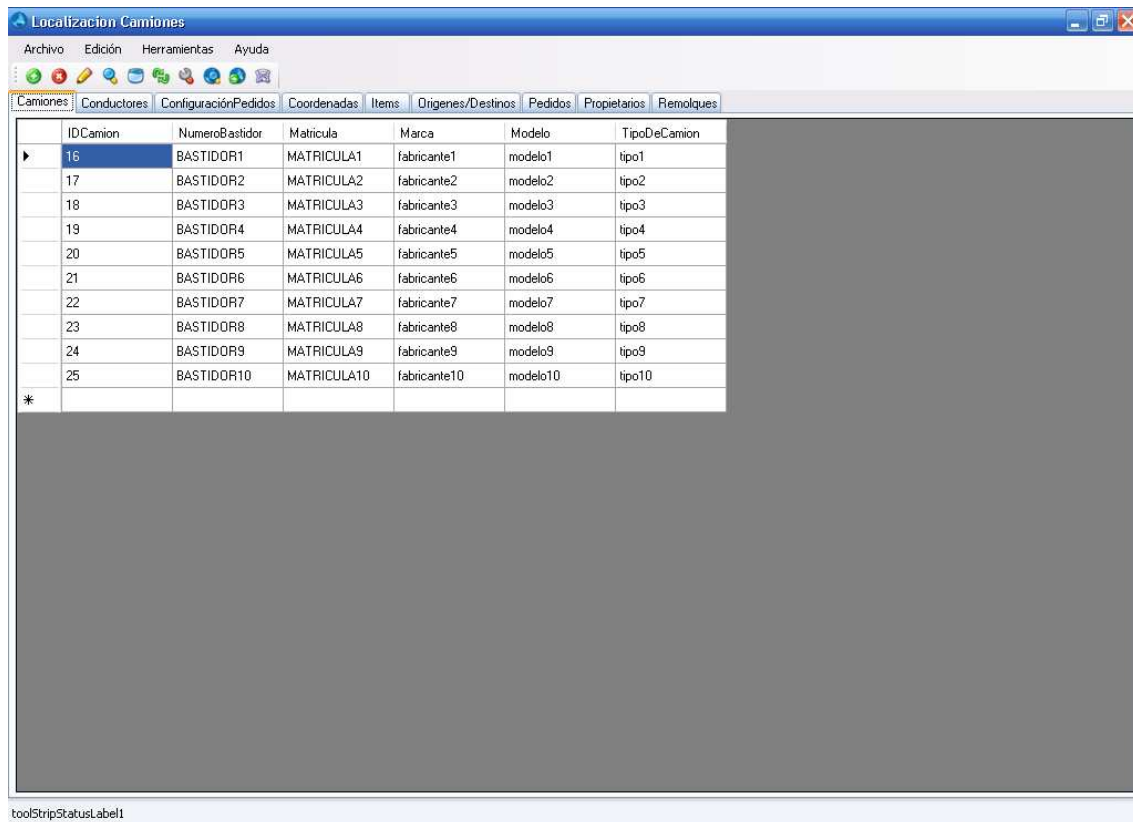


Ilustración 15. Captura de la pestaña “Camiones”.

- a. Campo “IDCamion”: Número entero que representa el identificador que es asignado por la base de datos a cada uno de los camiones registrados.
 - b. Campo “NumeroBastidor”: Cadena con el número de bastidor del camión.
 - c. Campo “Matricula”: Cadena con la matrícula del camión.
 - d. Campo “Marca”: Cadena con el fabricante del camión.
 - e. Campo “Modelo”: Cadena con el modelo de camión.
 - f. Campo “Tipo”: Cadena con el tipo de camión del que se trata.
2. Descripción pestaña “Conductores”. En la siguiente imagen se pueden ver los campos mostrados en esta pestaña:

IDConductor	DNI	Nombre	1Apellido	2Apellido	PermisoDeConducir
15	dni1	nombre1	primerapellido1	segundoapellido1	permiso1
16	dni2	nombre2	primerapellido2	segundoapellido2	permiso2
17	dni3	nombre3	primerapellido3	segundoapellido3	permiso3
18	dni4	nombre4	primerapellido4	segundoapellido4	permiso4
19	dni5	nombre5	primerapellido5	segundoapellido5	permiso5
20	dni6	nombre6	primerapellido6	segundoapellido6	permiso6
21	dni7	nombre7	primerapellido7	segundoapellido7	permiso7
22	dni8	nombre8	primerapellido8	segundoapellido8	permiso8
23	dni9	nombre9	primerapellido9	segundoapellido9	permiso9
24	dni10	nombre10	primerapellido10	segundoapellido10	permiso10
*					

Ilustración 16. Captura de la pestaña “Conductores”.

- a. Campo “IDConductor”: Número entero que representa el identificador que es asignado por la base de datos a cada uno de los conductores registrados.
 - b. Campo “DNI”: Cadena con el DNI del conductor.
 - c. Campo “Nombre”: Cadena con el nombre del conductor.
 - d. Campo “1Apellido”: Cadena con el primer apellido del conductor.
 - e. Campo “2Apellido”: Cadena con el segundo apellido del conductor.
 - f. Campo “PermisoDeConducir”: Cadena con el permiso de conducir del conductor.
3. Descripción pestaña “ConfiguraciónPedidos”. En la siguiente imagen se pueden ver los campos mostrados en esta pestaña:

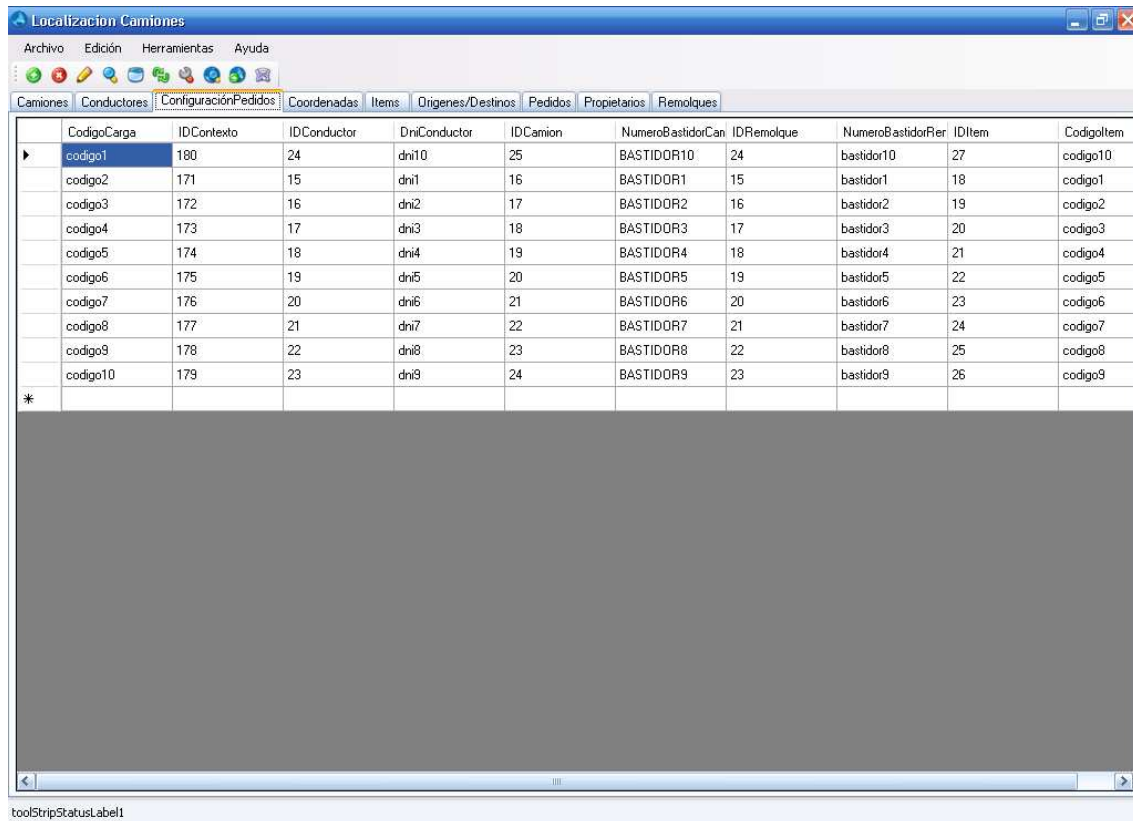


Ilustración 17. Captura de la pestaña “ConfiguraciónPedidos”.

- a. Campo “CodigoCarga”: Cadena con el código que permite identificar una carga. Se debe de poner un código diferente a la carga.
 - b. Campo “IDContexto”: Número entero con el identificador del contexto (o viaje) al que se ha asociado el pedido.
 - c. Campo “IDConductor”: Número entero con el identificador del conductor que cursa o cursará el pedido.
 - d. Campo “DniConductor”: Cadena con el DNI del conductor que cursa o cursará el pedido.
 - e. Campo “IDCamion”: Número entero con el identificador del camión que cursa o cursará el pedido.
 - f. Campo “NumeroBastidorCamion”: Cadena con el número de bastidor del camión que cursa o cursará el pedido.
 - g. Campo “IDRemolque”: Número entero con el identificador del remolque que cursa o cursará el pedido.
 - h. Campo “NumeroBastidorRemolque”: Cadena con el número de bastidor del remolque que cursa o cursará el pedido.
 - i. Campo “IDItem”: Número entero con el identificador del ítem que permitirá localizar al pedido.
 - j. Campo “CodigoItem”: Cadena con el código del ítem que permitirá localizar al pedido.
4. Descripción pestaña “Coordenadas”. En la siguiente imagen se pueden ver los campos mostrados en esta pestaña:

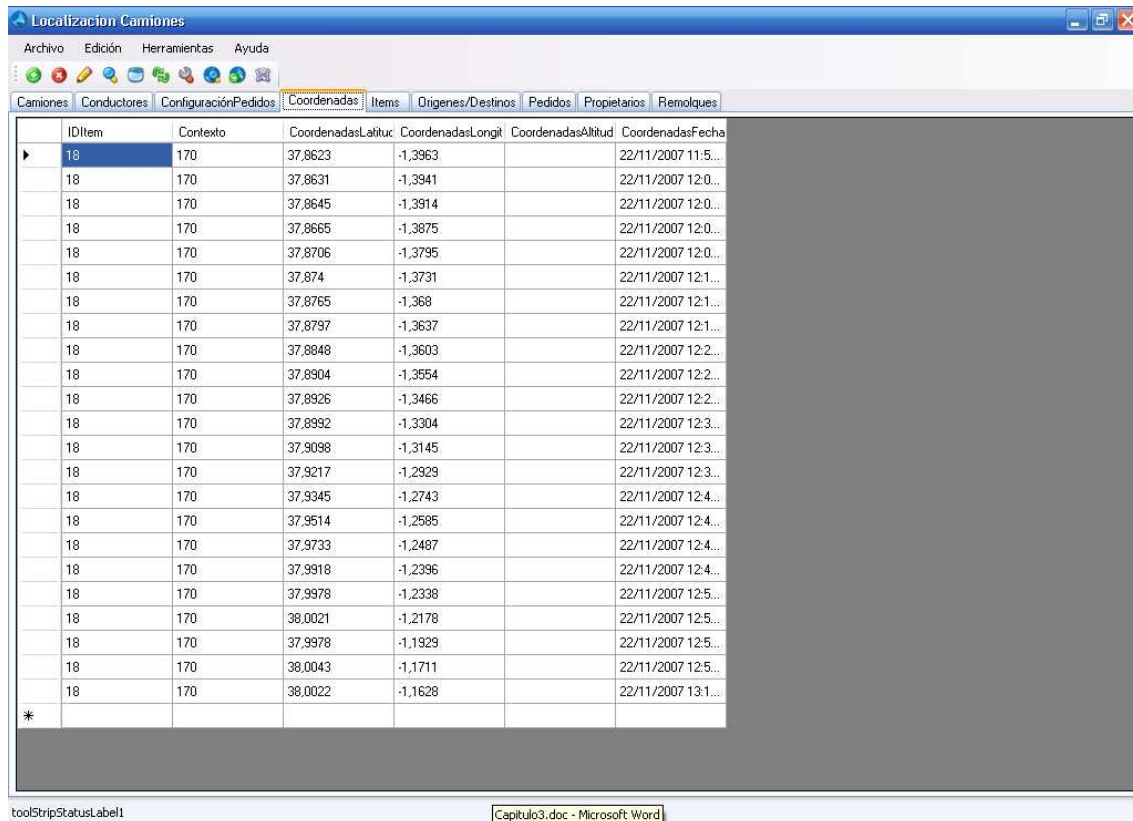


Ilustración 18. Captura de la pestaña “Coordenadas”.

- a. Campo “IDItem”: Número entero con el identificador del ítem al que pertenecen las coordenadas.
- b. Campo “Contexto”: Número entero con el identificador del contexto asociado al ítem.
- c. Campo “CoordenadasLatitud”: Latitud de las coordenadas.
- d. Campo “CoordenadasLongitud”: Longitud de las coordenadas.
- e. Campo “CoordenadasAltitud”: Altitud de las coordenadas.
- f. Campo “CoordenadasFecha”: Fecha y hora a la que fueron registradas las coordenadas.

5. Descripción pestaña “Items”. En la siguiente imagen se pueden ver los campos mostrados en esta pestaña:

IDItem	Codigo	Fabricante	Modelo	VersionFirmware	FechaCompra	ContextoActual
18	codigo1	fabricante1	modelo1	firmware1		171
19	codigo2	fabricante2	modelo2	firmware2		172
20	codigo3	fabricante3	modelo3	firmware3		173
21	codigo4	fabricante4	modelo4	firmware4		174
22	codigo5	fabricante5	modelo5	firmware5		175
23	codigo6	fabricante6	modelo6	firmware6	06/06/1996	176
24	codigo7	fabricante7	modelo7	firmware7	07/07/1997	177
25	codigo8	fabricante8	modelo8	firmware8	08/08/1998	178
26	codigo9	fabricante9	modelo9	firmware9	09/09/1999	179
27	codigo10	fabricante10	modelo10	firmware10	10/10/2000	180
*						

Ilustración 19. Captura de la pestaña “Items”.

- a. Campo “IDItem”: Número entero con el identificador asociado por la base de datos al ítem.
 - b. Campo “Codigo”: Cadena con el código asociado al ítem. Podrá ser desde un número de serie a un código representativo arbitrario.
 - c. Campo “Fabricante”: Cadena con el fabricante del ítem o dispositivo de localización.
 - d. Campo “Modelo”: Modelo del ítem.
 - e. Campo “VersionFirmware”: Cadena con el firmware y su versión.
 - f. Campo “FechaCompra”: Fecha de la compra o del registro del ítem.
 - g. Campo “ContextoActual”: Número entero con el identificador del contexto que tiene asociado actualmente el ítem.
6. Descripción pestaña “Origenes/Destinos”. En la siguiente imagen se pueden ver los campos mostrados en esta pestaña:

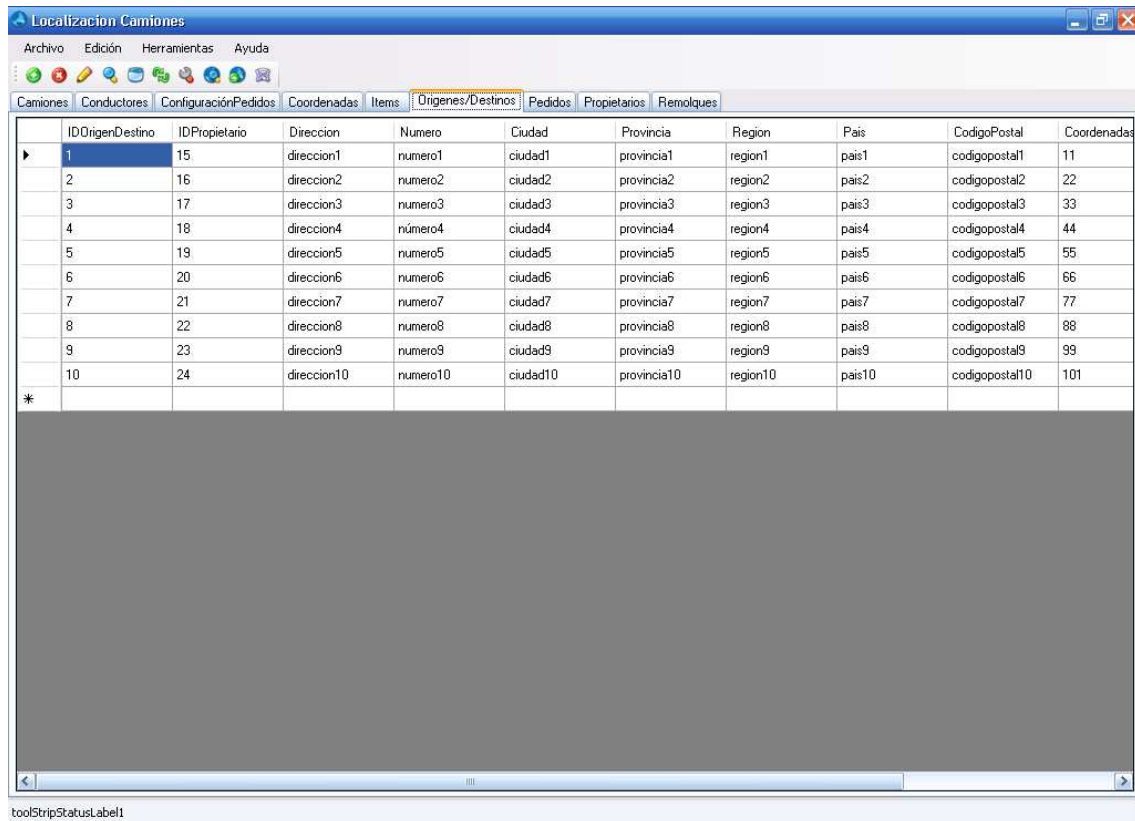


Ilustración 20. Captura de la pestaña “OrigenesDestinos”.

- a. Campo “IDOrigenDestino”: Número entero con el identificador del origen/destino.
- b. Campo “IDPropietario”: Número entero con el identificador del propietario al que pertenece el origen/destino.
- c. Campo “Direccion”: Cadena con la dirección del origen/destino.
- d. Campo “Numero”: Cadena con el número del domicilio.
- e. Campo “Ciudad”: Cadena con la ciudad a la que pertenece el origen/destino.
- f. Campo “Provincia”: Cadena con la provincia a la que pertenece el origen/destino.
- g. Campo “Region”: Cadena con la región a la que pertenece el origen/destino.
- h. Campo “Pais”: Cadena con el país al que pertenece el origen/destino.
- i. Campo “CodigoPostal”: Cadena con el código postal del origen/destino.
- j. Campo “CoordenadasLatitud”: Latitud de las coordenadas en las que se encuentra el origen/destino.
- k. Campo “CoordenadasLongitud”: Longitud de las coordenadas en las que se encuentra el origen/destino.
- l. Campo “CoordenadasAltitud”: Altitud de las coordenadas en las que se encuentra el origen/destino.

7. Descripción pestaña “Pedidos”. En la siguiente imagen se pueden ver los campos mostrados en esta pestaña:

IDCarga	IDContexto	IDItem	CodigoItem	CodigoCarga	TipoCarga	ClaseCarga	PesoCarga	IDOrigen	CiudadOrigen
31	180	27	codigo10	codigo1	tipo1	clase1	11	1	ciudad1
32	171	18	codigo1	codigo2	tipo2	clase2	22	2	ciudad2
33	172	19	codigo2	codigo3	tipo3	clase3	33	3	ciudad3
34	173	20	codigo3	codigo4	tipo4	clase4	44	4	ciudad4
35	174	21	codigo4	codigo5	tipo5	clase5	55	5	ciudad5
36	175	22	codigo5	codigo6	tipo6	clase6	66	6	ciudad6
37	176	23	codigo6	codigo7	tipo7	clase7	77	7	ciudad7
38	177	24	codigo7	codigo8	tipo8	clase8	88	8	ciudad8
39	178	25	codigo8	codigo9	tipo9	clase9	99	9	ciudad9
40	179	26	codigo9	codigo10	tipo10	clase10	1010	10	ciudad10
*									

Ilustración 21. Captura de la pestaña “Pedidos”.

- a. Campo “IDCarga”: Número entero con el identificador asociado por la base de datos al pedido.
- b. Campo “IDContexto”: Número entero con el identificador del contexto al que está asociado el pedido.
- c. Campo “IDItem”: Número entero con el identificador del ítem asociado al pedido.
- d. Campo “CodigoItem”: Cadena con el código del ítem asociado al pedido.
- e. Campo “CodigoCarga”: Cadena con el código identificativo del pedido.
- f. Campo “TipoCarga”: Cadena con el tipo de carga del pedido.
- g. Campo “ClaseCarga”: Clase de carga del pedido. Ejemplo: en el caso de melones de galia, “melones” sería el tipo de carga y “galia” sería la clase de carga.
- h. Campo “PesoCarga”: Peso del pedido.
- i. Campo “IDOrigen”: Identificador del origen de la carga.
- j. Campo “CiudadOrigen”: Ciudad de origen de la carga.
- k. Campo “IDDestino”: Identificador del destino de la carga.
- l. Campo “CiudadDestino”: Ciudad de destino de la carga.
- m. Campo “IDConductor”: Identificador del conductor encargado de entregar el pedido.
- n. Campo “DniConductor”: DNI del conductor encargado de entregar el pedido.
- o. Campo “IDCamion”: Identificador del camión encargado de llevar el pedido.
- p. Campo “NumeroBastidorCamion”: Número de bastidor del camión encargado de llevar el pedido.

- q. Campo "IDRemolque": Identificador del remolque llevado por el camión.
- r. Campo "NumeroBastidorRemolque": Número de bastidor del remolque llevado por el camión.

8. Descripción pestaña "Propietarios". En la siguiente imagen se pueden ver los campos mostrados en esta pestaña:

	IDPropietario	Cif/Dni/Nif	NombreFiscal	NombreComercial	NombrePropietario	Apellido1Propietario	Apellido2Propietario	Telefonos
▶	15	dni2	nombrefiscal2	nombrecomercial2	nombrepropietario2	primerapellido2	segundoapellido2	telefono2
	16	dni3	nombrefiscal3	nombrecomercial3	nombrepropietario3	primerapellido3	segundoapellido3	telefono3
	17	dni4	nombrefiscal4	nombrecomercial4	nombrepropietario4	primerapellido4	segundoapellido4	telefono4
	18	dni5	nombrefiscal5	nombrecomercial5	nombrepropietario5	primerapellido5	segundoapellido5	telefono5
	19	dni6	nombrefiscal6	nombrecomercial6	nombrepropietario6	primerapellido6	segundoapellido6	telefono6
	20	dni7	nombrefiscal7	nombrecomercial7	nombrepropietario7	primerapellido7	segundoapellido7	telefono7
	21	dni8	nombrefiscal8	nombrecomercial8	nombrepropietario8	primerapellido8	segundoapellido8	telefono8
	22	dni9	nombrefiscal9	nombrecomercial9	nombrepropietario9	primerapellido9	segundoapellido9	telefono9
	23	dni10	nombrefiscal10	nombrecomercial10	nombrepropietari...	primerapellido10	segundoapellido10	telefono10
	24	dni11	nombrefiscal11	nombrecomercial11	nombrepropietari...	primerapellido11	segundoapellido11	telefono11
*								

Ilustración 22. Captura de la pestaña "Propietarios".

- a. Campo "IDPropietario": Número entero con el identificador del propietario al que pertenece uno o varios orígenes/destinos (normalmente serán almacenes o fábricas).
- b. Campo "Cif/Dni/Nif": Dependiendo de si el propietario es una persona o una empresa se utilizará uno u otro.
- c. Campo "NombreFiscal": En el caso de que el propietario sea una empresa aquí se introducirá su nombre fiscal.
- d. Campo "NombreComercial": En el caso de que el propietario sea una empresa aquí se introducirá su nombre comercial.
- e. Campo "NombrePropietario": En el caso de que el propietario sea una persona aquí se introducirá su nombre.
- f. Campo "Apellido1Propietario": En el caso de que el propietario sea una persona aquí se introducirá su primer apellido.
- g. Campo "Apellido2Propietario": En el caso de que el propietario sea una persona aquí se introducirá su segundo apellido.
- h. Campo "Telefonos": Teléfono/Teléfonos de contacto.

9. Descripción pestaña “Remolques”. En la siguiente imagen se pueden ver los campos mostrados en esta pestaña:

IDRemolque	NumeroBastidor	Matricula	Marca	Modelo	TipoDeRemolque
15	bastidor1	MATRICULA1	fabricante1	modelo1	tipo1
16	bastidor2	MATRICULA2	fabricante2	modelo2	tipo2
17	bastidor3	MATRICULA3	fabricante3	modelo3	tipo3
18	bastidor4	MATRICULA4	fabricante4	modelo4	tipo4
19	bastidor5	MATRICULA5	fabricante5	modelo5	tipo5
20	bastidor6	MATRICULA6	fabricante6	modelo6	tipo6
21	bastidor7	MATRICULA7	fabricante7	modelo7	tipo7
22	bastidor8	MATRICULA8	fabricante8	modelo8	tipo8
23	bastidor9	MATRICULA9	fabricante9	modelo9	tipo9
24	bastidor10	MATRICULA10	fabricante10	modelo10	tipo10

Ilustración 23. Captura de la pestaña “Remolques”.

- Campo “IDRemolque”: Número entero que representa el identificador que es asignado por la base de datos a cada uno de los remolques registrados.
- Campo “NumeroBastidor”: Cadena con el número de bastidor del remolque.
- Campo “Matricula”: Cadena con la matrícula actual del remolque.
- Campo “Marca”: Cadena con el fabricante del remolque.
- Campo “Modelo”: Cadena con el modelo de remolque.
- Campo “Tipo”: Cadena con el tipo de remolque del que se trata.

3.9.5 Descripción de los formularios disponibles

Como ya se ha comentado anteriormente, en cada una de las pestañas se puede acceder a una serie de formularios para poder crear, modificar, eliminar y filtrar la información mostrada en las tablas.

Los formularios que se mostrarán dependerán de la pestaña que esté activa. Por ejemplo, al estar la pestaña “Camiones” activa, los formularios que se mostrarán al pulsar sobre “Nuevo”, “Eliminar”, “Modificar” o “Filtrar” serán los correspondientes a esta pestaña.

A continuación se describirá brevemente cada uno de los formularios y se explicará qué contenido deberá introducirse en cada uno de los campos disponibles de

cada formulario. Para su descripción se seguirá el orden en el que son mostradas las diferentes pestañas.

1. Formularios disponibles para la pestaña “Camiones”:
 - 1.1. Formulario para crear un nuevo registro: Permite registrar un nuevo camión en la base de datos.

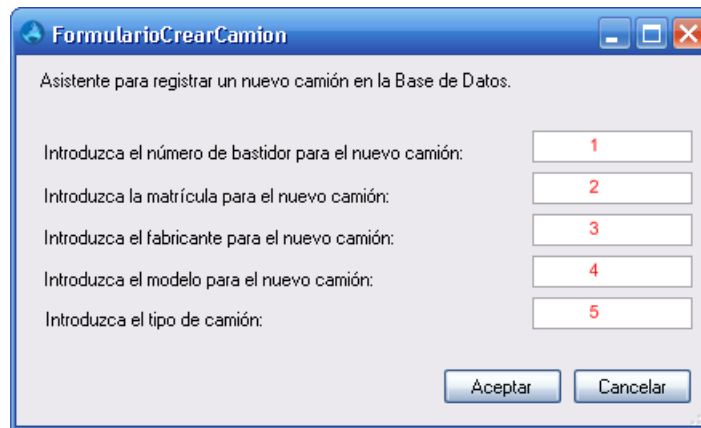


Ilustración 24. Captura del formulario para registrar un camión.

- 1.1.1. Cadena con el número de bastidor del camión. No permite valor nulo.
 - 1.1.2. Cadena con la matrícula del camión. No permite valor nulo.
 - 1.1.3. Cadena con el fabricante del camión. No permite valor nulo.
 - 1.1.4. Cadena con el modelo del camión. No permite valor nulo.
 - 1.1.5. Cadena con el tipo del camión. Sí permite valor nulo.
- 1.2. Formulario para eliminar un registro: Permite eliminar los datos de un camión registrado en la base de datos. Seleccionando la opción deseada, se podrá eliminar el camión por su identificador, por su matrícula o por su número de bastidor.

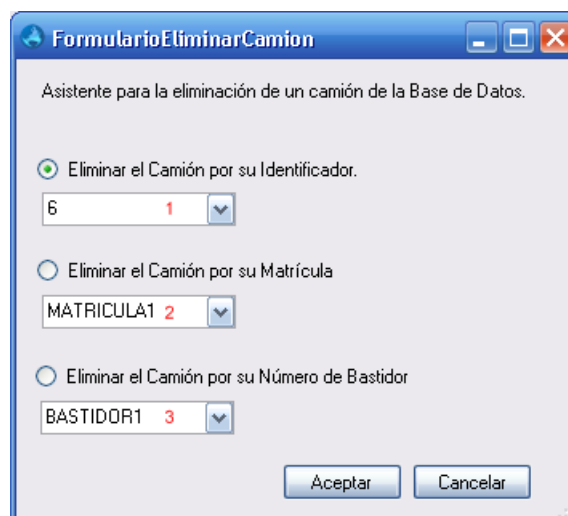


Ilustración 25. Captura del formulario para eliminar un camión.

- 1.2.1. Identificador del camión a eliminar.
 - 1.2.2. Matrícula del camión a eliminar.
 - 1.2.3. Número de bastidor del camión a eliminar.
- 1.3. Formulario para modificar un registro: Permite modificar los datos de un camión registrado en la base de datos.

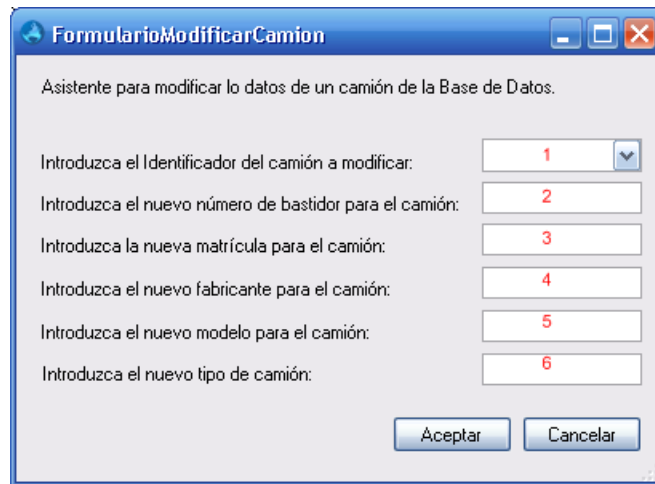


Ilustración 26. Captura del formulario para modificar un camión.

- 1.3.1. Identificador del camión a modificar. No permite valor nulo.
 - 1.3.2. Cadena con el número de bastidor del camión. No permite valor nulo.
 - 1.3.3. Cadena con la matrícula del camión. No permite valor nulo.
 - 1.3.4. Cadena con el fabricante del camión. No permite valor nulo.
 - 1.3.5. Cadena con el modelo del camión. No permite valor nulo.
 - 1.3.6. Cadena con el tipo del camión. Sí permite valor nulo.
- 1.4. Formulario para filtrar datos: Permite filtrar los datos de la tabla.

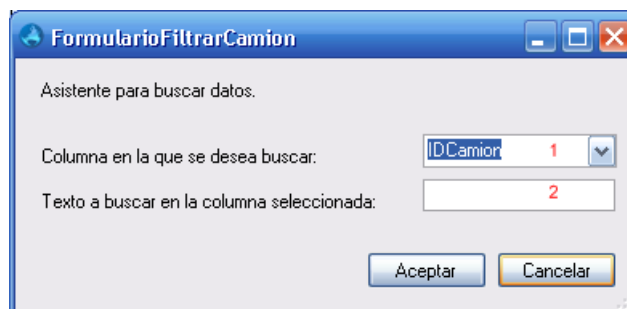


Ilustración 27. Captura del formulario para filtrar los camiones.

- 1.4.1. Nombre de la columna en la que buscar datos.
- 1.4.2. Cadena a buscar en la columna seleccionada en el anterior campo.

2. Formularios disponibles para la pestaña “Conductores”:

2.1. Formulario para crear un nuevo registro: Permite registrar un nuevo conductor en la base de datos.

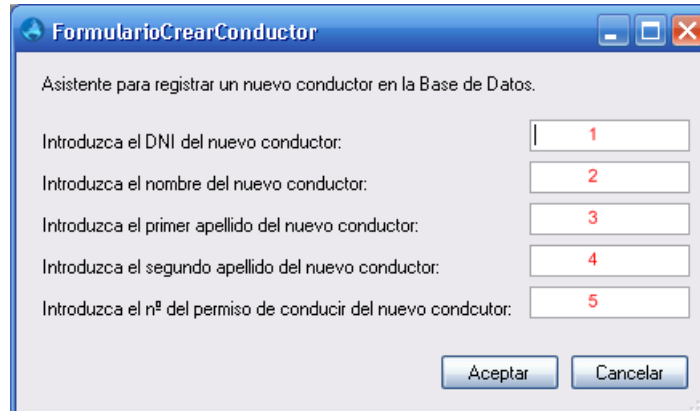


Ilustración 28. Captura del formulario para crear un conductor.

- 2.1.1. Cadena con el DNI del conductor. No permite valor nulo.
- 2.1.2. Cadena con el nombre del conductor. No permite valor nulo.
- 2.1.3. Cadena con el primer apellido del conductor. No permite valor nulo.
- 2.1.4. Cadena con el segundo apellido del conductor. No permite valor nulo.
- 2.1.5. Cadena con el permiso de conducir del conductor. No permite nulos.

2.2. Formulario para eliminar un registro: Permite eliminar los datos de un conductor registrado en la base de datos. Seleccionando la opción deseada, se podrá eliminar el conductor por su identificador o por su DNI.

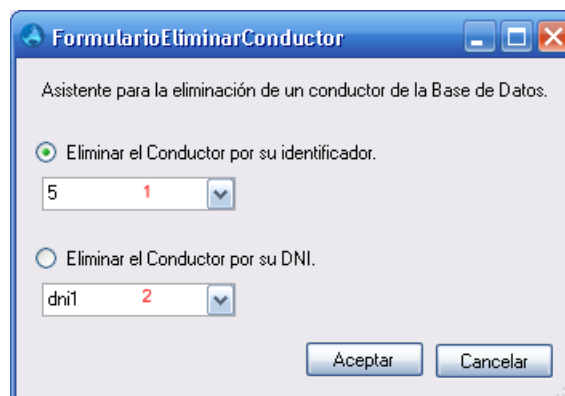


Ilustración 29. Captura del formulario para eliminar un conductor.

- 2.2.1. Identificador del conductor a eliminar.
- 2.2.2. Cadena con el DNI del conductor a eliminar.

2.3. Formulario para modificar un registro: Permite modificar los datos de un conductor registrado en la base de datos.

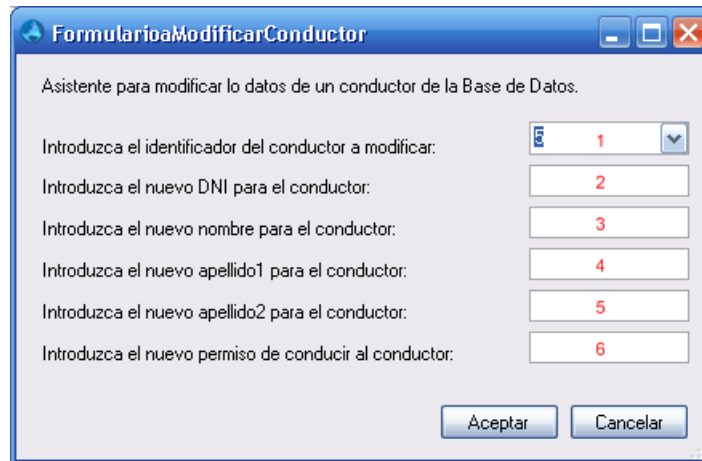


Ilustración 30. Captura del formulario para modificar un conductor.

- 2.3.1. Identificador del conductor a modificar. No permite valor nulo.
- 2.3.2. Cadena con el DNI del conductor. No permite valor nulo.
- 2.3.3. Cadena con el nombre del conductor. No permite valor nulo.
- 2.3.4. Cadena con el primer apellido del conductor. No permite valor nulo.
- 2.3.5. Cadena con el segundo apellido del conductor. No permite valor nulo.
- 2.3.6. Cadena con el permiso de conducir del conductor. No permite nulos.

2.4. Formulario para filtrar datos: Permite filtrar los datos de la tabla.

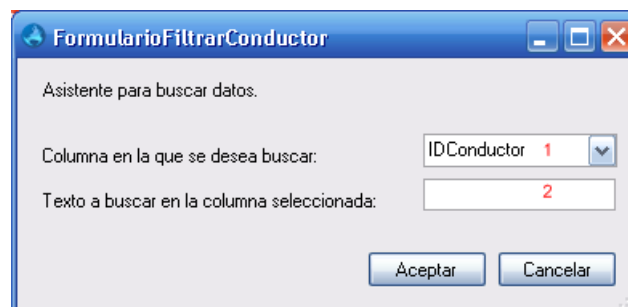


Ilustración 31. Captura del formulario para filtrar conductores.

- 2.4.1. Nombre de la columna en la que buscar datos.
 - 2.4.2. Cadena a buscar en la columna seleccionada en el anterior campo.
3. Formularios disponibles para la pestaña "ConfiguraciónPedidos":
- 3.1. Formulario para crear un nuevo registro: Permite registrar un nuevo contexto o viaje en la base de datos.

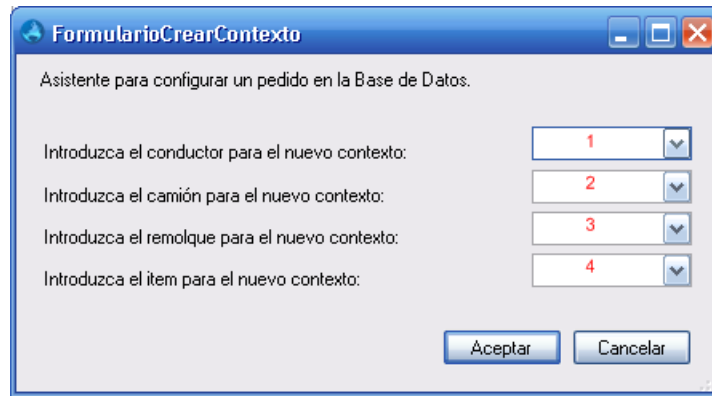


Ilustración 32. Captura del formulario para crear un contexto.

- 3.1.1. Identificador del conductor a asociar al contexto. Sí permite valor nulo.
 - 3.1.2. Identificador del camión a asociar al contexto. Sí permite valor nulo.
 - 3.1.3. Identificador del remolque a asociar al contexto. Sí permite valor nulo.
 - 3.1.4. Identificador del ítem a asociar al contexto. Sí permite valor nulo.
- 3.2. Formulario para eliminar un registro: Permite eliminar los datos de un contexto o viaje registrado en la base de datos.

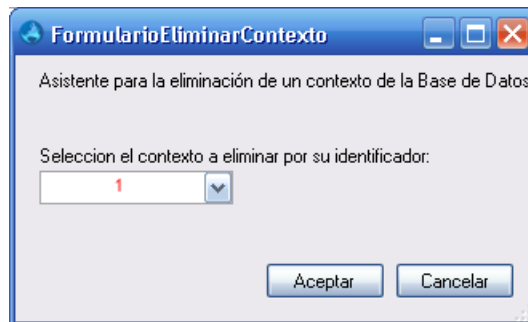


Ilustración 33. Captura del formulario para eliminar un contexto.

- 3.2.1. Identificador del contexto a eliminar. No permite valor nulo.
- 3.3. Formulario para modificar un registro: Permite modificar los datos de un contexto registrado en la base de datos.

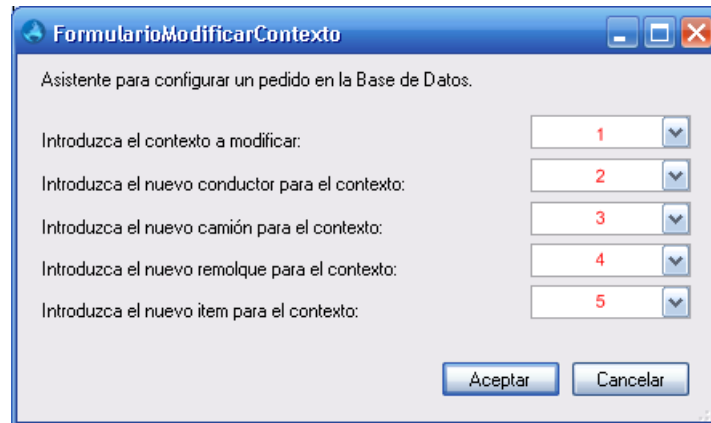


Ilustración 34. Captura del formulario para modificar un contexto.

- 3.3.1. Identificador del contexto a modificar. No permite valor nulo.
- 3.3.2. Identificador del conductor a asociar al contexto. Sí permite valor nulo.
- 3.3.3. Identificador del camión a asociar al contexto. Sí permite valor nulo.
- 3.3.4. Identificador del remolque a asociar al contexto. Sí permite valor nulo.
- 3.3.5. Identificador del ítem a asociar al contexto. Sí permite valor nulo.

3.4. Formulario para filtrar datos: Permite filtrar los datos de la tabla.

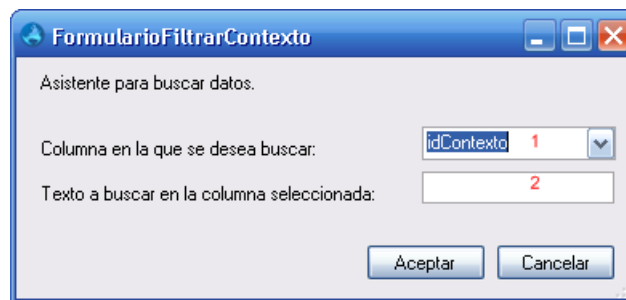


Ilustración 35. Captura del formulario para filtrar contextos.

- 3.4.1. Nombre de la columna en la que buscar datos.
- 3.4.2. Cadena a buscar en la columna seleccionada en el anterior campo.

- 4. Formularios disponibles para la pestaña "Coordenadas":
 - 4.1. Formulario para filtrar datos: Permite filtrar los datos de la tabla.

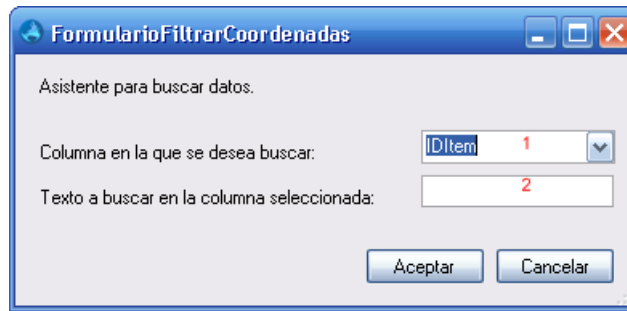


Ilustración 36. Captura del formulario para filtrar coordenadas.

- 4.1.1. Nombre de la columna en la que buscar datos.
 - 4.1.2. Cadena a buscar en la columna seleccionada en el anterior campo.
5. Formularios disponibles para la pestaña "Items":
- 5.1. Formulario para crear un nuevo registro: Permite registrar un nuevo ítem en la base de datos.

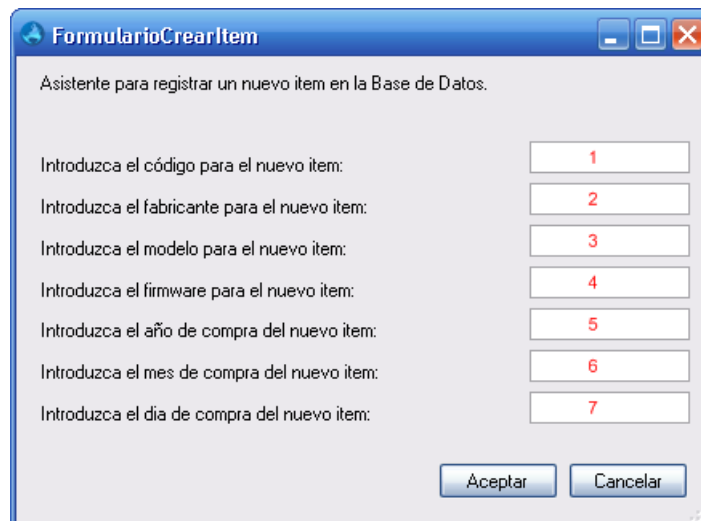


Ilustración 37. Captura del formulario para crear un ítem.

- 5.1.1. Cadena con el código o número de serie del ítem. No permite valor nulo.
 - 5.1.2. Cadena con el fabricante del ítem. No permite valor nulo.
 - 5.1.3. Cadena con el modelo del ítem. No permite valor nulo.
 - 5.1.4. Cadena con el firmware o su versión. Sí permite valor nulo.
 - 5.1.5. Entero con el año de la compra del ítem. Sí permite valor nulo.
 - 5.1.6. Entero con el mes de la compra del ítem. Sí permite valor nulo.
 - 5.1.7. Entero con el día de la compra del ítem. Sí permite valor nulo.
- 5.2. Formulario para eliminar un registro: Permite eliminar los datos de un ítem registrado en la base de datos. Seleccionando la opción deseada, se podrá eliminar el ítem por su identificador o por su código.

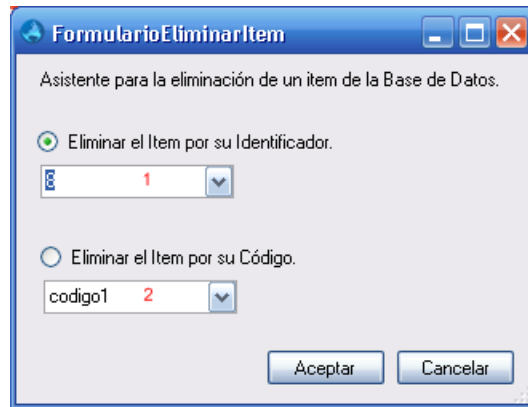


Ilustración 38. Captura del formulario para eliminar un ítem.

- 5.2.1. Identificador del ítem a eliminar.
- 5.2.2. Cadena con el código del ítem a eliminar.
- 5.3. Formulario para modificar un registro: Permite modificar los datos de un ítem registrado en la base de datos.

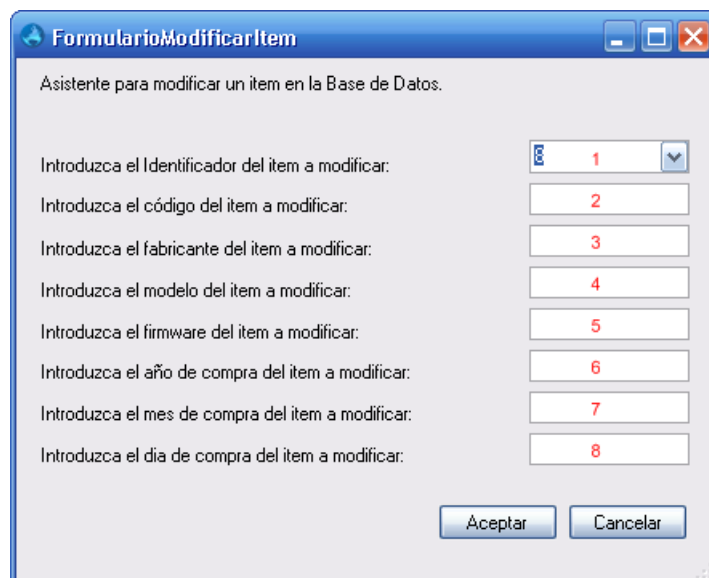


Ilustración 39. Captura del formulario para modificar un ítem.

- 5.3.1. Identificador del ítem a modificar. No permite valor nulo.
 - 5.3.2. Cadena con el código o número de serie del ítem. No permite valor nulo.
 - 5.3.3. Cadena con el fabricante del ítem. No permite valor nulo.
 - 5.3.4. Cadena con el modelo del ítem. No permite valor nulo.
 - 5.3.5. Cadena con el firmware o su versión. Sí permite valor nulo.
 - 5.3.6. Entero con el año de la compra del ítem. Sí permite valor nulo.
 - 5.3.7. Entero con el mes de la compra del ítem. Sí permite valor nulo.
 - 5.3.8. Entero con el día de la compra del ítem. Sí permite valor nulo.
- 5.4. Formulario para filtrar datos: Permite filtrar los datos de la tabla.

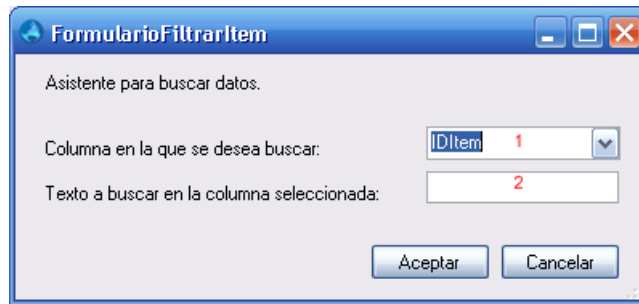


Ilustración 40. Captura del formulario para filtrar ítems.

- 5.4.1. Nombre de la columna en la que buscar datos.
 - 5.4.2. Cadena a buscar en la columna seleccionada en el anterior campo.
6. Formularios disponibles para la pestaña “Orígenes/Destinos”:
- 6.1. Formulario para crear un nuevo registro: Permite registrar un nuevo origen/destino en la base de datos.

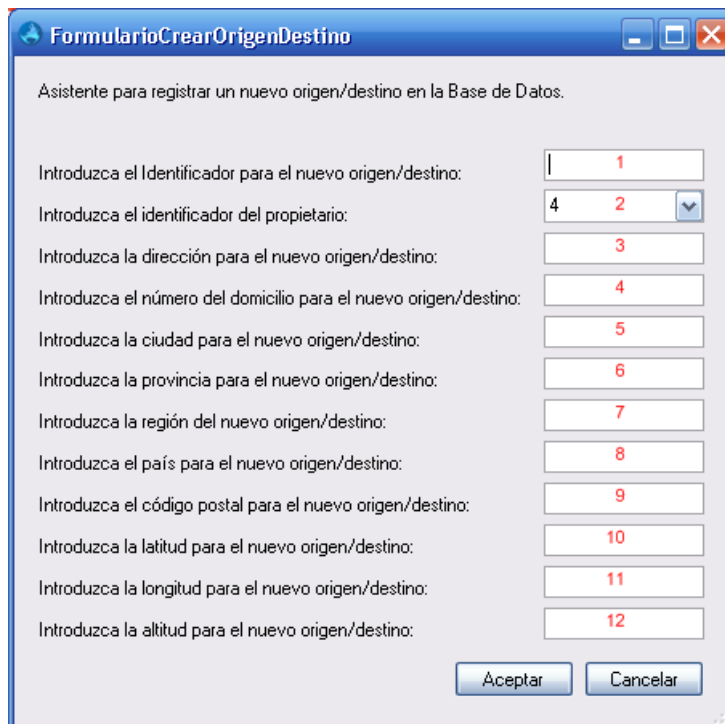


Ilustración 41. Captura del formulario para crear un origen/destino.

- 6.1.1. Identificador del origen/destino. No permite valor nulo.
- 6.1.2. Identificador del propietario del origen/destino. Sí permite valor nulo.
- 6.1.3. Cadena con la dirección del origen/destino. No permite valor nulo.
- 6.1.4. Cadena con el número de la dirección. Sí permite valor nulo.
- 6.1.5. Cadena con la ciudad del origen/destino. No permite valor nulo.
- 6.1.6. Cadena con la provincia del origen/destino. Sí permite valor nulo.
- 6.1.7. Cadena con la región del origen/destino. No permite valor nulo.
- 6.1.8. Cadena con el país del origen/destino. No permite valor nulo.
- 6.1.9. Cadena con el código postal del origen/destino. Sí permite valor nulo.

- 6.1.10. Latitud del origen/destino. Sí permite valor nulo.
- 6.1.11. Longitud del origen/destino. Sí permite valor nulo.
- 6.1.12. Altitud del origen/destino. Sí permite valor nulo.

6.2. Formulario para eliminar un registro: Permite eliminar los datos de un origen/destino registrado en la base de datos.

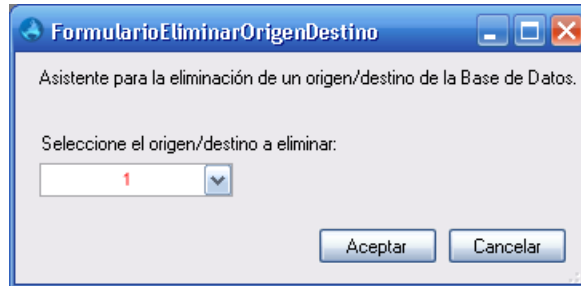


Ilustración 42. Captura del formulario para eliminar un origen/destino.

6.2.1. Identificador del origen/destino a eliminar.

6.3. Formulario para modificar un registro: Permite modificar los datos de un origen/destino registrado en la base de datos.

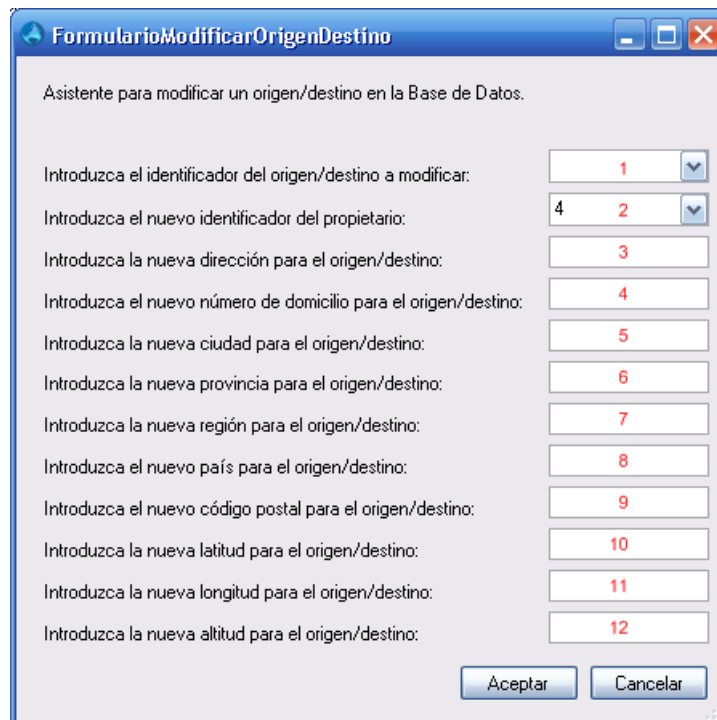


Ilustración 43. Captura del formulario para modificar un origen/destino.

- 6.3.1. Identificador del origen/destino. No permite valor nulo.
- 6.3.2. Identificador del propietario del origen/destino. Sí permite valor nulo.
- 6.3.3. Cadena con la dirección del origen/destino. No permite valor nulo.
- 6.3.4. Cadena con el número de la dirección. Sí permite valor nulo.

- 6.3.5. Cadena con la ciudad del origen/destino. No permite valor nulo.
- 6.3.6. Cadena con la provincia del origen/destino. Sí permite valor nulo.
- 6.3.7. Cadena con la región del origen/destino. No permite valor nulo.
- 6.3.8. Cadena con el país del origen/destino. No permite valor nulo.
- 6.3.9. Cadena con el código postal del origen/destino. Sí permite valor nulo.
- 6.3.10. Latitud del origen/destino. Sí permite valor nulo.
- 6.3.11. Longitud del origen/destino. Sí permite valor nulo.
- 6.3.12. Altitud del origen/destino. Sí permite valor nulo.

6.4. Formulario para filtrar datos: Permite filtrar los datos de la tabla.

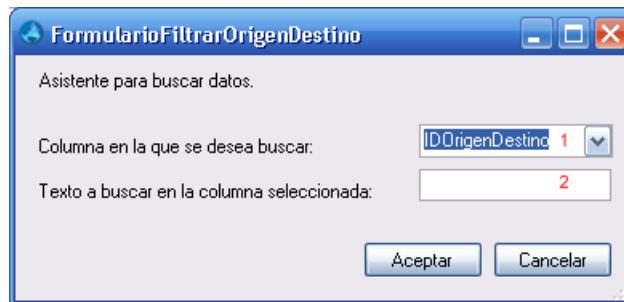


Ilustración 44. Captura del formulario para filtrar orígenes/destinos.

- 6.4.1. Nombre de la columna en la que buscar datos.
- 6.4.2. Cadena a buscar en la columna seleccionada en el anterior campo.

7. Formularios disponibles para la pestaña “Pedidos”:

7.1. Formulario para crear un nuevo registro: Permite registrar un nuevo pedido en la base de datos.

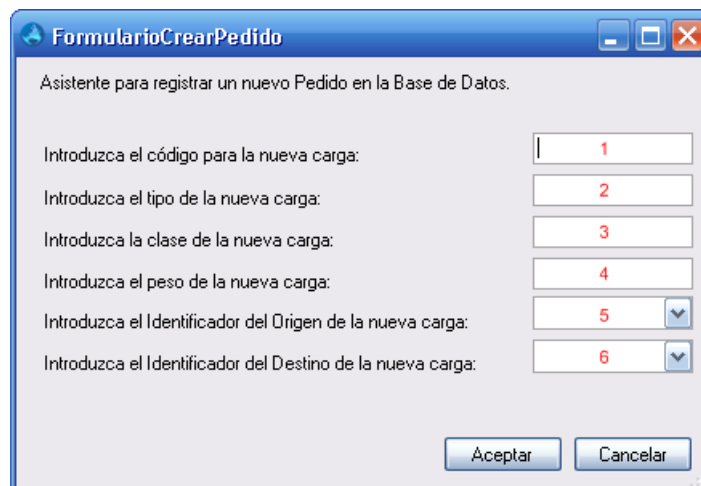


Ilustración 45. Captura del formulario para crear un pedido.

- 7.1.1. Cadena con el código del pedido. No permite valor nulo.
- 7.1.2. Cadena con el tipo del pedido. Sí permite valor nulo.
- 7.1.3. Cadena con la clase del pedido. Sí permite valor nulo.
- 7.1.4. Peso del pedido. Sí permite valor nulo.

- 7.1.5. Identificador del origen del pedido. Sí permite valor nulo.
 - 7.1.6. Identificador del destino del pedido. Sí permite valor nulo.
- 7.2. Formulario para eliminar un registro: Permite eliminar los datos de un pedido registrado en la base de datos. Seleccionando la opción deseada, se podrá eliminar el pedido por su identificador o por su código.

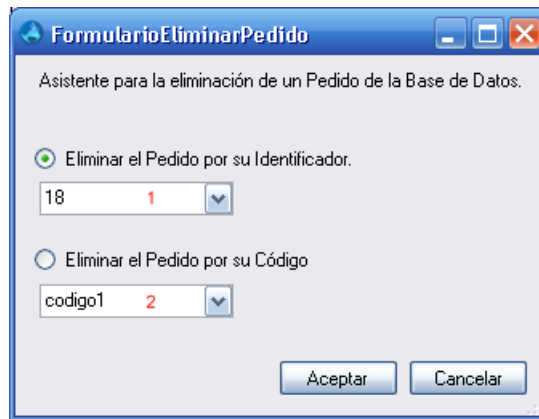


Ilustración 46. Captura del formulario para eliminar un pedido.

- 7.2.1. Identificador del pedido a eliminar.
 - 7.2.2. Cadena con el código del pedido a eliminar.
- 7.3. Formulario para modificar un registro: Permite modificar los datos de un pedido registrado en la base de datos.

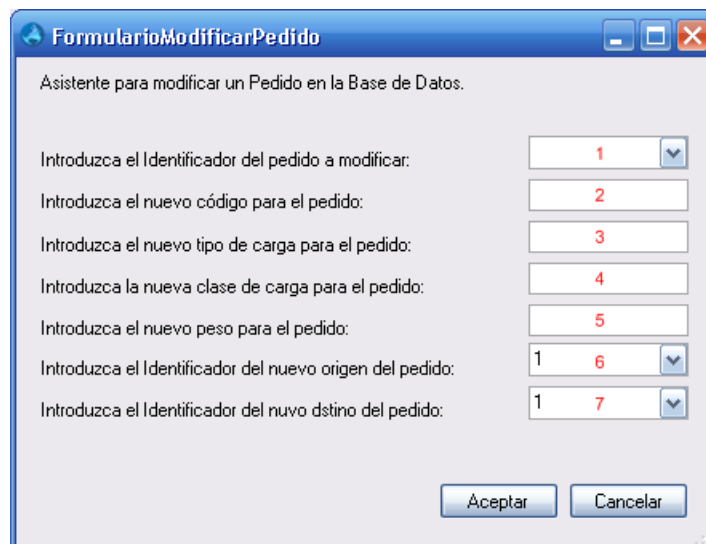


Ilustración 47. Captura del formulario para modificar un pedido.

- 7.3.1. Identificador del pedido a modificar. No permite valor nulo.
- 7.3.2. Cadena con el código del pedido. No permite valor nulo.
- 7.3.3. Cadena con el tipo del pedido. Sí permite valor nulo.
- 7.3.4. Cadena con la clase del pedido. Sí permite valor nulo.

- 7.3.5. Peso del pedido. Sí permite valor nulo.
- 7.3.6. Identificador del origen del pedido. Sí permite valor nulo.
- 7.3.7. Identificador del destino del pedido. Sí permite valor nulo.

7.4. Formulario para filtrar datos: Permite filtrar los datos de la tabla.

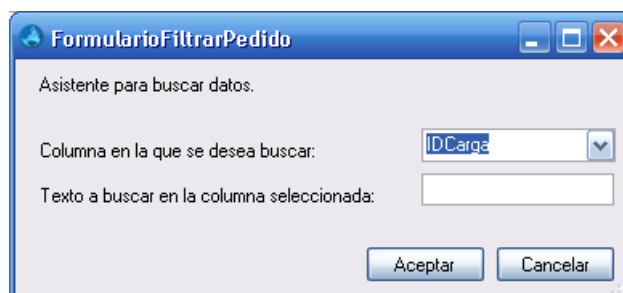


Ilustración 48. Captura del formulario para filtrar pedidos.

- 7.4.1. Nombre de la columna en la que buscar datos.
 - 7.4.2. Cadena a buscar en la columna seleccionada en el anterior campo.
- 7.5. Formulario para configurar un pedido: Permite asociar una carga a un contexto.

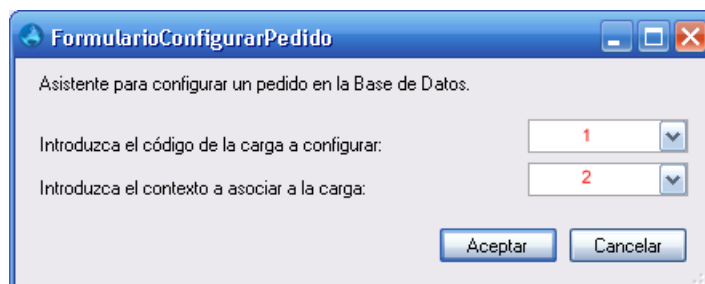


Ilustración 49. Captura del formulario para configurar un pedido.

- 7.5.1. Código de la carga o pedido que se desea configurar.
 - 7.5.2. Identificador del contexto al que asociar una carga o pedido.
8. Formularios disponibles para la pestaña "Propietarios":
- 8.1. Formulario para crear un nuevo registro: Permite registrar un nuevo propietario en la base de datos.

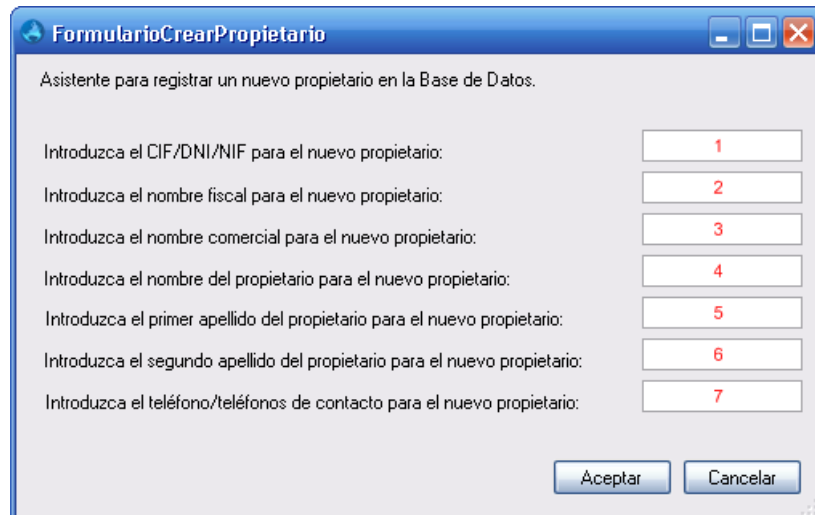


Ilustración 50. Captura del formulario para crear un propietario.

- 8.1.1. Cadena con el CIF/DNI/NIF del propietario. No permite valor nulo.
 - 8.1.2. Cadena con el nombre fiscal del propietario. Sí permite valor nulo.
 - 8.1.3. Cadena con el nombre comercial del propietario. Sí permite valor nulo.
 - 8.1.4. Cadena con el nombre del propietario. Sí permite valor nulo.
 - 8.1.5. Cadena con el primer apellido del propietario. Sí permite valor nulo.
 - 8.1.6. Cadena con el segundo apellido del propietario. Sí permite valor nulo.
 - 8.1.7. Cadena con el/los teléfono/s del propietario. Sí permite valor nulo.
- 8.2. Formulario para eliminar un registro: Permite eliminar los datos de un propietario registrado en la base de datos. Seleccionando la opción deseada, se podrá eliminar el propietario por su identificador o por su CIF/DNI/NIF.

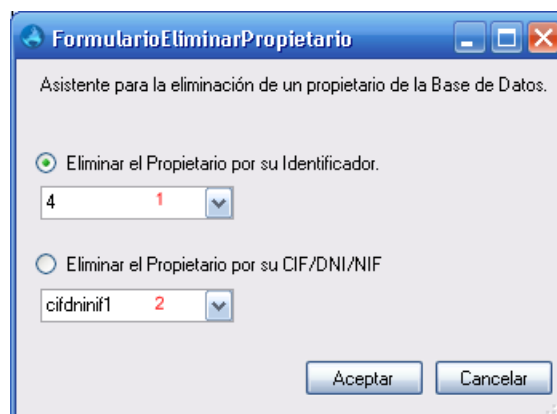


Ilustración 51. Captura del formulario para eliminar un propietario.

- 8.2.1. Identificador del propietario a eliminar.
 - 8.2.2. Cadena con el CIF/DNI/NIF del propietario a eliminar.
- 8.3. Formulario para modificar un registro: Permite modificar los datos de un propietario registrado en la base de datos.

FormularioModificarPropietario

Asistente para modificar los datos de un propietario en la Base de Datos.

Introduzca el identificador del propietario a modificar: 1

Introduzca el nuevo CIF/DNI/NIF para el propietario: 2

Introduzca el nuevo nombre fiscal para el propietario: 3

Introduzca el nuevo nombre comercial para el propietario: 4

Introduzca el nuevo nombre del propietario para el propietario: 5

Introduzca el nuevo primer apellido para el propietario: 6

Introduzca el nuevo segundo apellido para el propietario: 7

Introduzca el/los nuevo/s teléfono/s de contacto para el propietario: 8

Aceptar Cancelar

Ilustración 52. Captura del formulario para modificar un propietario.

- 8.3.1. Identificador del propietario a modificar. No permite valor nulo.
- 8.3.2. Cadena con el CIF/DNI/NIF del propietario. No permite valor nulo.
- 8.3.3. Cadena con el nombre fiscal del propietario. Sí permite valor nulo.
- 8.3.4. Cadena con el nombre comercial del propietario. Sí permite valor nulo.
- 8.3.5. Cadena con el nombre del propietario. Sí permite valor nulo.
- 8.3.6. Cadena con el primer apellido del propietario. Sí permite valor nulo.
- 8.3.7. Cadena con el segundo apellido del propietario. Sí permite valor nulo.
- 8.3.8. Cadena con el/los teléfono/s del propietario. Sí permite valor nulo.

8.4. Formulario para filtrar datos: Permite filtrar los datos de la tabla.

FormularioFiltrarPropietario

Asistente para buscar datos.

Columna en la que se desea buscar: IDPropietario

Texto a buscar en la columna seleccionada:

Aceptar Cancelar

Ilustración 53. Captura del formulario para filtrar propietarios.

- 8.4.1. Nombre de la columna en la que buscar datos.
- 8.4.2. Cadena a buscar en la columna seleccionada en el anterior campo.

9. Formularios disponibles para la pestaña “Remolques”:

- 9.1. Formulario para crear un nuevo registro: Permite registrar un nuevo remolque en la base de datos.

FormularioCrearRemolque

Asistente para registrar un nuevo remolque en la Base de Datos.

Introduzca el número de bastidor dor para el nuevo remolque: 1

Introduzca la matrícula para el nuevo remolque: 2

Introduzca el fabricante para el nuevo remolque: 3

Introduzca el modelo para el nuevo remolque: 4

Introduzca el tipo de remolque: 5

Aceptar Cancelar

Ilustración 54. Captura del formulario para crear un remolque.

- 9.1.1. Cadena con el número de bastidor del remolque. No permite valor nulo.
 - 9.1.2. Cadena con la matrícula del remolque. No permite valor nulo.
 - 9.1.3. Cadena con el fabricante del remolque. No permite valor nulo.
 - 9.1.4. Cadena con el modelo del remolque. No permite valor nulo.
 - 9.1.5. Cadena con el tipo de remolque. Sí permite valor nulo.
- 9.2. Formulario para eliminar un registro: Permite eliminar los datos de un remolque registrado en la base de datos. Seleccionando la opción deseada, se podrá eliminar el remolque por su identificador o por su número de bastidor.

FormularioEliminarRemolque

Asistente para la eliminación de un remolque de la Base de Datos.

Eliminar el Remolque por su Identificador.

5 1

Eliminar el Remolque por su Número de Bastidor

bastidor1 2

Aceptar Cancelar

Ilustración 55. Captura del formulario para eliminar un remolque.

- 9.2.1. Identificador del remolque a eliminar.
 - 9.2.2. Cadena con el número de bastidor del remolque a eliminar.
- 9.3. Formulario para modificar un registro: Permite modificar los datos de un remolque registrado en la base de datos.

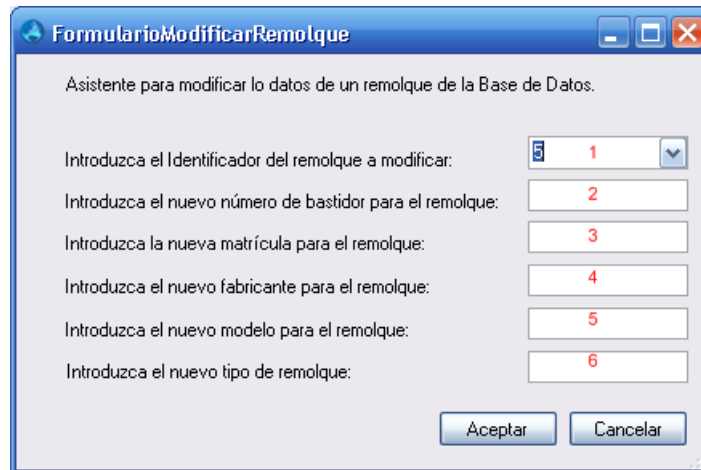


Ilustración 56. Captura del formulario para modificar un remolque.

- 9.3.1. Identificador del remolque a eliminar. No permite valor nulo.
- 9.3.2. Cadena con el número de bastidor del remolque. No permite valor nulo.
- 9.3.3. Cadena con la matrícula del remolque. No permite valor nulo.
- 9.3.4. Cadena con el fabricante del remolque. No permite valor nulo.
- 9.3.5. Cadena con el modelo del remolque. No permite valor nulo.
- 9.3.6. Cadena con el tipo de remolque. Sí permite valor nulo.

9.4. Formulario para filtrar datos: Permite filtrar los datos de la tabla.

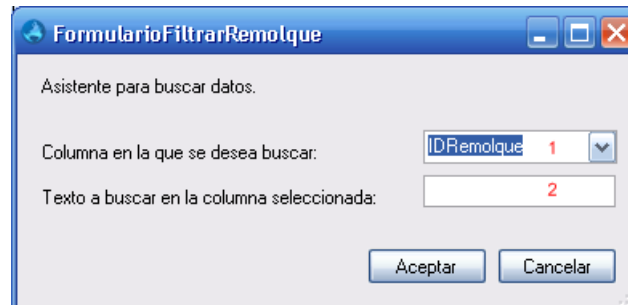


Ilustración 57. Captura del formulario para filtrar remolques.

- 9.4.1. Nombre de la columna en la que buscar datos.
- 9.4.2. Cadena a buscar en la columna seleccionada en el anterior campo.

3.9.6 Descripción de las herramientas para mostrar puntos y rutas en Google Earth.

Una de las funcionalidades implementadas en la aplicación *Cliente* es la capacidad de poder mostrar puntos y rutas, de los diferentes ítems, en el Google Earth. Para poder acceder a estas herramientas se podrá hacer desde el menú "Herramientas" o desde los botones ubicados en la barra de botones.

La herramienta para mostrar la posición de un ítem visualiza en Google Earth la última posición conocida de éste. Al acceder a la herramienta se muestra un cuadro de diálogo en el que se elige el ítem que se desea localizar y, tras pulsar en “Aceptar”, se abre automáticamente el Google Earth mostrando su última posición conocida.

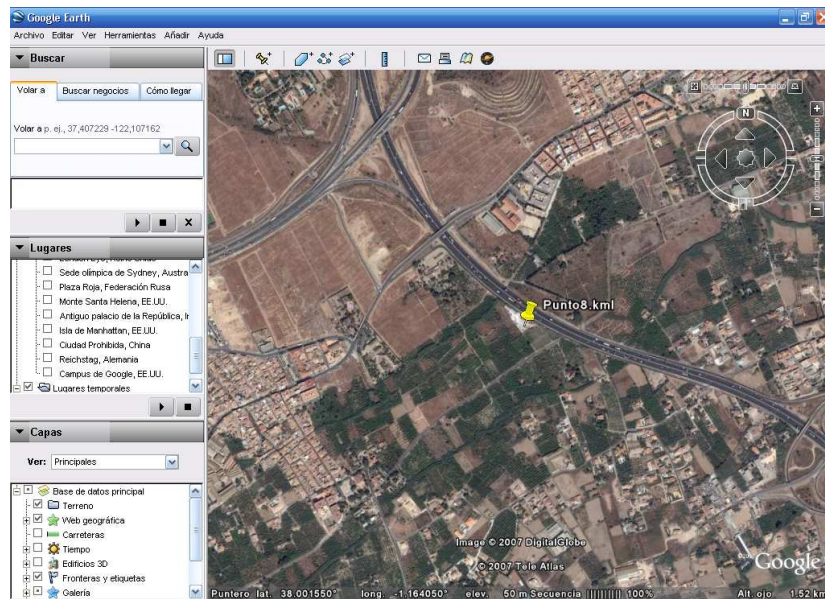


Ilustración 58. Captura de la visualización en Google Earth de un punto.

La herramienta para mostrar la ruta seguida por un ítem es muy similar a la existente para mostrar la posición. La principal diferencia radica en que al mostrar una ruta se examina cuál es el contexto que está cursando actualmente el ítem y, cogiendo las coordenadas registradas con este contexto, se construye la ruta siguiendo los puntos ordenados por fecha.

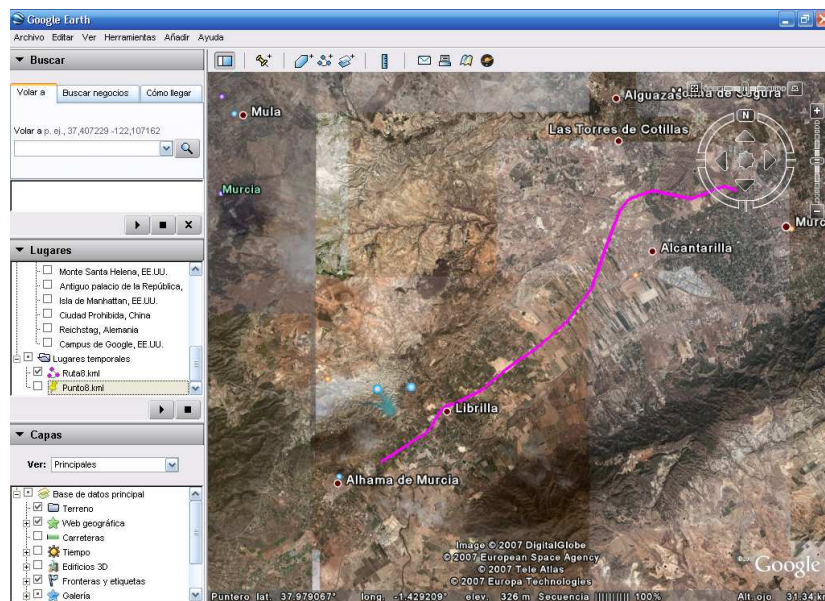


Ilustración 59. Captura de la visualización en Google Earth de una ruta.

Capítulo 4

Conclusiones y líneas futuras

4.1 Conclusiones.

4.1.1 Conclusiones de Microsoft SQL Server.

Las experiencias con Microsoft SQL Server han sido realmente positivas, sobre todo gracias a la herramienta de configuración, SQL Server Management Studio, que proporciona Microsoft.

Durante el desarrollo, SQL Server ha demostrado ser una de las bases de datos más potentes que existe en la actualidad en el mercado. Realmente sería difícil decidir si es mejor o peor que otras bases de datos como MySQL o PostgreSQL, pero sí está claro que si se va a desarrollar una aplicación para Windows y se va a usar Visual Studio, es mejor usar SQL Server por la integración con éste.

El diseño de la base de datos ha sido tan sencillo como cuando se crean tablas en Microsoft Word. Tan sólo es necesario crear la tabla, indicar los nombres de los campos, el tipo de datos de cada campo y configurar las propiedades deseadas en el panel de configuración.

Los principales puntos negativos que se pueden encontrar con esta base de datos son la mala documentación proporcionada por MSDN y el elevado coste por MB en la versión completa. Existiendo otras alternativas gratuitas y más compatibles con otros entornos de desarrollo, ya que no siempre saldrá rentable optar por esta base de datos cuya versión gratuita contiene ciertas limitaciones importantes.

4.1.2 Conclusiones de Microsoft Visual Studio C#.

La implementación en C# de una aplicación tan compleja como la desarrollada en este Proyecto Fin de carrera, ha permitido aprender un lenguaje de programación relativamente nuevo y desconocido para muchos.

Durante el desarrollo, C# ha demostrado ser un lenguaje de programación sencillo y potente. Gracias al compilador y al entorno de desarrollo Visual Studio, la programación ha sido bastante sencilla y positiva si se compara con las experiencias con otros lenguajes en otros entornos de desarrollo.

Como alumno, han sido varios los lenguajes de programación utilizados a lo largo de la carrera y no es necesario mencionar los problemas que pueden dar otros lenguajes como C o C++. En este sentido, la aparición de lenguajes como C# o Java aumentan la productividad en el desarrollo de aplicaciones, aunque este aumento de productividad conlleva un sacrificio en el consumo de recursos hardware.

Por otro lado, Visual Studio y .NET Framework proporcionan muchas facilidades para la integración con otros productos de Microsoft. Así, cuando se utilizan las tecnologías de Microsoft, el desarrollo no presenta casi dificultades. Esto no ocurre así si no todos los productos pertenecen a Microsoft, en cuyo caso se presentaría algunas dificultades en el desarrollo.

La parte más negativa en el desarrollo con C# ha sido la documentación escasa y complicada que proporciona MSDN. En muchas ocasiones, frente a dudas surgidas durante el desarrollo, la documentación proporcionada por los APIs de Microsoft no era suficiente. En este aspecto, Microsoft deberá seguir trabajando para conseguir APIs similares a los proporcionados por Java, ya que por ahora no están a la altura.

4.1.3 Conclusiones del Sistema de Localización.

Se ha conseguido desarrollar una aplicación que cumple con los objetivos fijados en el inicio de este Proyecto Fin de Carrera. Esta primera versión no está libre de errores y ni exenta de futuras modificaciones para su futura mejora. Esta aplicación no podrá utilizarse todavía en empresas, ya que antes de eso tendrá que ser probada el tiempo suficiente y corregir todos los errores que existan.

A pesar de ser una primera versión de pruebas, el resultado obtenido ha sido muy positivo, y puede incentivar la inversión en futuros desarrollos relacionados con este Proyecto Fin de Carrera.

Ha sido muy positivo todo lo aprendido durante el desarrollo de este Proyecto Fin de Carrera. Los conocimientos obtenidos tanto de la estructura a seguir para este tipo de sistemas, como los conocimientos obtenidos con las herramientas utilizadas, son muy valiosos.

Tras la experiencia tenida y observando detenidamente los diferentes sistemas de localización existentes, se puede valorar los grandes avances que se han hecho en este campo. Estos avances estaban accesibles para muy pocas personas hace unos años, pero ahora casi cualquier persona puede realizar una sencilla aplicación de localización utilizando las numerosas herramientas gratuitas disponibles en Internet.

4.2 Líneas futuras.

Los resultados obtenidos con esta primera versión de pruebas han sido bastante satisfactorios. A pesar de ello, para poder competir en el mercado será necesario que se implementen nuevas funcionalidades y se mejoren las ya implementadas.

Durante el desarrollo de la aplicación se han ido descubriendo fallos u otras formas mejores de implementar partes del código. Por este motivo, si en un futuro se desea que esta aplicación pueda competir en un mercado bastante reñido, será necesario mejorar la implementación y la funcionalidad ofrecida.

Como posibles futuras implementaciones, se puede sugerir añadir las siguientes funcionalidades al sistema:

- Representación simultánea de todas las rutas y puntos en Google Earth.
- Distinción entre diferentes tipos de usuarios. Dependiendo del tipo de usuario que se sea, se podrá acceder a unas funcionalidades u otras.
- Interfaces para poder usar otras bases de datos.
- Opciones de copia de seguridad.
- Mejor manejo de los errores ocurridos.
- Crear instalador para una rápida instalación y configuración del sistema.
- Permitir mandar a los vehículos la ruta que se desea que sigan.

Capítulo 5

Bibliografía

BIBLIOGRAFÍA DE MICROSOFT SQL SERVER EXPRESS

- [1] Documentación MSDN de Microsoft.
<http://msdn2.microsoft.com/es-es/library/ms203721.aspx>
- [2] Microsoft SQL Server 2005 Express Edition for Dummies.
Robert Schneider; Wiley Publishing, Inc.

BIBLIOGRAFÍA DE VISUAL STUDIO C# 2005 EXPRESS EDITION

- [4] Documentación MSDN de Microsoft.
[http://msdn2.microsoft.com/es-es/library/kx37x362\(VS.80\).aspx](http://msdn2.microsoft.com/es-es/library/kx37x362(VS.80).aspx)
- [5] Pro C# 2005 and the .NET Platform, Third Edition.
Andrew Troelsen; Apress.
- [6] Expert C# 2005 Business Objects, Second Edition.
Rockford Lhotka; Apress.

BIBLIOGRAFÍA DE GOOGLE EARTH

- [7] Documentación KML de Google.
<http://code.google.com/apis/kml/documentation/index.html>
- [8] Hacking Google Maps and Google Earth.
Martin C. Brown; Wiley Publishing, Inc.