



UNIVERSIDAD
POLITECNICA
DE VALENCIA

UNIVERSIDAD POLITÉCNICA DE VALENCIA

**Escuela Técnica Superior de Ingeniería
Informática**

Ingeniería Técnica en Informática de Gestión

PROYECTO FIN DE CARRERA

**Aplicación web desarrollada con Google Web
Toolkit, Google Maps, Android, Spring e
Hibernate para la localización GPS.**

Autor: Enrique Albert Caballero.

Tutor: Dr. Joan Josep Fons i Cors.

Empresa: Proyectos Joboscan.

Agradecimientos

Me gustaría dar las gracias a las siguientes personas que me han ayudado en el desarrollo del proyecto fin de carrera (PFC):

A mi codirector en la empresa, José Daniel Boscá Candel, por cada uno de los consejos que me ha dado, porque siempre ha tenido una respuesta para las muchas preguntas que le he hecho, y sobretodo porque siempre confió en mí.

También me gustaría dar las gracias a el tutor del proyecto el Dr. Joan Josep Fons i Cors por acceder a dirigirme el proyecto pese a la cantidad de proyectos que está dirigiendo.

Y por último, quiero dedicarle este trabajo y el título de Ingeniero Técnico en Informática a mi familia y a mi novia, por todos los ánimos que me dieron.

Índice de contenido

0. Resumen	7
1. Introducción	8
1.1 Motivación.....	8
1.2 Soluciones existentes.....	8
1.3 Solución propuesta.....	10
1.3.1 Aplicación Android.....	11
1.3.2 Aplicación Web.....	11
1.3.3 Aplicación servidor.....	11
1.4 Resumen de contenidos del documento.....	11
1.5 Esquema general del sistema.....	12
2. Tecnologías implicadas	14
2.1 Sistema GPS.....	14
2.1.1 Introducción.....	14
2.1.2 Cálculo de la posición.....	15
2.1.3 Precisión y disponibilidad del sistema GPS.....	15
2.2 MySQL.....	16
2.3 JAVA.....	17
2.4 Android.....	17
2.4.1 Introducción.....	17
2.4.2 Arquitectura.....	18
2.4.3 Diseño y desarrollo.....	19
2.4.4 Aplicaciones.....	20
2.5 Google web toolkit.....	20
2.5.1 Introducción.....	20
2.5.2 Desarrollo.....	20
2.5.3 Arquitectura GWT.....	21
2.5.4 Características.....	21
3. Especificación de requisitos	23
3.1 Introducción.....	23
3.1.1 Propósito.....	23
3.1.2 Ámbito.....	23
3.1.3 Definiciones, acrónimos y abreviaturas	23
3.1.4 Referencia.....	23
3.1.5 Visión global.....	23
3.2 Descripción general.....	24
3.2.1 Perspectiva del producto.....	24
3.2.2 Funciones del producto.....	24
3.2.3 Características del usuario.....	24
3.2.4 Restricciones generales.....	24
3.2.5 Supuestos y dependencias.....	25
3.3 Requisitos específicos.....	25
3.3.1 Interfaces externas.....	25
3.3.2 Funciones.....	25
3.3.3 Requisitos de rendimiento.....	27
3.3.4 Restricciones de diseño.....	27
3.3.5 Atributos del sistema.....	27

3.3.6 Otros requisitos.....	27
4. Análisis.....	28
4.1 Casos de uso.....	28
4.2 Plantillas casos de uso.....	30
4.3 Diagrama de clases.....	31
4.4 Diagramas de secuencia	32
5. Diseño.....	33
5.1 Bocetos.....	33
5.1.1 Aplicación Android.....	33
5.1.2 Aplicación Web.....	34
6. Implementación.....	37
6.1 Bases de datos.....	37
6.2 Arquitectura distribuida.....	37
6.3 Código.....	38
6.4 Interfaz gráfica.....	39
6.4.1 Dispositivo móvil.....	40
6.4.2 Navegador web.....	43
7. Etapas de la aplicación.....	59
7.1 Fase previa.....	59
7.2 Primera fase.....	59
7.3 Segunda fase.....	59
7.4 Tercera fase.....	59
7.5 Cuarta fase.....	59
7.6 Quinta fase.....	60
7.7 Cronología.....	60
8. Conclusiones.....	61
8.1 Conclusiones.....	61
8.2 Futuras ampliaciones.....	61
9. Referencias.....	62
10. Apéndice.....	63
10.1 Plantillas casos de uso.....	63
10.2 Diagramas de secuencia.....	77

Índice de ilustraciones

Ilustración 1. Captura del sistema de localización de Micronav.....	8
Ilustración 2. Captura del sistema de localización de GlobalSat.....	9
Ilustración 3. Captura del sistema de localización de beatslive.....	10
Ilustración 4. Esquema general del sistema.....	13
Ilustración 5. Constelación de satélites GPS.....	14
Ilustración 6. Cálculo posición GPS.....	15
Ilustración 7. Arquitectura de Android.....	19
Ilustración 8. Boceto android pantalla inicial.....	33
Ilustración 9. Boceto android enviar coordenadas.....	33
Ilustración 10. Boceto web login.....	34
Ilustración 11. Boceto web pantalla principal.....	35
Ilustración 12. Boceto web localizar.....	35
Ilustración 13. Boceto web rutas.....	36
Ilustración 14. Esquema bases de datos.....	37
Ilustración 15 Aplicación Android pantalla inicial.....	40
Ilustración 16 Aplicación Android aviso GPS y datos.....	40
Ilustración 17 Aplicación Android buscando señal.....	41
Ilustración 18 Aplicación Android señal encontrada.....	41
Ilustración 19 Aplicación Android envío de coordenadas.....	42
Ilustración 20 Aplicación Android fin envío de coordenadas.....	42
Ilustración 21 Aplicación web página inicial.....	43
Ilustración 22 Aplicación web formulario usuario.....	44
Ilustración 23 Aplicación web aviso envío correo electrónico.....	44
Ilustración 24 Aplicación web formulario login.....	45
Ilustración 25 Aplicación web olvido contraseña.....	45
Ilustración 26 Aplicación web pantalla principal.....	46
Ilustración 27 Aplicación web menú usuario.....	46
Ilustración 28 Aplicación web formulario modificar usuario.....	47
Ilustración 29 Aplicación web lista vehículos.....	47
Ilustración 30 Aplicación web formulario vehículo nuevo.....	48
Ilustración 31 Aplicación web formulario modificar vehículo.....	48
Ilustración 32 Aplicación web confirmar el borrado de un vehículo.....	49
Ilustración 33 Aplicación web localizar un vehículo.....	50
Ilustración 34 Aplicación web localizar varios vehículos.....	51
Ilustración 35 Aplicación web confirmar borrado de ruta.....	51
Ilustración 36 Aplicación web log de la ruta.....	52
Ilustración 37 Aplicación web lista de las rutas.....	52
Ilustración 38 Aplicación web ruta.....	53
Ilustración 39 Aplicación web calle vehículo.....	54
Ilustración 40 Aplicación web calle fin de ruta.....	54
Ilustración 41 Aplicación web menú administrador.....	54
Ilustración 42 Aplicación web lista dispositivos.....	55
Ilustración 43 Aplicación web formulario dispositivo.....	55
Ilustración 44 Aplicación web confirmación borrado dispositivo.....	56
Ilustración 45 Aplicación web lista conductores.....	56
Ilustración 46 Aplicación web formulario conductor.....	57
Ilustración 47 Aplicación web confirmación borrado conductor.....	57

0. Resumen

En los últimos años, se han empezado a desarrollar sistemas cada vez más sofisticados para la localización y el seguimiento de GPS de vehículos. Este tipo de aplicaciones resultan de gran utilidad para aquellas empresas que están buscando, la manera de realizar un seguimiento de los vehículos de dicha empresa, la visualización en tiempo real de la ruta que siguen los vehículos, la visualización en un mapa de la ruta que tiene que completar los vehículos y de las estadísticas de cada ruta.

Con la realización de este proyecto de fin de carrera (PFC), buscamos construir una aplicación Web que consiga realizar la localización y seguimiento por GPS de una flota de vehículos con la tecnología de Android. Dado el avance de las nuevas tecnologías y las comunicaciones móviles, las personas tienen el alcance de usar la aplicación Web en cualquier momento desde su navegador Web de cualquier teléfono móvil o PDA con conexión a Internet. Dicha aplicación Web consiste en una aplicación Web basada en el lenguaje de programación JAVA, con la API v2 de Google Maps, que ofrece la posibilidad de realizar el seguimiento y visualización gráficamente (en un mapa de Google Maps) de una flota de vehículos, y la ruta seguida de cada uno de ellos sobre un mapa de Google Maps de manera dinámica y en tiempo real.

La aplicación Web va ligada a una aplicación Android basada en el lenguaje de programación Java, que ofrece el envío de coordenadas a cada cierto tiempo, definido por la velocidad a la que se está circulando, para el seguimiento en tiempo real de la flota de vehículos.

Para la realización de dichas aplicaciones se han utilizado el entorno de programación de Eclipse, con el paquete de Android-SDK-Windows, plugin de GWT, plugin Designer y el plugin de Hibernate.

1. Introducción

1.1 Motivación

Durante la lectura de esta memoria se demostrará una serie de aplicaciones que son accesibles a los usuarios desde cualquier navegador Web.

Dado el avance de las nuevas tecnologías y las comunicaciones móviles, las personas tienen el alcance de usar dichas aplicaciones en cualquier momento. Estamos en una época en la que casi cada persona posee un teléfono móvil o PDA con conexión a Internet.

Así pues dadas estas tecnologías y nuevos usos de comunicación como Internet, en este trabajo de fin de carrera ofrece la posibilidad de realizar el seguimiento y visualización gráficamente de una flota de vehículos, y la ruta seguida por cada uno de ellos sobre un mapa de manera dinámica y en tiempo real. Los usuarios tendrán la posibilidad de visualizar la posición actual del vehículo y las rutas que ha realizado a través de un navegador web.

Hemos elegido estas tecnologías ya que cualquier persona puede tener acceso a Internet, ya sea desde una computadora o desde cualquier teléfono móvil con acceso a Internet, y que nos ofrecen todas las características necesarias para crear este proyecto.

Actualmente hay sistemas parecidos al desarrollado en este proyecto, pero con costes muy superiores al nuestro tanto en inversión inicial como en mantenimiento mensual.

1.2 Soluciones existentes

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

Seguimiento y Localización de Flotas de Micronav.

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 para realizar el seguimiento y control en tiempo real de los vehículos.



Ilustración 1. Captura del sistema de localización de Micronav.

Seguimiento y Localización de Flotas de GlobalSat.

El programa muestra sobre la cartografía las rutas y tiempo de paradas de nuestra flota de vehículos. Pudiendo consultar trayectos ocurridos hasta ese momento, o en cualquier día pasado. También se puede consultar la velocidad de los vehículos en todo momento. Esta información se representa tanto gráficamente, como en forma de listados.

Automáticamente, se transmiten las posiciones de vehículos a su PC de la oficina, obteniendo la información de la localización de las rutas realizadas (en tiempo real, **cada 10 segundos**). Además, podremos diseñar fichas de 'Resultado de la visita' personalizadas que se sincronizan automáticamente con los móviles, y que a medida que se cumplimentan en la ruta, se transmiten a la base de datos de la oficina en tiempo real.

También tenemos la posibilidad de **asignar chinchetas a nuestros clientes**, para el control del cumplimiento de las rutas de la flota de vehículos. La información de los clientes a visitar se transmite automáticamente a los móviles de la flota, pudiendo ser consultada en los vehículos.

Otra funcionalidad para el control de la flota, es la **posibilidad de programar alarmas** que envían avisos automáticos mediante SMS a móviles, o vía correo electrónico. Estas alarmas avisan de si el vehículo a estado o no en cierto sitio, y admiten todo tipo de condiciones.



Ilustración 2. Captura del sistema de localización de GlobalSat.

Seguimiento y Localización de Flotas de beatslive.

Seguimiento en tiempo real de vehículo o flotas sobre mapa sin necesidad de recargar la página web.

Consultar movimientos de un vehículo sobre datos históricos (hasta 6 meses o tiempo superior bajo demanda).

Disponer de un informe detallado de la ruta seguida. Controlar las paradas realizadas y tiempos empleados.



Ilustración 3. Captura del sistema de localización de beatslive.

1.3 Solución propuesta

Se requiere de una serie de aplicaciones que permita el seguimiento y visualización de una flota de vehículos y la ruta seguida cada uno de ellos sobre un mapa con visión de zoom de manera dinámica y en tiempo real. El proyecto consta de una Aplicación Android, una Aplicación Web y una Aplicación de servidor.

La aplicación Web utiliza la Api v2 de Google Maps como base de la aplicación. En esta aplicación se puede acceder de dos formas, como cliente o administrador, con diferentes funciones en cada una que detallamos a continuación.

El usuario a través de la Web, podrá hacer las siguientes funciones:

1. Visualizar una ruta de un vehículo seleccionado.
2. Ver en tiempo real la posición de los vehículos seleccionados.
3. Tener una visión a pie de calle del punto que ha seleccionado.
4. Añadir, editar y eliminar vehículos.
5. Ver el log de la ruta y eliminarla.

El administrador, aparte de realizar las funciones del cliente, podrá desempeñar las siguientes funciones:

1. Administrar la creación, edición y eliminación de conductores.
2. Administrar la creación, edición y eliminación de dispositivos.

La aplicación Android, debe estar alojada en un dispositivo móvil de dicha tecnología. La función de la aplicación Android se basa en que dicho cliente al inicio de la ruta activará la aplicación para el envío de coordenadas, y cuando la ruta se ha finalizado correctamente, el usuario desactivará dicha aplicación para no enviar mas coordenadas.

La aplicación de servidor está escuchando a través de un puerto las coordenadas que envían los dispositivos.

1.3.1 Aplicación Android

Se plantea desarrollar una aplicación Android que permita a los usuarios enviar las coordenadas, cada cierto tiempo según a la velocidad en la que se están desplazando, en tiempo real a una base de datos (MySQL). Cada vehículo tendrá un teléfono móvil con GPS con sistema operativo Android, que al inicio de la ruta el usuario activará dicha aplicación, enviando así las coordenadas del vehículo en todo momento hasta la finalización de dicha ruta con la que el usuario desactivará dicha aplicación dejando de enviar las coordenadas.

1.3.2 Aplicación Web

Se plantea desarrollar una aplicación Web realizada en JAVA, que permita a los usuarios el seguimiento y la visualización de las rutas. Además, debe permitir a un administrador de la aplicación Web añadir, borrar o modificar información registrada en la base de datos (rutas, conductor, vehículo, dispositivo).

El usuario, a través del navegador Web, podrá hacer diferentes tipos de consulta:

- Visualización del mapa con zoom, distintos tipos de mapa (Normal, Híbrido, Satélite).
- Elección de una ruta a visualizar, con distinción de cada ruta con un color diferente.
- Log de la ruta: inicio y fin de la ruta, número de coordenadas recibidas.
- Localizar los vehículos seleccionados, con distinción de cada uno con un color diferente.

1.3.3 Aplicación servidor

En el sistema debe existir una conexión entre los dispositivos y la aplicación web. Para ello se pretende realizar una aplicación que reciba las coordenadas de los dispositivos y las almacenara en la base de datos, diferenciando cada uno de los dispositivos en el caso que hayan varios enviando coordenadas simultáneamente.

1.4 Resumen de contenidos del documento

Este documento está dividido en ocho apartados bien diferenciados. Los apartados son los siguientes:

- **Introducción:** en este apartado queda reflejado la motivación por la que he decidido hacer el proyecto, las soluciones que existen, un resumen del documento y un esquema del proyecto.
- **Tecnologías implicadas:** información sobre las tecnologías que he utilizado para elaborar el proyecto. GPS, MySQL, JAVA, Android, etc...
- **Especificación de requisitos:** describe las necesidades del cliente.
- **Análisis:** contiene diagrama de clases, casos de uso, diagrama de secuencias, etc...
- **Diseño:** abarca los bocetos de la interfaz gráfica de las aplicaciones, tanto web como android.

- **Implementación:** explica que bases de datos se ha decidido implantar, qué tipo de arquitectura se ha seleccionado, código de la aplicación y cómo es la interfaz gráfica.
- **Etapas de la aplicación:** indica las fases por las que ha pasado el proyecto y la cronología.
- **Conclusiones:** muestra a que conclusiones he llegado y que futuras ampliaciones podría sufrir la aplicación.
- **Referencias:** enumeración de cada una de las páginas web que he necesitado visitar para obtener información.
- **Apéndice:** incluye todas las plantillas de casos de uso y todos los diagramas de secuencia.

1.5 Esquema general del sistema

La aplicación debe seguir un esquema donde quede reflejado todo el sistema de comunicaciones entre los elementos que intervienen en la aplicación.

Los elementos que intervienen en la aplicación son:

- **Satélites:** nos permiten obtener las coordenadas de los dispositivos, a partir de tres satélites ya se puede calcular una coordenada con buena precisión.
- **Dispositivo:** es el encargado de recoger, a través del GPS, las coordenadas de los satélites y enviarlas, a través de Internet (WIFI/3G), a un servidor que las almacena en la base de datos.
- **Servidor:** ejecuta una aplicación que esta escuchando por un puerto a los dispositivo que envían las coordenadas. Las coordenadas las almacena en la base de datos con la facilidad que nos proporciona Hibernate, ya que nos evita elaborar sentencias complicadas de SQL. También se ejecuta la aplicación web para que se pueda acceder a ella mediante un navegador web.
- **Bases de datos:** nos permite tener almacenado todos los datos del sistema, coordenadas, rutas, conductores, dispositivos, usuarios y vehículos.
- **Cliente web:** desde cualquier navegador web se puede acceder a la aplicación web, sólo hace falta tener conexión a Internet.

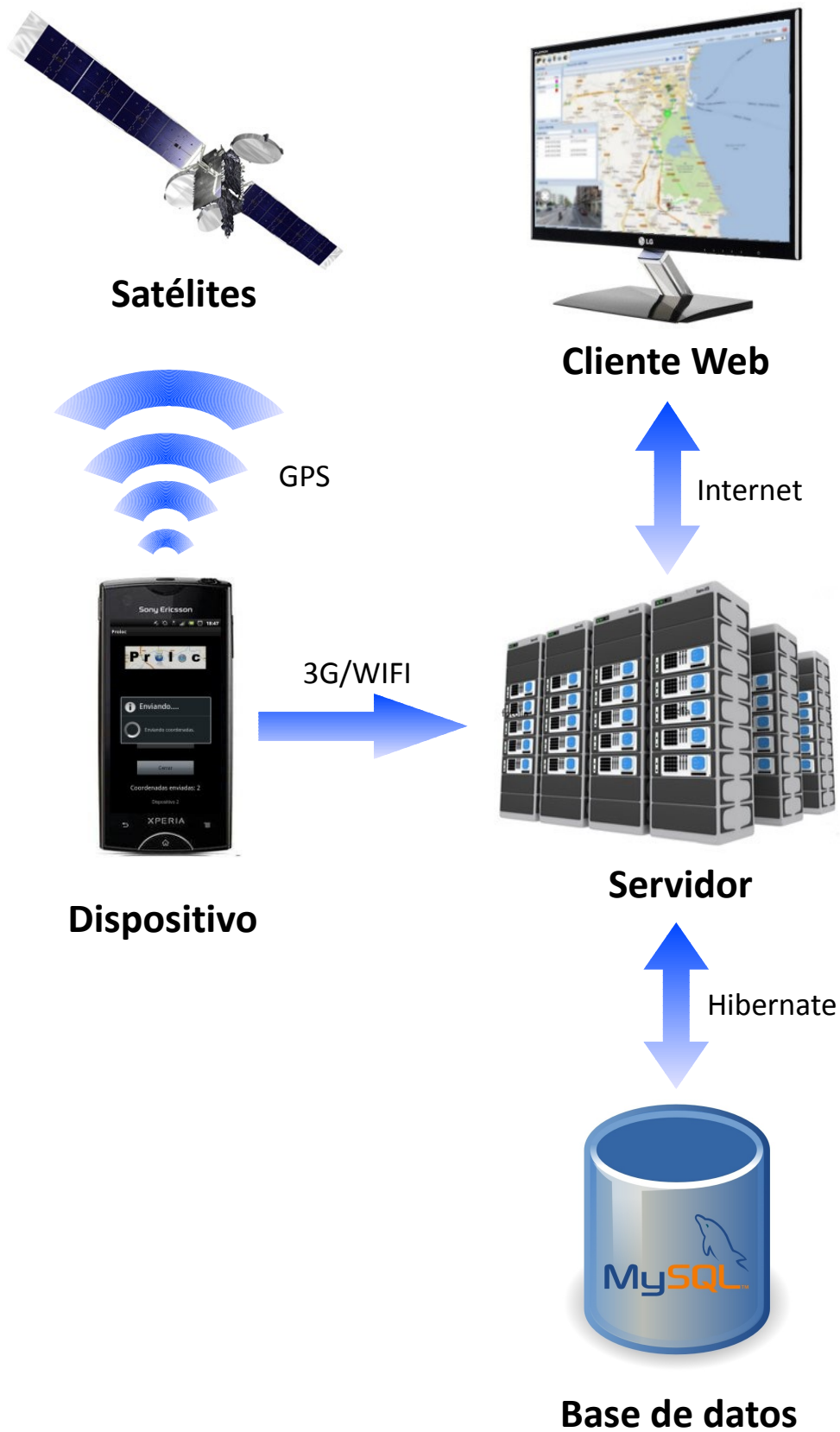


Ilustración 4. Esquema general del sistema.

2. Tecnologías implicadas

2.1 Sistema GPS

2.1.1 Introducción

El GPS (Global Positioning System: sistema de posicionamiento global) o NAVSTAR-GPS es un sistema global de navegación por satélite (GNSS) que permite determinar en todo el mundo la posición de un objeto, una persona o un vehículo con una precisión hasta de centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión. El sistema fue desarrollado, instalado y actualmente operado por el Departamento de Defensa de los Estados Unidos.

El GPS funciona mediante una red de 24 satélites en órbita sobre el globo, a 20.200 kph, con trayectorias sincronizadas para cubrir toda la superficie de la Tierra. Cuando se desea determinar la posición, el receptor que se utiliza para ello localiza automáticamente como mínimo tres satélites de la red, de los que recibe unas señales indicando la identificación y la hora del reloj de cada uno de ellos. Con base en estas señales, el aparato sincroniza el reloj del GPS y calcula el tiempo que tardan en llegar las señales al equipo, y de tal modo mide la distancia al satélite mediante "triangulación" (método de trilateración inversa), la cual se basa en determinar la distancia de cada satélite respecto al punto de medición. Conocidas las distancias, se determina fácilmente la propia posición relativa respecto a los tres satélites. Conociendo además las coordenadas o posición de cada uno de ellos por la señal que emiten, se obtiene la posición absoluta o coordenadas reales del punto de medición. También se consigue una exactitud extrema en el reloj del GPS, similar a la de los relojes atómicos que llevan a bordo cada uno de los satélites.

La antigua Unión Soviética construyó un sistema similar llamado GLONASS, ahora gestionado por la Federación Rusa.

Actualmente la Unión Europea está desarrollando su propio sistema de posicionamiento por satélite, denominado Galileo.

A su vez, la República Popular China está implementando su propio sistema de navegación, el denominado Beidou, que prevén que cuente con entre 12 y 14 satélites entre 2011 y 2015. Para 2020, ya plenamente operativo deberá contar con 30 satélites. De momento (abril 2011), ya tienen 8 en órbita.

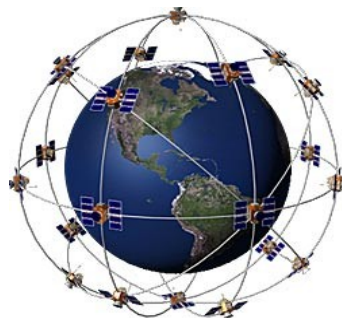


Ilustración 5. Constelación de satélites GPS.

2.1.2 Cálculo de la posición

La posición del receptor se obtiene con la trisección de tres esferas. El centro de cada esfera es la posición del satélite, y el radio es la distancia del receptor al satélite. Por tanto, para poder realizar la trisección, hay que conocer con precisión la posición de cada uno de los satélites que estamos utilizando, y también, conocer la distancia de cada uno de ellos al receptor.

Técnicamente, la trigonometría nos dice que necesitamos 4 satélites para situarnos sin ninguna ambigüedad. A la práctica podemos obtener la posición con solo 3 satélites, si rechazamos las soluciones absurdas.

Los satélites comunican su posición y la de su constelación de forma regular. La posición del satélite proporciona la información de los centros de las esferas.

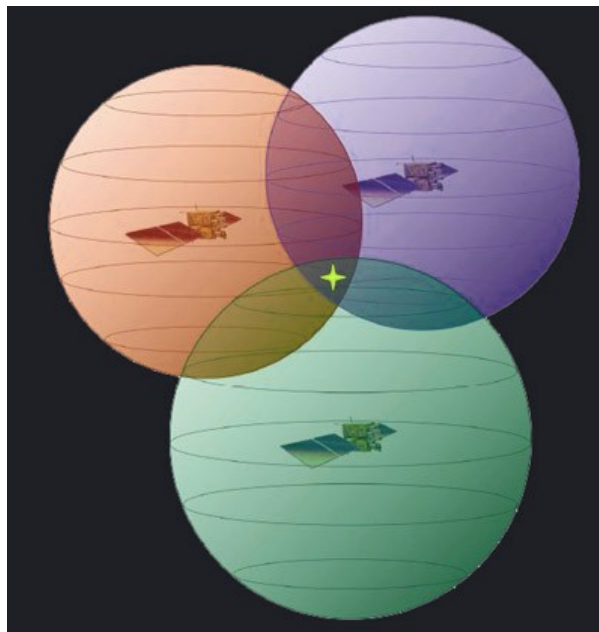


Ilustración 6. Cálculo posición GPS.

El receptor calcula la distancia midiendo el tiempo que tarda en recibir la señal que viene del satélite. Esta señal es una señal binaria pseudoaleatoria conocida por el receptor. El receptor evalúa la correlación de esta secuencia binaria con la secuencia que tiene guardada dentro un registro de desplazamiento. Cuando las dos secuencias coinciden se genera un pulso de amplitud máxima. Si la trama generada por el satélite y la generada por el receptor tienen el mismo instante inicial de tiempo, el intervalo entre la generación y el pulso de amplitud máxima sería igual al tiempo que tarda el señal en llegar al receptor.

2.1.3 Precisión y disponibilidad del sistema GPS

La técnica de procesamiento o localización mediante el sistema GPS puede hacerse con medidas de fase o código. El posicionamiento de vehículos utiliza medidas de código, mientras que las medidas de fase son usadas en aplicaciones geodésicas y topográficas.

Las medidas de código son mucho menos precisas que las de fase. Ofrecen precisiones que oscilan alrededor de un metro en el mejor de los casos, y alrededor de 15 metros en el caso más desfavorable.

El posicionamiento de las medidas para código puede llevarse a cabo de dos maneras:

Modo absoluto: se basa en el cálculo de la posición a partir de un único receptor.

Modo diferencial: se basa en la ayuda de una estación base capaz de generar y suministrar correcciones diferenciales al receptor. Estas correcciones son los factores que debe aplicar el receptor durante el cálculo de la posición para eliminar gran parte de las fuentes de error que afectan a las medidas.

2.2 MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB –desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009- desarrolla MySQL como software libre de un esquema de licenciamiento dual.

Por otro lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSIC.

MySQL es un gestor de base de datos sencillo de usar e increíblemente rápido. También es uno de los motores de base de datos más usados en Internet, la principal razón de esto es que es gratis para aplicaciones no comerciales.

Las características principales de MySQL son:

- Es un gestor de base de datos. Una base de datos es un conjunto de datos y un gestor de base de datos es una aplicación capaz de manejar este conjunto de datos de manera eficiente y cómoda.
- Es una base de datos relacional. Una base de datos relacional es un conjunto de datos que están almacenados en tablas entre las cuales se establecen unas relaciones para manejar los datos de una forma eficiente y segura. Para usar y gestionar una base de datos relacional se usa el lenguaje estándar de programación SQL.
- Es Open Source. El código fuente de MySQL se puede descargar y está accesible a cualquiera, por otra parte, usa la licencia GPL para aplicaciones no comerciales.
- Es una base de datos muy rápida, segura y fácil de usar. Gracias a la colaboración de muchos usuarios, la base de datos se ha ido mejorando optimizándose en velocidad. Por eso es una de las bases de datos más usadas en Internet.
- Existe una gran cantidad de software que la usa.

3 JAVA

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un bytecode, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el bytecode es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del bytecode por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrolladas por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre diciembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java aún no lo es).

El lenguaje Java se creó con cinco objetivos principales:

1. Debería usar la metodología de la programación orientada a objetos.
2. Debería permitir la ejecución de un mismo programa en múltiples sistemas operativos.
3. Debería incluir por defecto soporte para trabajo en red.
4. Debería diseñarse para ejecutar código en sistemas remotos de forma segura.
5. Debería ser fácil de usar y tomar lo mejor de otros lenguajes orientados a objetos, como C++.

2.4 Android

2.4.1 Introducción

Android es un sistema operativo móvil basado en Linux, que junto con aplicaciones middleware, está enfocado para ser utilizado en dispositivos móviles como teléfonos inteligentes, tabletas, Google TV y otros dispositivos. Es desarrollado por la Open Handset Alliance, la cual es liderada por Google.

Fue desarrollado inicialmente por Android Inc., una firma comprada por Google en 2005. Es el principal producto de la Open Handset Alliance, un conglomerado de fabricantes y desarrolladores de hardware, software y operadores de servicio. Las unidades vendidas de teléfonos inteligentes con Android se ubican en el primer puesto en los Estados Unidos, en el segundo y tercer trimestres de 2010, con una cuota de mercado de 43,6% en el tercer trimestre. A nivel mundial alcanzó una cuota de mercado del 50,9% durante el cuarto trimestre de 2011, más del doble que el segundo sistema operativo (iOS de iPhone) con más cuota.

Tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos. A la fecha, se han sobrepasado las 400.000 aplicaciones (de las cuales, dos tercios son gratuitas) disponibles para la tienda de aplicaciones oficial de Android: Google Play, sin tener en cuenta aplicaciones de otras tiendas no oficiales para Android, como pueden ser la App Store de Amazon o la tienda de aplicaciones Samsung Apps de Samsung. Google Play es la tienda de aplicaciones en línea administrada por Google, aunque existe la posibilidad de obtener software externamente. Los programas están escritos en el lenguaje de programación Java. No obstante, no es un sistema operativo libre de malware, aunque la mayoría de ello es descargado de sitios de terceros.

El anuncio del sistema Android se realizó el 5 de noviembre de 2007 junto con la creación de la Open Handset Alliance, un consorcio de 78 compañías de hardware, software y telecomunicaciones dedicadas al desarrollo de estándares abiertos para dispositivos móviles. Google liberó la mayoría del código de Android bajo la licencia Apache, una licencia libre y de código abierto.

La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (surface manager), un framework OpenCore, una base de datos relacional SQLite, una API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic. El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2,8 millones de líneas de lenguaje C, 2,1 millones de líneas de Java y 1,75 millones de líneas de C++.

2.4.2 Arquitectura

Los componentes principales del sistema operativo de Android (cada sección se describe en detalle):

- **Aplicaciones:** las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.
- **Marco de trabajo de aplicaciones:** los desarrolladores tienen acceso completo a los mismos APIs del framework usados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes; cualquier aplicación puede publicar sus capacidades y cualquier otra aplicación puede luego hacer uso de esas capacidades (sujeto a reglas de seguridad del framework). Este mismo mecanismo permite que los componentes sean reemplazados por el usuario.
- **Bibliotecas:** Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.

- **Runtime de Android:** Android incluye un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Executable (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato.dex por la herramienta incluida "dx".
- **Núcleo Linux:** Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

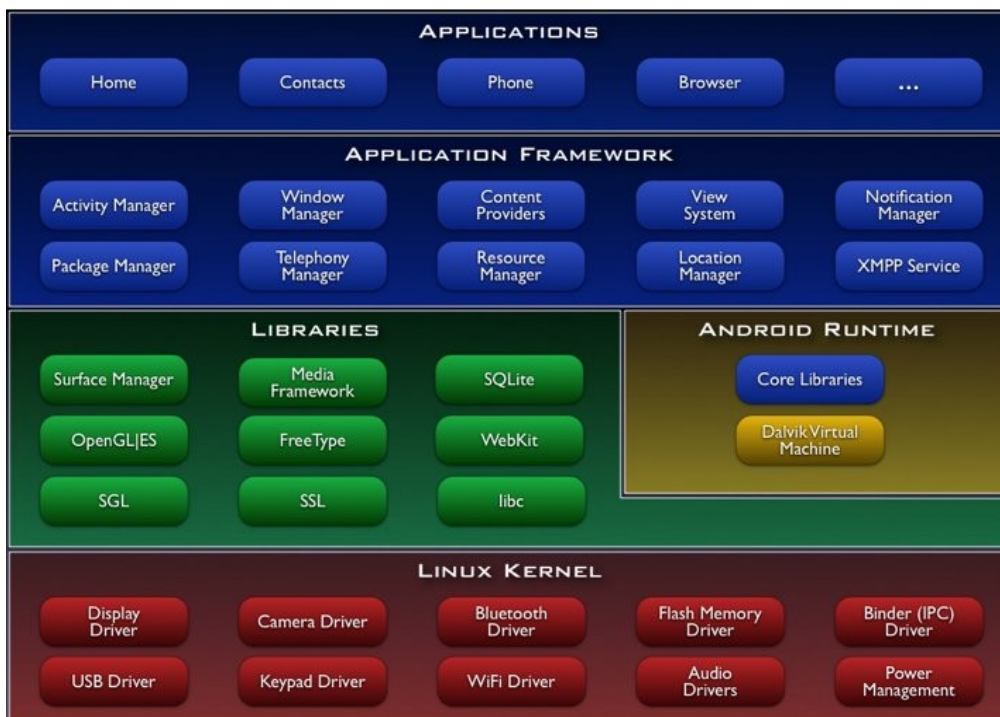


Ilustración 7. Arquitectura de Android.

2.4.3 Diseño y desarrollo

Android, al contrario que otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, se desarrolla de forma abierta y se puede acceder tanto al código fuente como al listado de incidencias donde se pueden ver problemas aún no resueltos y reportar problemas nuevos.

El que se tenga acceso al código fuente no significa que se pueda tener siempre la última versión de Android en un determinado móvil, ya que el código para soportar el hardware (controladores) de cada fabricante normalmente no es público, así que faltaría un trozo básico del firmware para poder hacerlo funcionar en dicho terminal, y porque las nuevas versiones de Android suelen requerir más recursos, por lo que los modelos más antiguos quedan descartados por razones de memoria (RAM), velocidad de procesador, etc.

2.4.4 Aplicaciones

Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android Software Development Kit (Android SDK), pero están disponibles otras herramientas de desarrollo, incluyendo un Kit de Desarrollo Nativo para aplicaciones o extensiones en C o C++, Google App Inventor, un entorno visual para programadores novatos y varios cruz aplicaciones de la plataforma web móvil marcos. y también es posible usar las librerías Qt gracias al proyecto Necessitas SDK.

El desarrollo de aplicaciones para Android no requiere aprender lenguajes complejos de programación. Todo lo que se necesita es un conocimiento aceptable de Java y estar en posesión del kit de desarrollo de software o «SDK» provisto por Google el cual se puede descargar gratuitamente.

2.5 Google web toolkit

2.5.1 Introducción

GWT o Google Web Toolkit es un framework creado por Google que permite ocultar la complejidad de varios aspectos de la tecnología AJAX. Es compatible con varios navegadores, lo cual es notorio ya que cada navegador suele necesitar código específico para lograr un front-end correcto en una aplicación web. El concepto de Google Web Toolkit es bastante sencillo, básicamente lo que se debe hacer es crear el código en Java usando cualquier IDE de Java y el compilador lo traducirá a HTML y JavaScript.

2.5.2 Desarrollo

Con la biblioteca GWT, los desarrolladores pueden crear y depurar aplicaciones AJAX en lenguaje JAVA usando el entorno de desarrollo que prefieran. Cuando una aplicación es desplegada, el compilador GWT traduce la aplicación Java a un archivo Javascript, que puede ser ofuscado para optimizar el rendimiento.

GWT no es sólo una interfaz de programación proporciona un conjunto de herramientas que permiten desarrollar funcionalidades Javascript de alto rendimiento en el navegador del cliente.

Una aplicación GWT puede ser ejecutada en dos modos:

- Modo host (Hosted mode): La aplicación se ejecuta como código bytecode de Java dentro de la Máquina Virtual de Java (JVM). Este modo es el más usado para desarrollo, soportando el cambio de código en caliente y el depurado.
- Modo web (Web mode): La aplicación se ejecuta como código Javascript y HTML puro, compilado a partir del código Java. Este modo se suele usar para el despliegue de la aplicación.

La utilidad de línea de comandos `applicationCreator` genera automáticamente todos los archivos necesarios para iniciar un proyecto GWT, incluso permite crear un proyecto para Eclipse.

Existen varios plugins de código abierto para ayudar a desarrollar en diferentes entornos de desarrollo, como GWT4NB para NetBeans, Cypal Studio for GWT para Eclipse o `gwtDeveloper` para Jdeveloper.

2.5.3 Arquitectura GWT

GWT contiene los siguientes componentes:

- **GWT Java-to-JavaScript Compiler:** la función de este componente es traducir el código desarrollado en Java al lenguaje JavaScript. Lo empleamos cuando usamos al GWT en modo web.
- **Hosted Web Browser:** este componente ejecuta la aplicación Java sin traducirla a JavaScript, en modo host usando la máquina virtual de Java.
- **JRE Emulation Library:** contiene las bibliotecas más importantes de las clases de Java: `java.lang` en donde se encuentran las clases fundamentales para poder programar en Java y un subconjunto de las clases del paquete `java.util`. `Java.lang` incluye, entre otras, la clase `java.lang.object` que es la clase fundamental de la que heredan o extienden todas las clases en Java. El resto de los paquetes no están soportados por GWT.
- **GWT Web UI Class Library:** contiene un conjunto de elementos de interfaz de usuario que permite la creación de objetos tales como textos, cajas de texto, imágenes y botones.

2.5.4 Características

- Componentes gráficos dinámicos y reusables: los programadores pueden usar clases prediseñadas para implementar comportamientos que de otra manera consumirían mucho tiempo, como arrastrar y soltar o menús en árbol.
- Simple mecanismo RPC.
- Gestión del historial del navegador web.
- Soporte para depurado de Java.
- Control de diferentes características del navegador.
- Integración con JUnit.
- Internacionalización.
- Los desarrolladores pueden mezclar código escrito en Javascript dentro del código Java usando la Interfaz Nativa Javascript (JSNI).
- Soporte para la API de Google (inicialmente, soporte para Google Gears).
- Es de código abierto.
- Los desarrolladores pueden diseñar y desarrollar sus aplicaciones orientadas a objetos. Errores comunes en Javascript, como la discrepancia de tipos de datos, son controlados en tiempo de compilación.

- El código Javascript generado puede ser ofuscado para optimizar el rendimiento.
- Existen un numeroso conjunto de bibliotecas desarrolladas por Google y terceros que amplían las funcionalidades de GWT.

3. Especificación de requisitos

3.1 Introducción

3.1.1 Propósito

Este documento ha sido redactado con el fin de marcar las pautas generales y las especificaciones que deberá seguir la aplicación a desarrollar.

En este documento quedan reflejadas las necesidades del cliente para que los desarrolladores encargados de la codificación de la aplicación, en su conjunto, puedan realizar las funciones de desarrollo a partir de las pautas marcadas en él y en los documentos realizados posteriormente.

La especificación de requisitos sirve de canal de comunicación entre las distintas partes implicadas en el desarrollo de la aplicación (Cliente-Desarrollador). La especificación está sujeta a revisiones por las partes implicadas, hasta alcanzar su aprobación. Una vez aprobado servirá de base al equipo de desarrollo para la construcción del nuevo sistema.

3.1.2 Ámbito

El objetivo del producto que vamos a describir es un sistema informático que localiza en tiempo real un dispositivo GPS colocado en un vehículo, mostrando la información recibida en un mapa.

3.1.3 Definiciones, acrónimos y abreviaturas

Definiciones

- **Ruta:** sucesión de puntos ordenados cronológicamente que indica la trazada que ha realizado un vehículo durante un espacio de tiempo.
- **Coordenadas:** es un sistema de referencia que utiliza las dos coordenadas angulares, latitud (Norte y Sur) y longitud (Este y Oeste) y sirve para determinar los ángulos laterales de la superficie terrestre.
- **Log:** fichero donde se almacenan las incidencias.
- **Widgets:** componentes.

Abreviaturas

- **GPS:** Global Positioning System. (Sistema de posicionamiento global)
- **GWT:** Google Web Toolkit.

3.1.4 Referencia

ANSI/IEEE std. 830/1998. Guía del IEEE para la Especificación de Requerimientos Software.

3.1.5 Visión global

Este apartado está estructurado en tres secciones. Estas son las siguientes:

En la primera sección se realiza una introducción al apartado y se proporciona una visión

general de la especificación de recursos del sistema.

En la segunda sección del apartado se realiza una descripción general del sistema, con el fin de conocer las principales funciones que éste debe realizar, sin entrar en excesivos detalles.

Por último, la tercera sección del apartado es aquella en la que se definen detalladamente los requisitos que debe satisfacer el sistema.

3.2 Descripción general

3.2.1 Perspectiva del producto

La aplicación PROLOC debe mostrar la posición y la ruta realizada en un espacio de tiempo determinado de cada uno de los vehículos que componen la flota, para tener una mejor gestión de la flota de vehículos de una empresa. Existen dos productos software relacionados, una aplicación web y una aplicación móvil.

3.2.2 Funciones del producto

Las funciones del producto las podemos organizar en varios bloques para una mejor claridad. Éstas son las siguientes:

Web

La aplicación debe recibir los datos del dispositivo GPS y almacenarlos en la base de datos. Los usuarios deben introducir su nombre de usuario y contraseña para acceder a la aplicación. La aplicación permite ver donde se encuentra el vehículo seleccionado en tiempo real y mostrar la ruta que ha recorrido un vehículo.

Móvil

La aplicación debe transmitir las coordenadas a la aplicación web. Creará un “log” donde quede reflejado el inicio y el fin de la aplicación.

3.2.3 Características del usuario

Este producto está pensado para los empresarios que quisieran tener un control de la flota de vehículos de su empresa para aumentar la eficiencia del personal encargado de los transportes. El usuario no necesita un nivel avanzado de conocimientos informáticos, ya que la interfaz gráfica es intuitiva y agradable para el usuario.

3.2.4 Restricciones generales

El dispositivo GPS tiene que estar en una ubicación donde no haya interferencias en la señal de GPS entre el dispositivo y los satélites. La aplicación móvil esta implementada sobre la plataforma Android.

La aplicación web está diseñada en el lenguaje JAVA. La interfaz gráfica está elaborada con GWT (Google Web Toolkit) con los widgets de GXT (Interfaz de Sencha). La base de datos se estructura en MySQL con Hibernate. Otras tecnologías que intervienen en la aplicación son Spring, Google Maps y Jetty.

3.2.5 Supuestos y dependencias

El sistema correrá sobre cualquier plataforma, ya que se trata de una aplicación web que se ejecuta en un navegador web. Las pruebas serán realizadas sobre la plataforma Windows XP. En cuanto al sistema móvil solo puede ejecutarse sobre la plataforma Android.

3.3 Requisitos específicos

3.3.1 Interfaces externas

El sistema móvil en fases posteriores puede ejecutarse en otras plataformas como puede ser Symbian (Nokia) o iOS (iPhone).

3.3.2 Funciones

Esta subsección está organizada según el tipo de usuario del sistema.

Usuario

Alta usuario: El usuario accederá a la aplicación mediante la URL y aparecerá un formulario donde podrá acceder al sistema si ya está registrado o darse de alta pulsando en el link correspondiente. Si no está registrado, el usuario deberá rellenar un formulario introduciendo sus datos personales: nombre, apellidos, DNI, empresa, correo electrónico y contraseña.

Salir usuario: El usuario en cualquier momento puede salir del sistema y aparece la página de acceso a la aplicación.

Modificar usuario: El usuario puede editar su perfil debido a algún cambio con respecto al inicio del registro.

Añadir vehículo: El usuario puede añadir vehículos que desee localizar. Debe rellenar un formulario con los siguientes campos: matrícula, modelo, tipo, color, empresa y conductor.

Borrar vehículo: El usuario puede eliminar un vehículo si ya no quiere que este sea localizado.

Modificar vehículo: El usuario puede modificar los atributos del vehículo excepto la matrícula.

Localizar: El usuario selecciona un vehículo en la lista de vehículos y pulsando el botón “Localizar”, en el mapa principal nos aparece una marca indicando donde se encuentra el vehículo en este instante.

Ver rutas: El usuario puede saber las rutas que ha realizado el. Tiene que seleccionar el vehículo, pulsar el botón “Ver rutas” y aparece una ventana con una lista de todas las rutas que ha realizado el vehículo. El usuario selecciona la ruta que quiere ver y en el mapa principal aparecerá mediante una línea del color del vehículo el trazado que ha realizado el vehículo en dicha ruta.

Eliminar ruta: El usuario selecciona una ruta de la lista de rutas y pulsando el botón “Eliminar ruta”, la elimina de la base de datos.

Ver log: El usuario puede ver el log de la ruta donde queda reflejado la fecha de inicio, fin y número de coordenadas recibidas.

Conductor

Activar dispositivo: El conductor puede activar el dispositivo mediante la aplicación móvil para que este empiece a emitir datos al servidor.

Desactivar dispositivo: El conductor puede desactivar el dispositivo una vez haya terminado el proceso de localización.

Administrador

El administrador puede hacer todas las funciones de un usuario.

Login administrador: El administrador debe introducir su nombre y contraseña para poder acceder al sistema.

Salir administrador: El administrador en cualquier momento puede salir del sistema y aparece la página de acceso a la aplicación.

Modificar datos: El administrador tiene la posibilidad de modificar sus datos personales que serán nombre, apellidos, DNI, correo electrónico y contraseña.

Añadir dispositivo: El administrador puede añadir un dispositivo mediante un formulario introduciendo un identificador numérico y un nombre. Más tarde este será asociado a un vehículo.

Eliminar dispositivo: El administrador puede eliminar un dispositivo siempre i cuando esté libre.

Modificar dispositivo: El administrador puede modificar los datos de un dispositivo.

Añadir conductor: El administrador puede añadir conductores a la plantilla de la empresa mediante un formulario. Debe introducir el nombre, apellidos, DNI, empresa y teléfono del conductor.

Modificar conductor: El administrador puede modificar los datos de los conductores.

Eliminar conductor: El administrador puede eliminar un conductor de la plantilla de la empresa.

3.3.3 Requisitos de rendimiento

La aplicación soportará la conexión de varios usuarios conectados simultáneamente con sus respectivos dispositivos conectados. El sistema debe responder de la forma más eficiente posible para poder mostrar la localización de los vehículos en tiempo real.

3.3.4 Restricciones de diseño

La interfaz gráfica será intuitiva para facilitar el manejo de la aplicación tanto a los usuarios como a los administradores. En la pantalla principal debe aparecer un mapa grande donde quedará plasmada toda la información que solicite el usuario. También debe aparecer un menú con las acciones que puede realizar el usuario como añadir un vehículo, localizar, mostrar una ruta, etc....

3.3.5 Atributos del sistema

Para acceder al sistema tendremos que introducir en un navegador web la dirección URL de la aplicación alojada en el servidor. El primer paso es acceder mediante el nombre y contraseña del usuario. El administrador es el encargado de la gestión del sistema y el usuario es el que utiliza la aplicación.

3.3.6 Otros requisitos

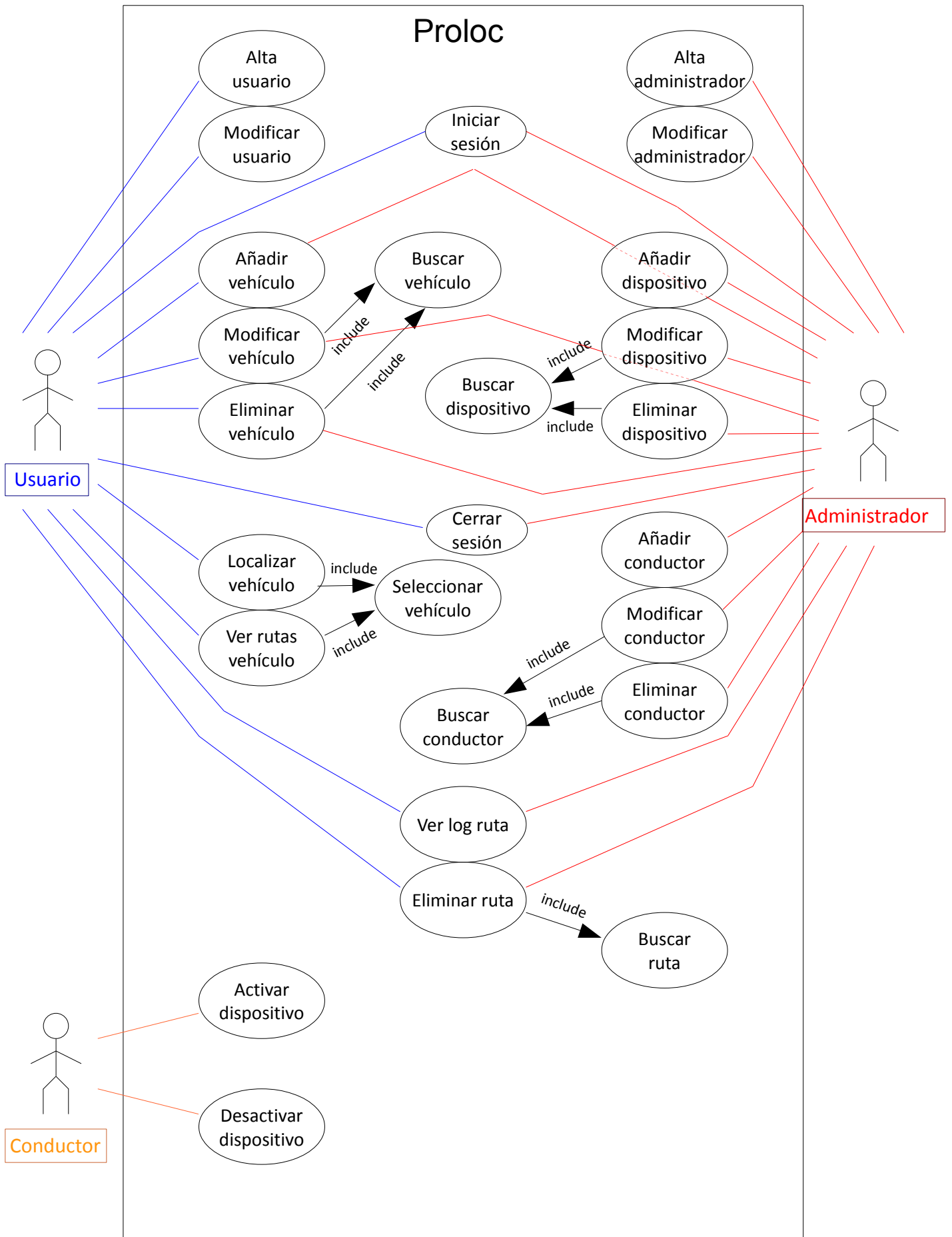
El dispositivo móvil debe disponer de un módulo GPS, conexión a internet y sistema operativo Android.

4. Análisis

4.1 Casos de uso

Un caso de uso es una descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores. En el contexto de ingeniería del software, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo la especialización y la generalización son relaciones. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo.

Los más comunes para la captura de requisitos funcionales, especialmente con el desarrollo del paradigma de la programación orientada a objetos, donde se originaron, si bien puede utilizarse con resultados igualmente satisfactorios con otros paradigmas de programación.



4.2 Plantillas casos de uso

A partir del esquema se realizan las plantillas de cada una de las funciones representadas anteriormente.

Esta es la plantilla que hace referencia a “Añadir vehículo”. Es una acción que puede realizar un usuario y queda reflejada la interacción del usuario con el sistema.

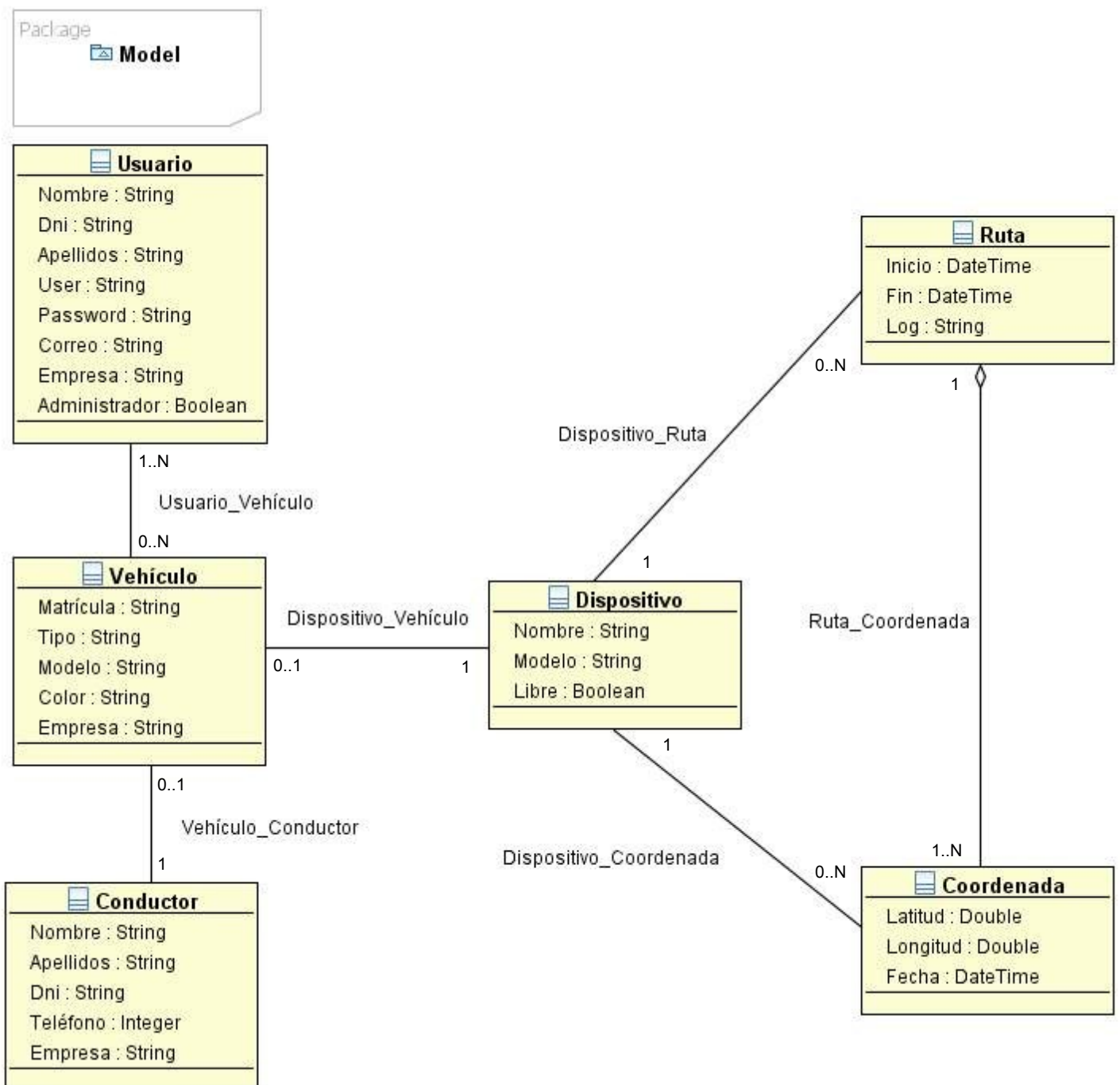
<i>Caso de uso</i>	Añadir vehículo
<i>Actores</i>	Usuario
<i>Propósito</i>	Añadir un vehículo en el sistema para localizarlo.
<i>Resumen</i>	El usuario pulsa el botón añadir vehículo. Aparece un formulario para rellenar los datos del vehículo. Una vez guardado, el nuevo vehículo aparecerá en la tabla de vehículos disponibles del usuario.
<i>Pre-condiciones</i>	El usuario se ha identificado en el sistema.
<i>Post-condiciones</i>	El vehículo se almacena en el sistema.
<i>Usuario</i>	<i>Sistema</i>
1. El usuario pulsa el botón añadir vehículo.	
	2. El sistema muestra el formulario añadir vehículo.
3. El usuario rellena los datos del vehículo.	
	4. El sistema comprueba si existe o hay algún error.
	5. El sistema muestra una ventana de confirmación.
6. El usuario confirma	
	7. El sistema almacena el nuevo vehículo y le asigna un dispositivo libre.

Todas las plantillas se ubican en el apéndice.

4.3 Diagrama de clases

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

Para la elaboración del diagrama de clases he utilizado una herramienta llamada *Moskitt*, que nos permite crear diagramas de una forma sencilla y rápida.

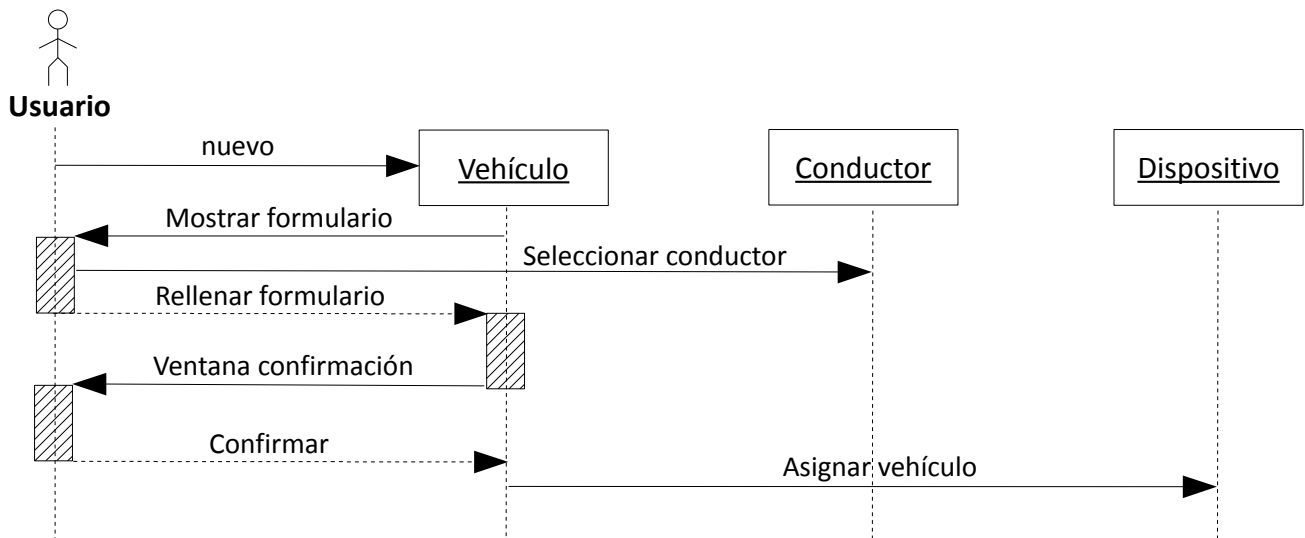


4.4 Diagramas de secuencia

Un diagrama de secuencia es un tipo de diagrama usado para modelar interacción entre objetos en un sistema según UML. Muestra la interacción de un conjunto de objetos en una aplicación a través del tiempo y se modela para cada caso de uso. Mientras que el diagrama de casos de uso permite el modelado de una vista business del escenario, el diagrama de secuencia contiene detalles de implementación del escenario, incluyendo los objetos y clases que se usan para implementar el escenario, y mensajes intercambiados entre los objetos.

Típicamente se examina la descripción de un caso de uso para determinar qué objetos son necesarios para la implementación del escenario. Si se dispone de la descripción de cada caso de uso como una secuencia de varios pasos, entonces se puede "caminar sobre" esos pasos para descubrir qué objetos son necesarios para que se puedan seguir los pasos. Un diagrama de secuencia muestra los objetos que intervienen en el escenario con líneas discontinuas verticales, y los mensajes pasados entre los objetos como flechas horizontales.

Este es el diagrama de secuencia de "Añadir vehículo". Podemos ver los objetos que intervienen, las acciones y el actor.



Todos los diagramas de secuencia se ubican en el apéndice.

5. Diseño

5.1 Bocetos

Antes de programar la interfaz gráfica de las aplicaciones, hay que hacer unos bocetos para tener una idea aproximada de la apariencia que tendrá la aplicación.

5.1.1 Aplicación Android

Los siguientes bocetos muestran como se espera que sea la interfaz de la aplicación android.

La pantalla inicial estará formada por una imagen del logo de la aplicación, un botón para activar el envío de coordenadas y un botón para cerrar la aplicación.

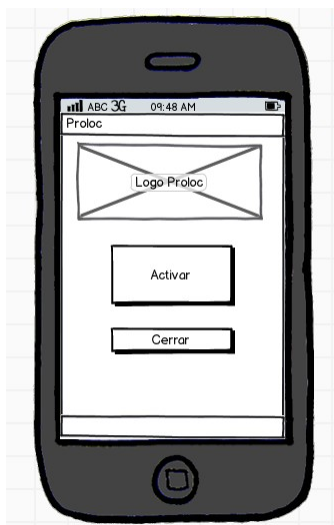


Ilustración 8. Boceto android pantalla inicial.

Cuando se pulsa el botón activar aparecerá una pantalla indicando que se están enviando las coordenadas.



Ilustración 9. Boceto android enviar coordenadas.

5.1.2 Aplicación Web

Los siguientes bocetos nos muestran una aproximación de lo que será la apariencia de la aplicación web.

Esta es la primera pantalla que aparece en la aplicación. Esta compuesta por un mapa en el fondo y una ventana para introducir los datos del usuario.

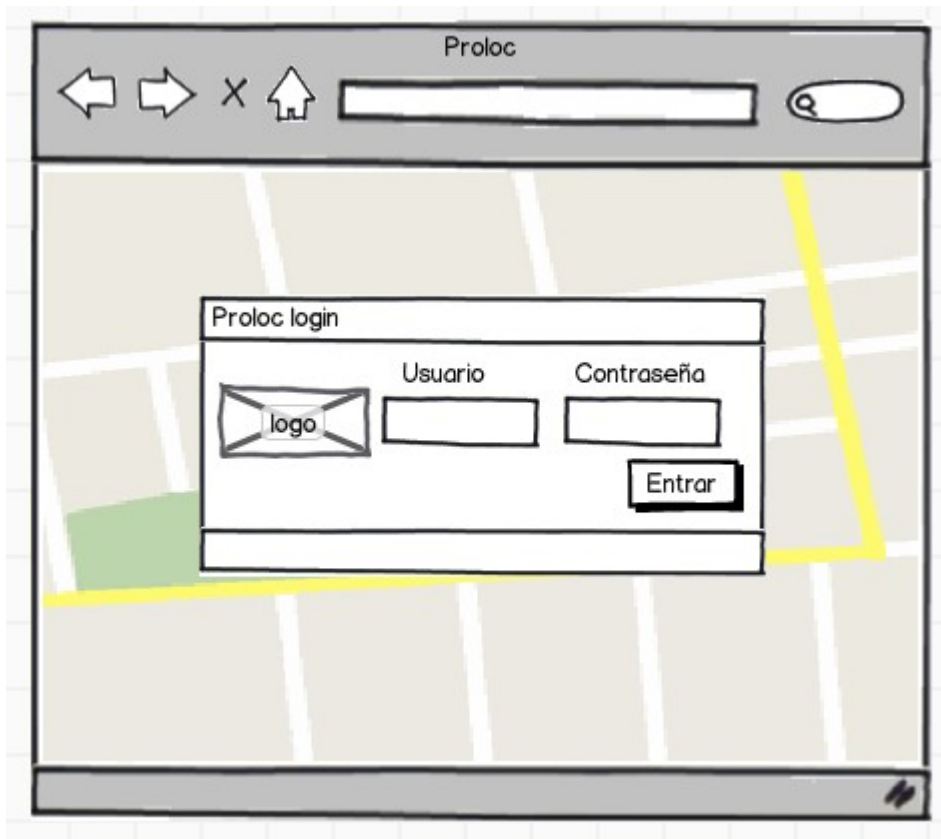


Ilustración 10. Boceto web login.

Al introducir bien los datos del usuario, el sistema permite acceder a la aplicación mostrando el mapa y una tabla con los vehículos que están asociados al usuario. La tabla contiene un botón para localizar al vehículo seleccionado y otro botón para mostrar las rutas que ha realizado el vehículo seleccionado.

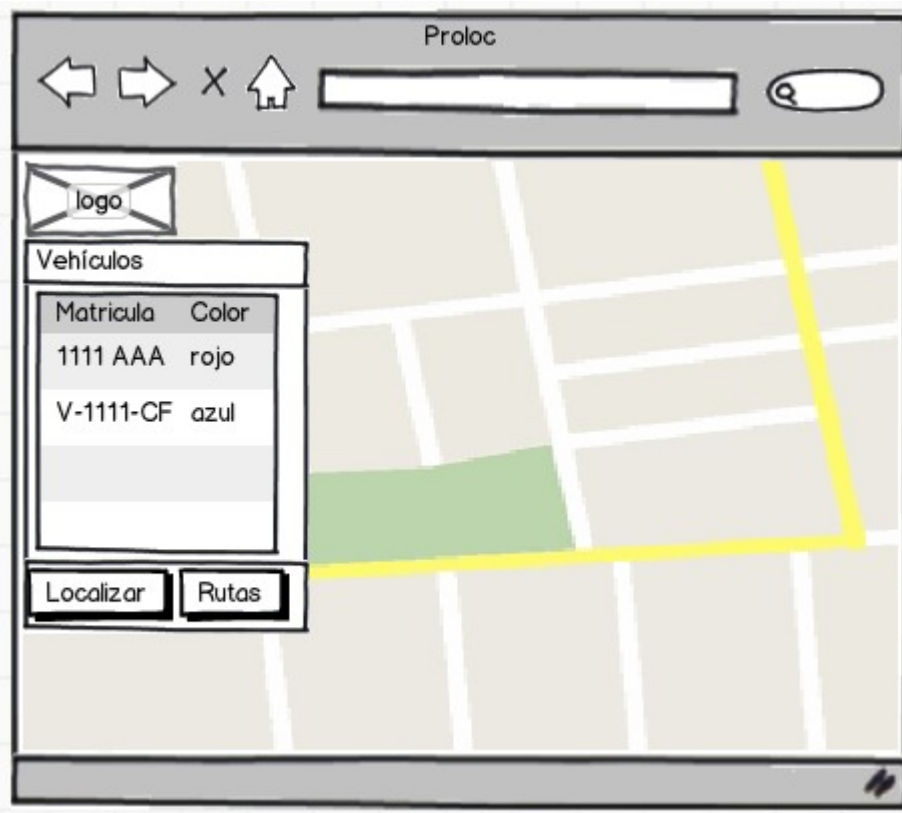


Ilustración 11. Boceto web pantalla principal.

Para localizar a un vehículo hay que pulsar el botón localizar y en el mapa aparece una marca donde indica donde se encuentra el vehículo.

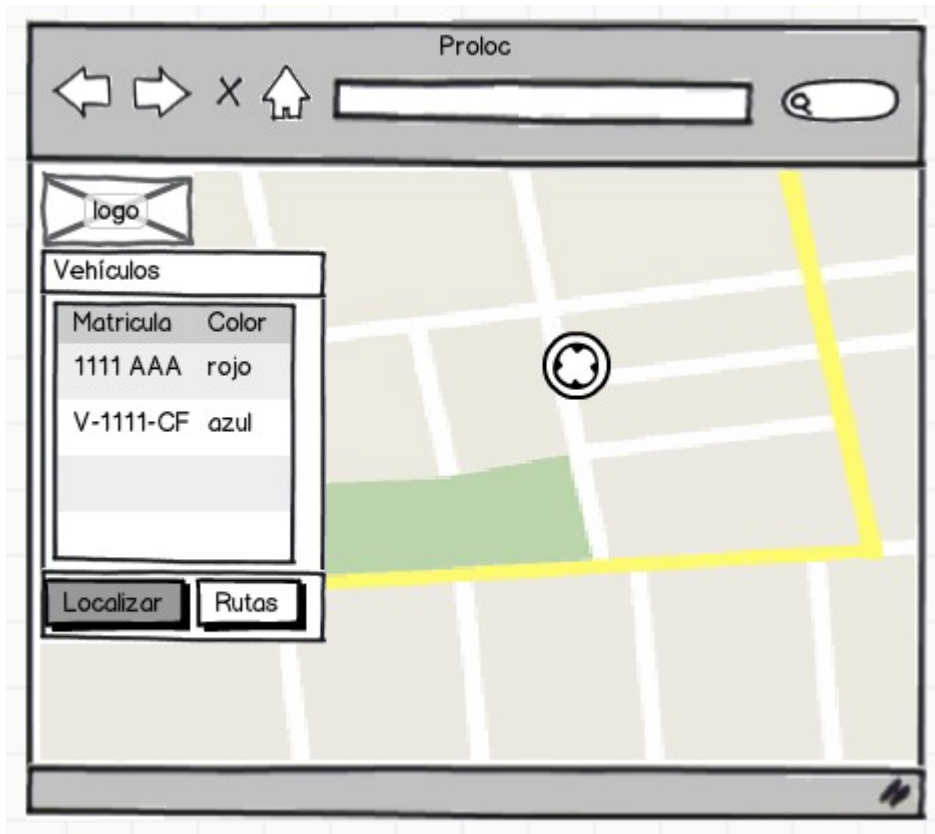


Ilustración 12. Boceto web localizar.

Para mostrar una ruta pulsamos el botón rutas y aparece una lista con las rutas realizadas. Al seleccionar una ruta la aplicación construye la ruta en el mapa.



Ilustración 13. Boceto web rutas.

6. Implementación

6.1 Bases de datos

Para la aplicación he decidido instalar una base de datos MySQL. Nuestra base de datos posee información acerca de las rutas, coordenadas, usuarios, vehículos, conductores y dispositivos. El esquema de la base de datos es la siguiente:

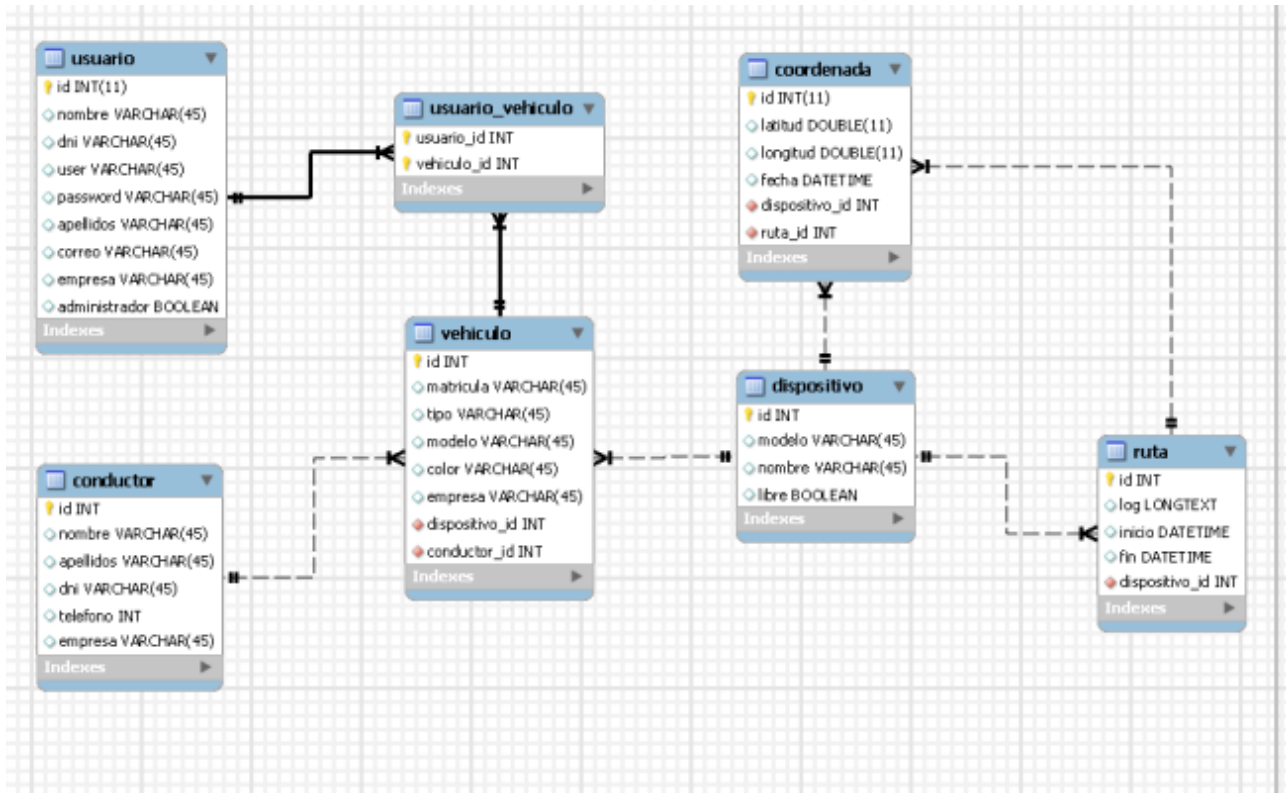


Ilustración 14. Esquema bases de datos.

6.2 Arquitectura distribuida

Una parte importante del diseño del proyecto es la elección de la arquitectura. En este proyecto se ha decidido realizar una arquitectura cliente/servidor.

Toda aplicación contiene código de presentación, código de procesamiento de base de datos y código de almacenamiento de datos. La arquitectura de las aplicaciones difiere según esta distribuido este código. La arquitectura se basa de tres:

- **Capa de Interfaz.** Esta capa soporta los servicios de presentación, los cuales proporcionan la interfaz necesaria para presentar información y reunir datos. También asegura los servicios de negocio necesarios para ofrecer las capacidades de transacciones requeridas e integrar al usuario con la aplicación para ejecutar un proceso de negocio. El cliente proporciona el contexto de interfaz, que generalmente puede ser un navegador Web, como Internet Explorer, Mozilla Firefox o Google Chrome, los cuales permiten ver los

datos remotos a través de una capa de presentación HTML. Esto ayuda a asegurar que los desarrolladores estén libres a la hora de escribir la lógica de negocios sin preocuparse acerca de cómo se muestra en la salida.

- **Capa de Negocio.** Una tarea de negocios es una operación definida por los requerimientos de la aplicación, como pasar los datos del servicio a la base de datos. Las de negocio son políticas que controlan el flujo de las tareas. Los servicios de negocio son el puente entre un usuario y los servicios de datos.

- **Capa de Datos.** La capa de datos se encuentra enlazada con la capa de negocio. El nivel de servicios de datos es responsable de:

- # Almacenar los datos.
- # Recuperar los datos.
- # Mantener los datos.
- # Integridad de los datos.

6.3 Código

En esta sección se muestra una pequeña parte del código de programación de la aplicación, ya que mostrar todo el código implicaría que la memoria fuese muy extensa.

Esta sería la parte del código que nos permite colocar el mapa de google maps en el fondo de la aplicación, asignando una serie de atributos según nuestras necesidades.

```

/*****
 * Mapa
 *****/

private MapWidget crearMapa(){

    map = new MapWidget();
    map.setWidth("800px");
    map.setHeight("400px");

    LatLng coordenadas = LatLng.newInstance(39.185632753444814, -0.38291215896606445);

    //mostramos el mapa centrado con las coordenadas (39.185632753444814, -0.38291215896606445)
    map.setCenter(coordenadas);

    //establecemos el nivel de zoom
    map.setZoomLevel(12);
    map.addMapZoomEndHandler(new MapZoomEndHandler() {

        @Override
        public void onZoomEnd(MapZoomEndEvent event) {
            map.setZoomLevel(event.getNewZoomLevel());
            limpiarMapa();
            if(vehiculosActivos.size() != 0){
                if(localizar)
                    localizar(vehiculosActivos);
                for(VehiculoModel vehiculo : vehiculosActivos){
                    if(ruta)
                        construirRuta(vehiculo, rutaModel);
                }
            }
        }
    });

    return map;
}

```

Añadimos al mapa controles y declaramos el tipo de mapa que deseamos que aparezca por defecto.

```

map.addControl(new MenuMapTypeControl());
map.addControl(new LargeMapControl3D());
map.addControl(new ScaleControl());
map.setCurrentMapType(MapType.getNormalMap());

```

6.4 Interfaz gráfica

En este apartado se representa la capa de interfaz de la aplicación que se muestra en el navegador web y en el dispositivo móvil.

6.4.1 Dispositivo móvil

La captura 1 muestra el inicio de la aplicación android, donde podemos activar el envío de coordenadas o salir de la aplicación.

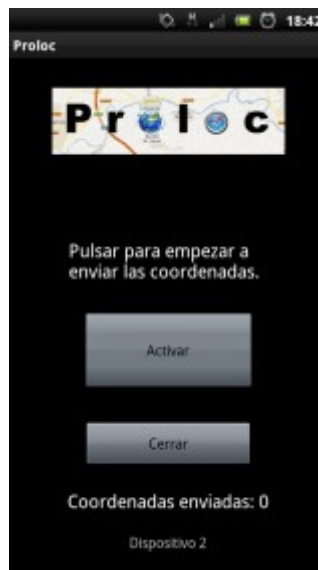


Ilustración 15 Aplicación Android pantalla inicial.

La captura 2 muestra un aviso para recordar que hay que tener activado el GPS y habilitado el acceso a los datos a través de la red móvil.



Ilustración 16 Aplicación Android aviso GPS y datos.

La captura 3 muestra un mensaje de espera mientras el dispositivo está buscando la señal de los satélites.

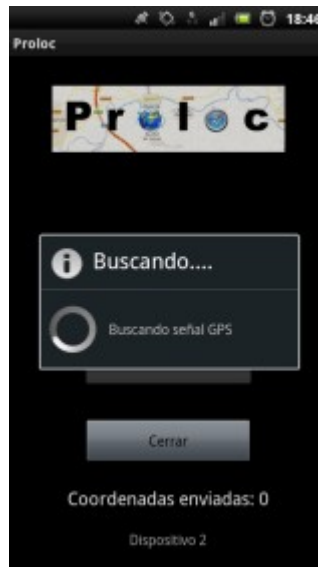


Ilustración 17 Aplicación Android buscando señal.

Cuando el dispositivo encuentra la señal, crea una ruta y envía la primera coordenada.

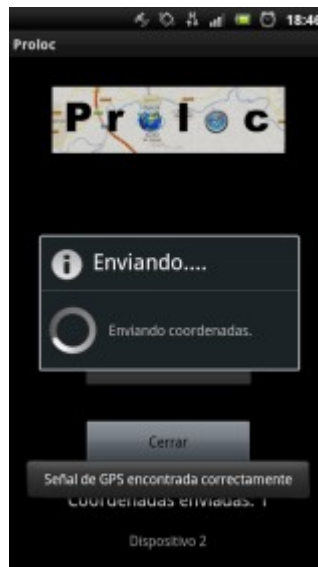


Ilustración 18 Aplicación Android señal encontrada.

La captura 5 muestra el envío periódico de coordenadas según la velocidad en la que se está desplazado el dispositivo. La periodicidad es la siguiente:

0 → 20 km/h	25 s
20 → 50 km/h	20 s
50 → 90 km/h	15 s
+90 km/h	10 s

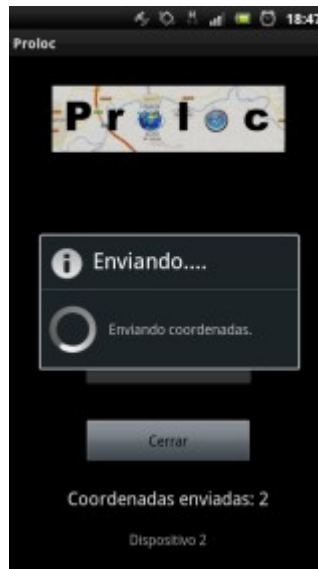


Ilustración 19 Aplicación Android envío de coordenadas.

La captura 6 muestra la pantalla inicial pero con el botón de iniciar desactivado, ya que se ha terminado la ruta al pulsar el botón de retroceso del dispositivo móvil. Para realizar una nueva ruta hay que salir de la aplicación y volverla a iniciar.

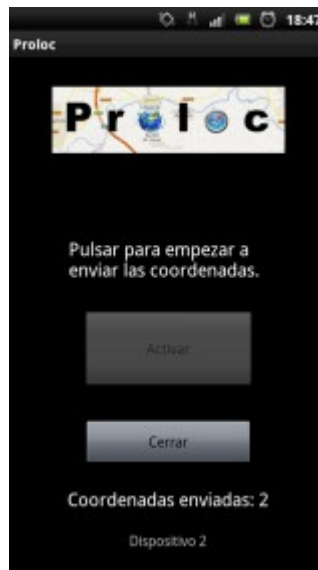


Ilustración 20 Aplicación Android fin envío de coordenadas.

6.4.2 Navegador web

En esta captura aparece la primera pantalla del programa, podemos introducir el usuario y contraseña. Si no tienes una cuenta, puedes darte de alta. También te pueden facilitar la contraseña en caso de olvido.

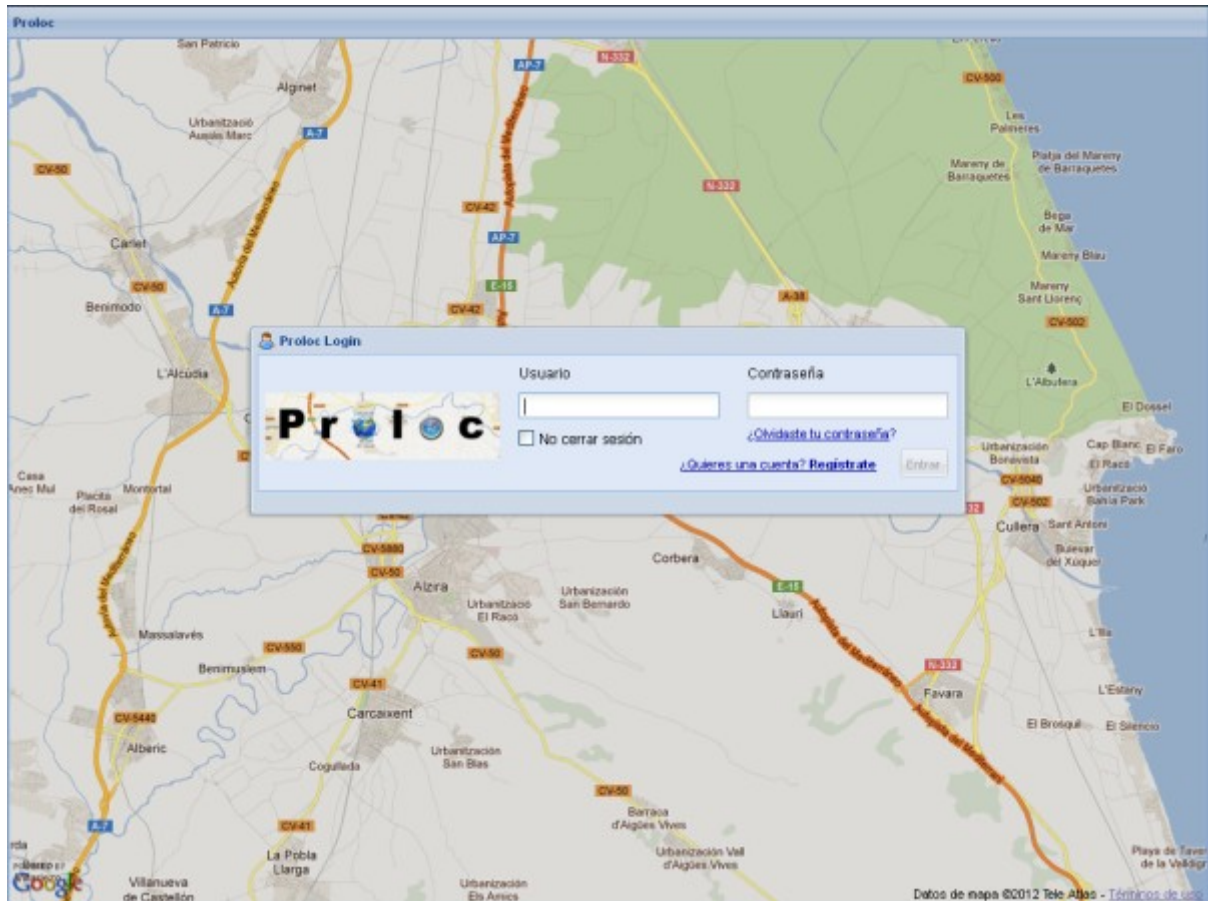
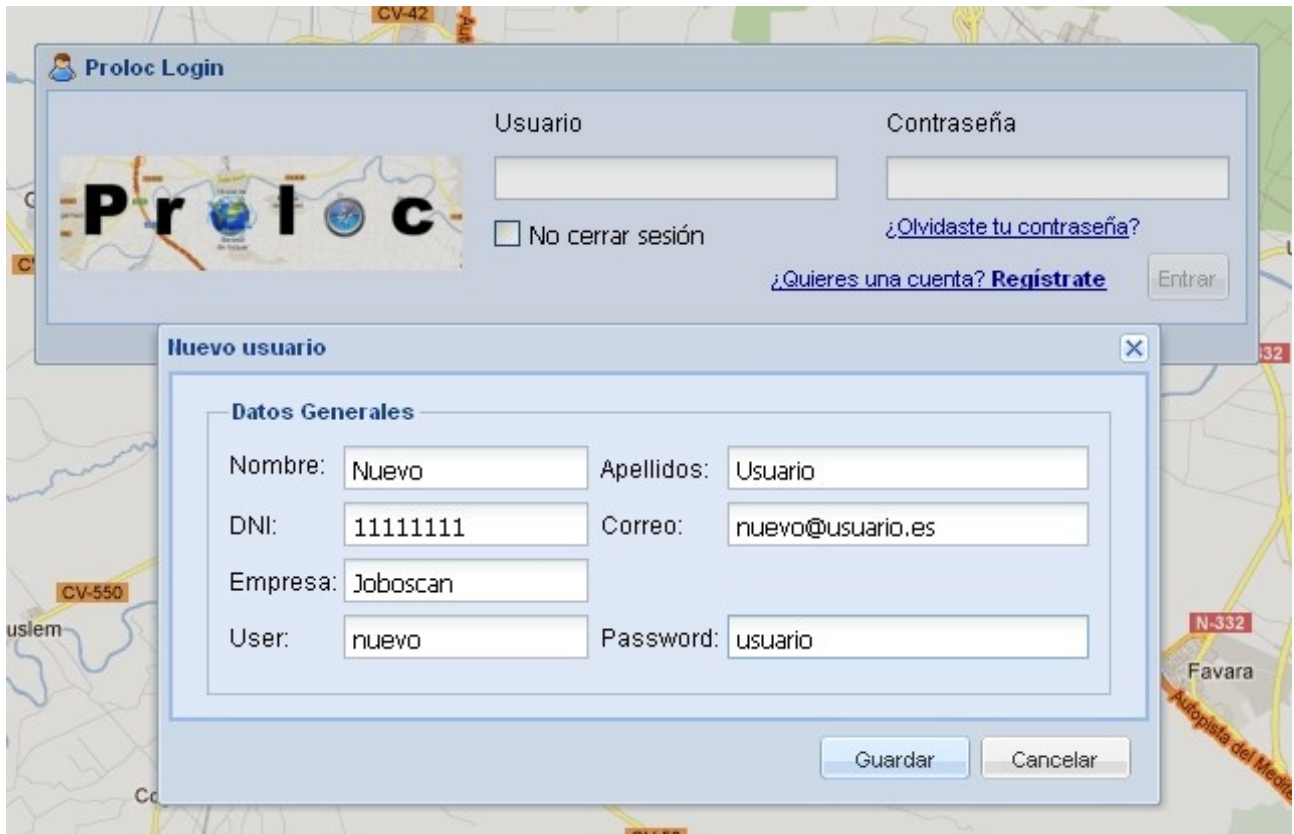


Ilustración 21 Aplicación web página inicial.

Este es el formulario de alta de usuario, donde introduces los datos del usuario para poder acceder al sistema.



The image shows a web application interface for 'Proloc Login'. It features a main login form with fields for 'Usuario' and 'Contraseña', a checkbox for 'No cerrar sesión', and links for '¿Olvidaste tu contraseña?' and '¿Quieres una cuenta? Regístrate'. An 'Entrar' button is also present. Overlaid on this is a 'Nuevo usuario' dialog box with the following fields:

Datos Generales			
Nombre:	Nuevo	Apellidos:	Usuario
DNI:	11111111	Correo:	nuevo@usuario.es
Empresa:	Joboscan		
User:	nuevo	Password:	usuario

Buttons for 'Guardar' and 'Cancelar' are located at the bottom of the dialog box.

Ilustración 22 Aplicación web formulario usuario.

Una vez introducidos los datos aparece un mensaje de aviso. Este es el de la captura 3.



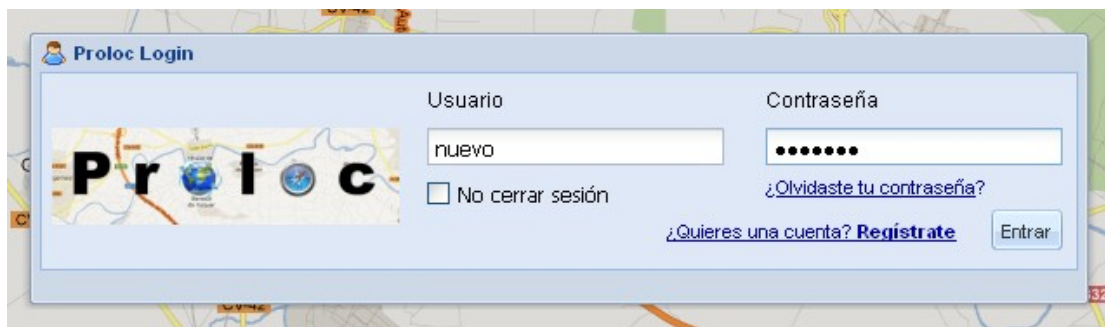
The image shows the same 'Proloc Login' interface as in the previous screenshot, but with a 'Nuevo usuario' dialog box displaying a confirmation message:

Nuevo usuario
En breves instantes recibirás un correo electrónico para confirmar el registro.

An 'OK' button is located at the bottom of the dialog box.

Ilustración 23 Aplicación web aviso envío correo electrónico.

Después de recibir el correo electrónico ya puedes acceder con tus datos en el sistema.



The screenshot shows a web browser window titled "Proloc Login". On the left is a logo for "Proloc" featuring a globe and a compass. The main form has two input fields: "Usuario" (User) containing the text "nuevo" and "Contraseña" (Password) containing seven dots. Below the user field is a checkbox labeled "No cerrar sesión". To the right of the password field is a link that says "¿Olvidaste tu contraseña?". At the bottom right of the form is a button labeled "Entrar". Below the form, there are two links: "¿Quieres una cuenta? [Regístrate](#)".

Ilustración 24 Aplicación web formulario login.

Si la contraseña o el usuario no son correctos, aparece un aviso como el de la captura 5 y si te olvidaste la contraseña te la pueden facilitar con el correo electrónico.



The screenshot shows the same "Proloc Login" interface as in the previous image, but with an error message. The "Usuario" field still contains "nuevo", but the "Contraseña" field is empty. A red error message "Autenticación fallada." is displayed below the user field. The "¿Olvidaste tu contraseña?" link is now active. A modal dialog box titled "Identifica tu cuenta" is open in the foreground. The dialog contains the "Proloc" logo and the text: "Para que podamos cambiar tu contraseña... Escribe tu dirección de correo electrónico." Below this text is an email input field containing "nuevo@usuario.es". At the bottom of the dialog are two buttons: "Aceptar" and "Cancelar".

Ilustración 25 Aplicación web olvido contraseña.

Si el usuario y la contraseña son correctos. El sistema te permite acceder a la aplicación.

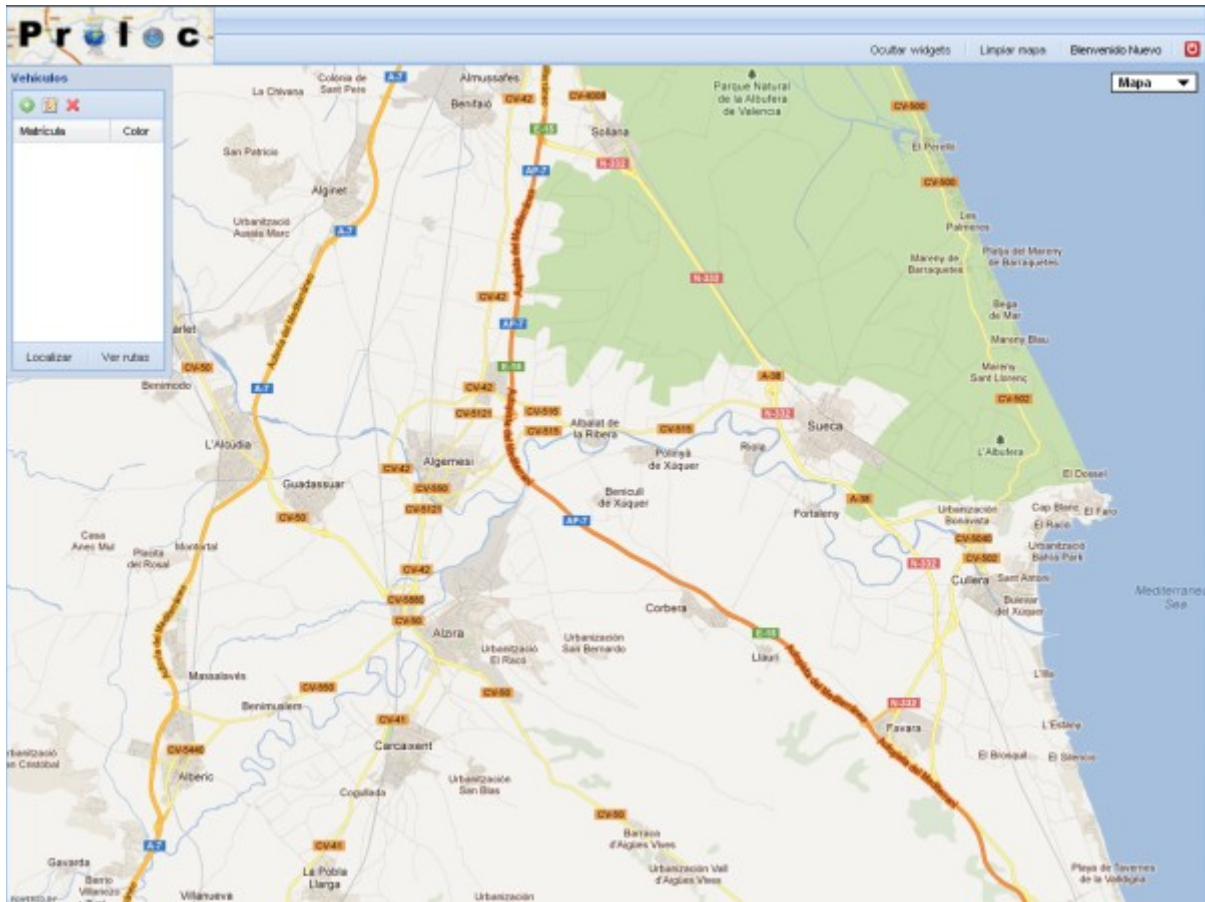


Ilustración 26 Aplicación web pantalla principal.

El menú de usuario está formado por 4 botones:

- **Ocultar widgets:** podemos ocultar todas las ventanas de la aplicación para que nos quede el mapa despejado, si lo volvemos a pulsar aparecen otra vez.
- **Limpiar mapa:** eliminamos todas las marcas que hemos añadido al mapa como pueden ser: rutas, localizaciones, etc...
- **Bienvenido:** si lo pulsas, aparece un formulario con los datos del usuario para poder modificar los datos. Ilustración 28.
- **Salir:** podemos salir de la aplicación y nos redirecciona a la primera pantalla de la aplicación.

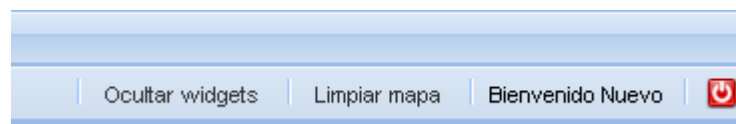
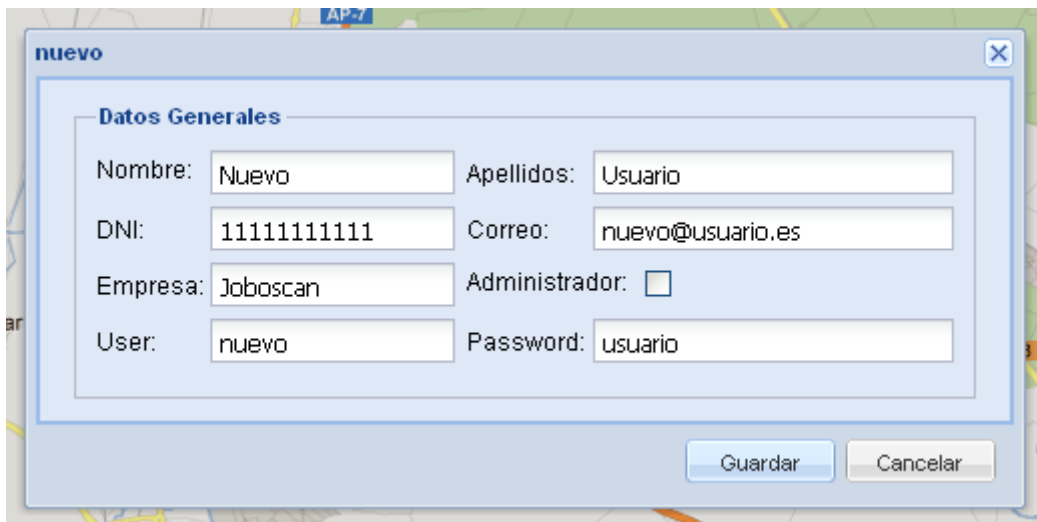


Ilustración 27 Aplicación web menú usuario.

Este es el formulario del usuario, donde se puede modificar los datos.



nuevo

Datos Generales

Nombre: Apellidos:

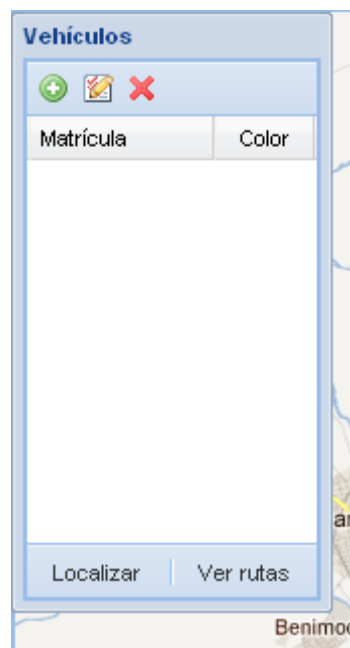
DNI: Correo:

Empresa: Administrador:

User: Password:

Ilustración 28 Aplicación web formulario modificar usuario.

Si es un usuario nuevo no tendrá ningún vehículo en la lista.



Vehiculos

Matrícula	Color
-----------	-------

Benimod

Ilustración 29 Aplicación web lista vehículos.

Con el botón añadir vehículos aparece un formulario para añadir un nuevo vehículo.

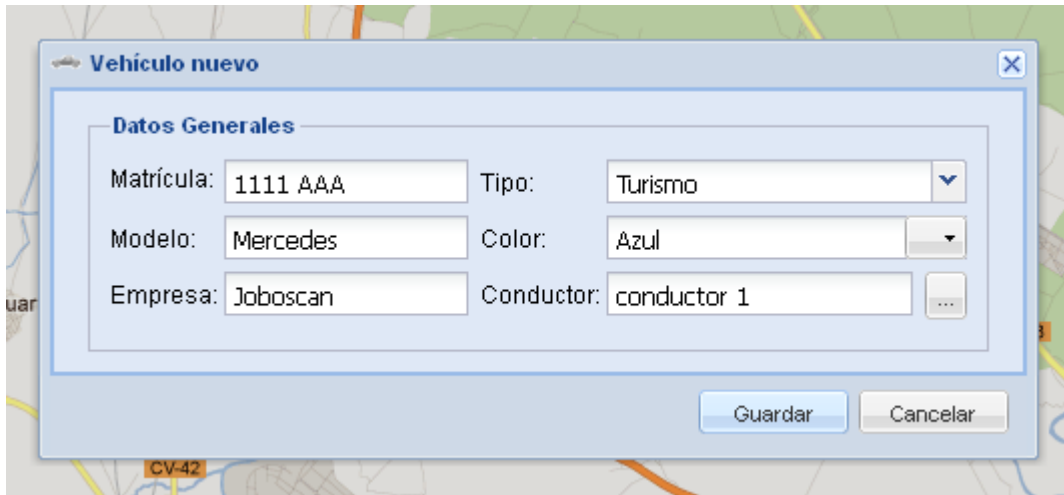


Ilustración 30 Aplicación web formulario vehículo nuevo.

Si pulsamos el botón modificar o hacemos doble clic en un vehículo de la lista aparecerá el formulario del vehículo para modificar los datos.



Ilustración 31 Aplicación web formulario modificar vehículo.

Con el botón de eliminar podemos eliminar un vehículo del sistema. Antes de eliminarlo nos muestra un aviso para confirmar el borrado.



Ilustración 32 Aplicación web confirmar el borrado de un vehículo.

Para localizar en tiempo real a un vehículo de la lista, primero tenemos que seleccionarlo y pulsar el botón “Localizar”. Nos aparece en el mapa el icono de color del vehículo que hemos seleccionado. Cada 5 segundos se refresca la posición en el mapa.

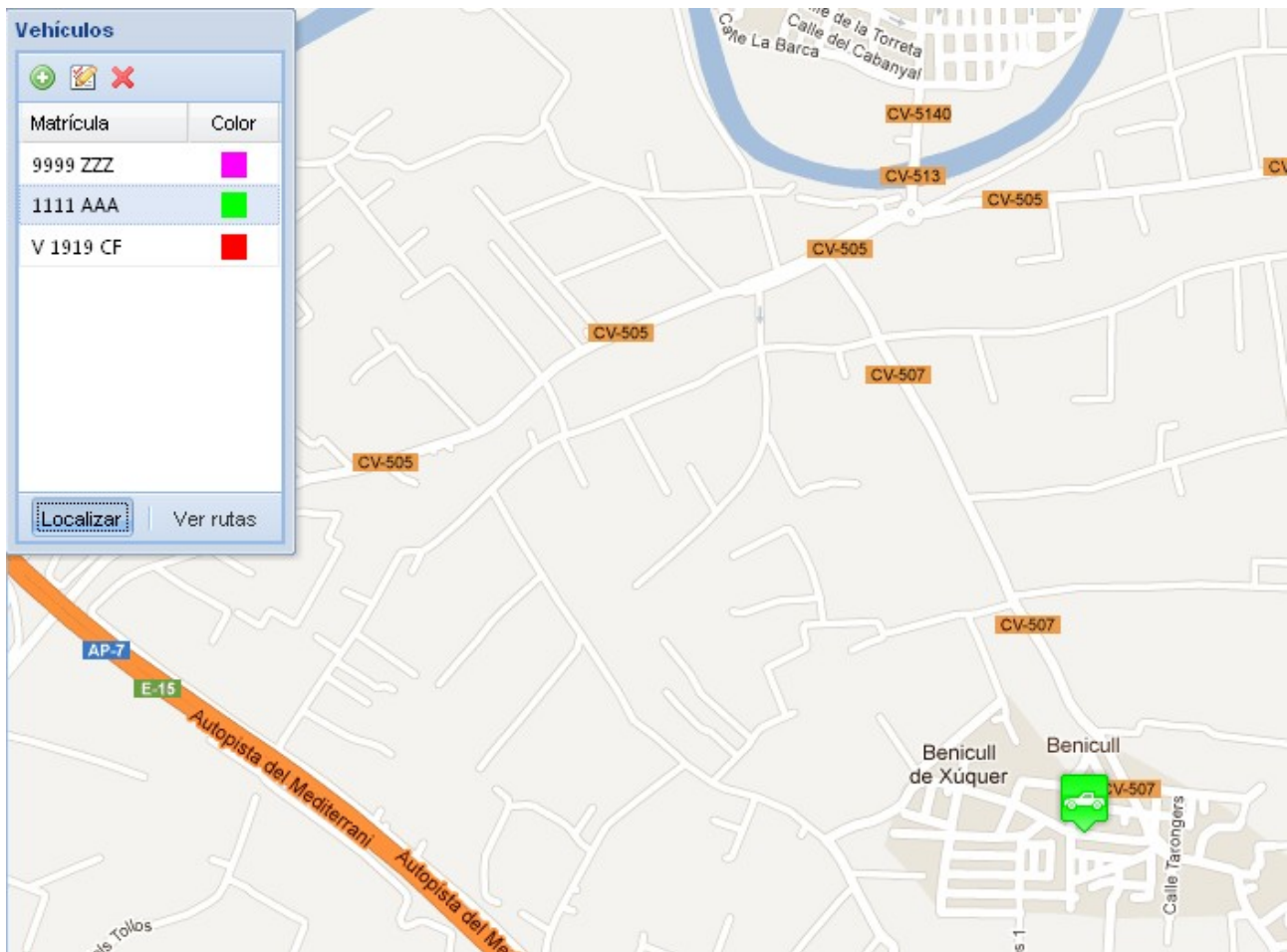


Ilustración 33 Aplicación web localizar un vehículo.

También se pueden localizar varios vehículos a la vez.

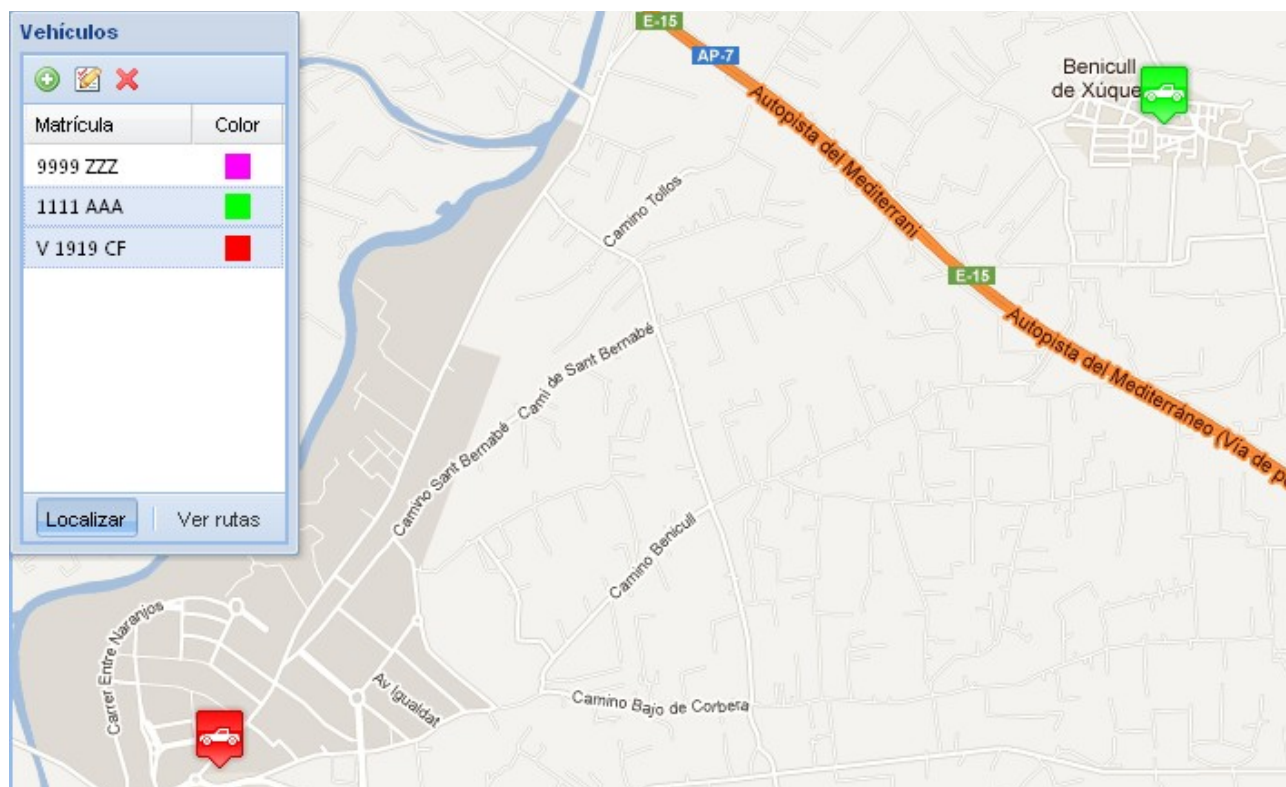


Ilustración 34 Aplicación web localizar varios vehículos.

Para ver las rutas realizadas por un vehículo, primero tenemos que seleccionarlo y hacer clic en el botón “Ver rutas”. Nos aparece una lista con todas las rutas registradas por el vehículo seleccionado. Podemos eliminar rutas (Ilustración 35), ver el log de esa ruta (Ilustración 36) y mostrarla en el mapa (Ilustración 38).

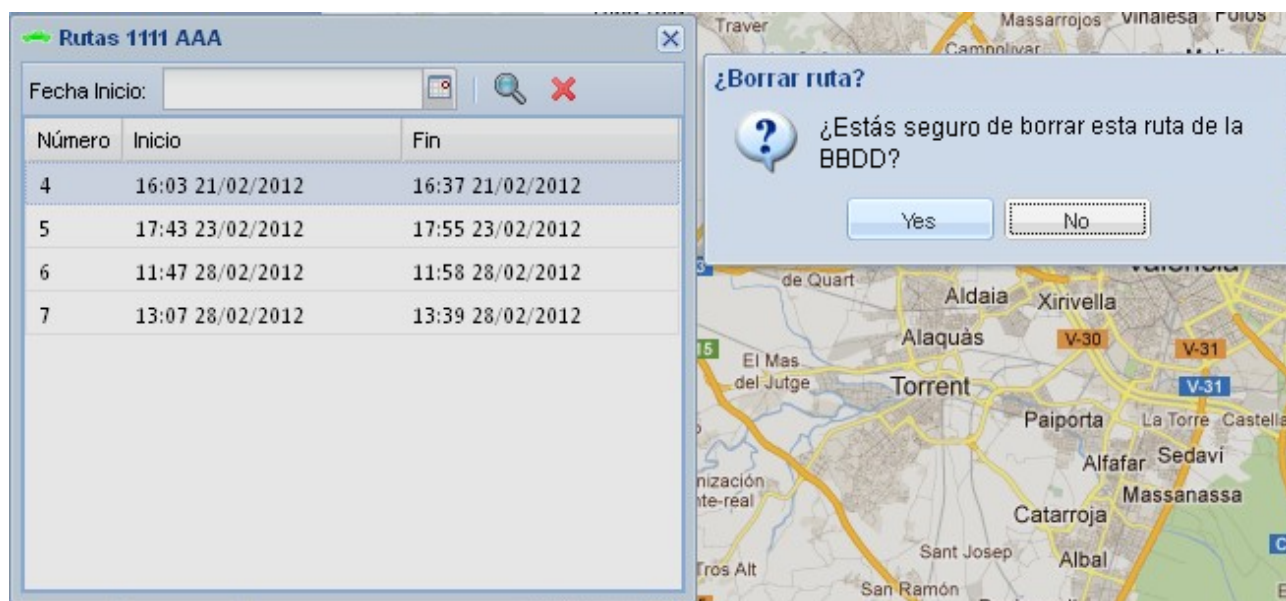


Ilustración 35 Aplicación web confirmar borrado de ruta.

Si pulsamos el botón “Ver log” aparece una ventana que nos muestra la información de la ruta, dispositivo conectado, inicio de la ruta, fin de la ruta y número de coordenadas de la ruta.

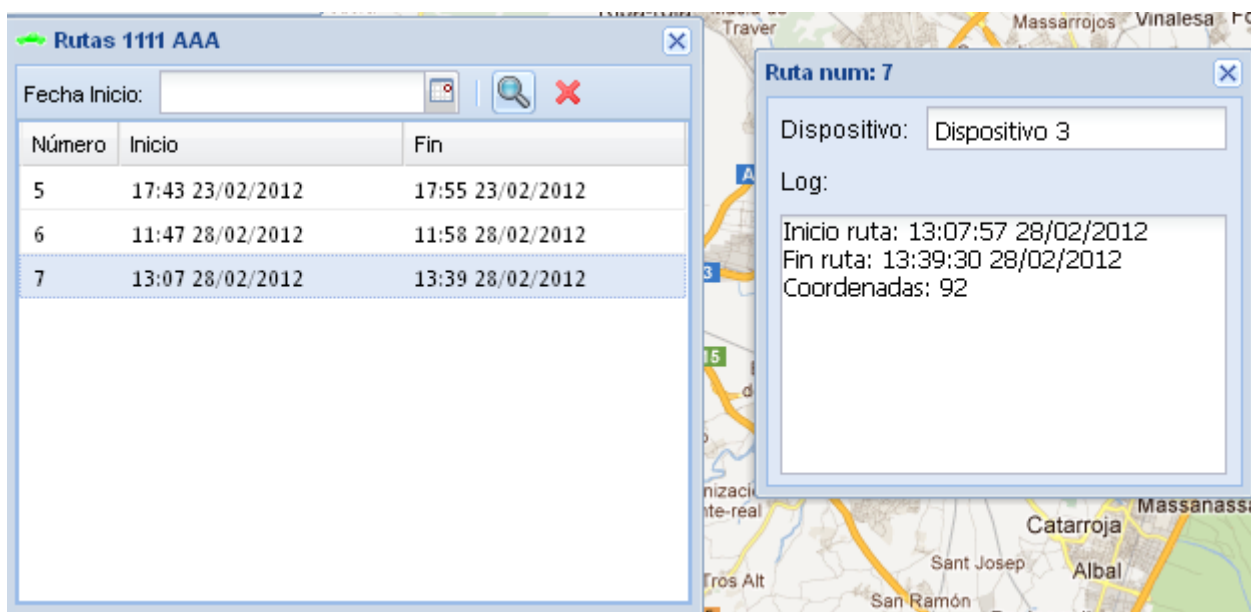


Ilustración 36 Aplicación web log de la ruta.

También se pueden filtrar las rutas según la fecha de inicio.

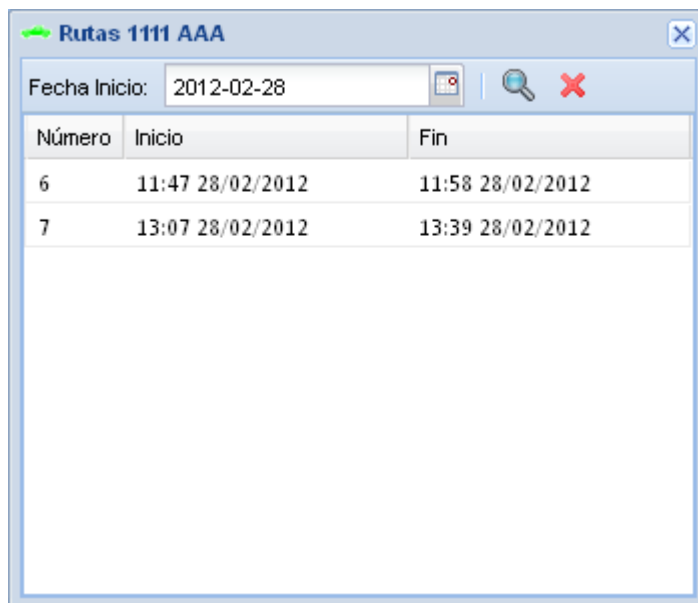


Ilustración 37 Aplicación web lista de las rutas.

Para mostrar una ruta en el mapa, hacemos doble clic en la que deseemos y nos aparece en el mapa el recorrido realizado por el vehículo. También aparece un control de ruta donde podemos situar el icono sobre la ruta, iniciar desplazamiento automático, pararlo y reiniciarlo.

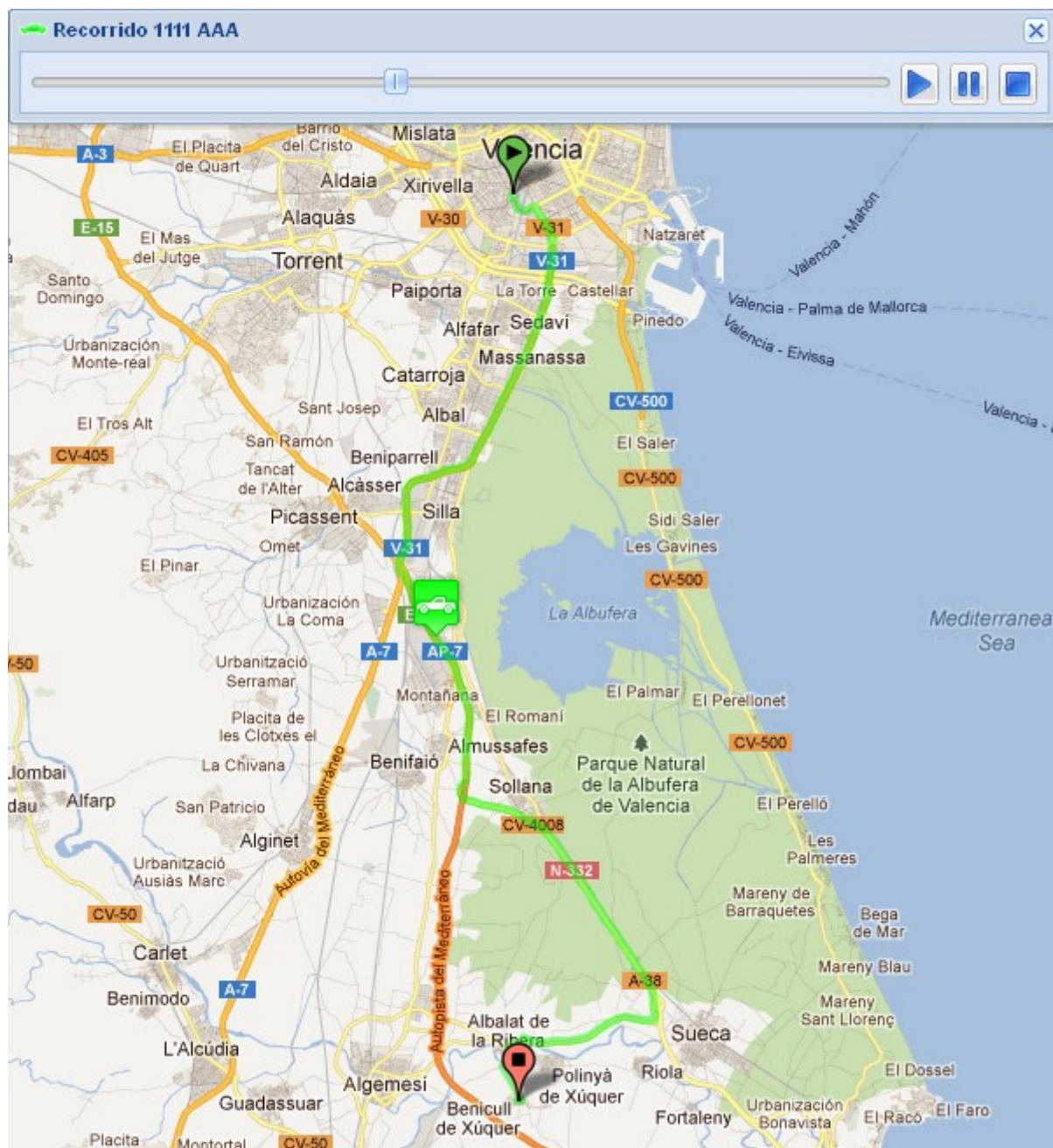


Ilustración 38 Aplicación web ruta.

Si pulsamos sobre el icono del vehículo nos aparece una ventana con la imagen de la calle donde se encuentra el vehículo en ese instante.

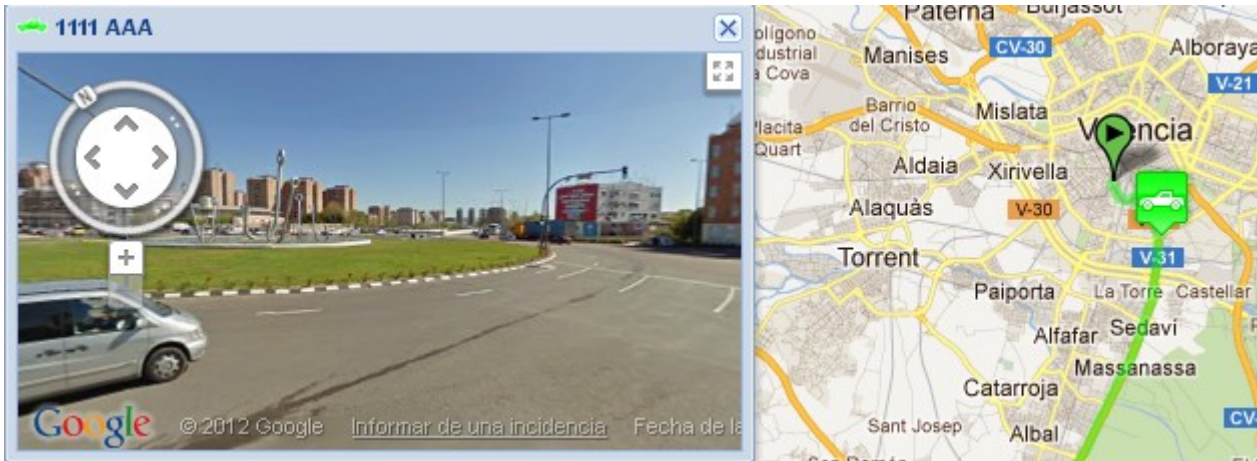


Ilustración 39 Aplicación web calle vehículo.

Si pulsamos sobre los iconos de inicio o de fin de ruta, también nos aparece la ventana con la imagen de la calle donde están situados.



Ilustración 40 Aplicación web calle fin de ruta.

Si el que accede a la aplicación es el administrador aparece un nuevo botón en el menú llamado "Gestión administrador".



Ilustración 41 Aplicación web menú administrador.

La ventana de gestión administrador contiene tres listas: dispositivos (Ilustración 42), conductores (Ilustración 45) y rutas (Ilustración 48).

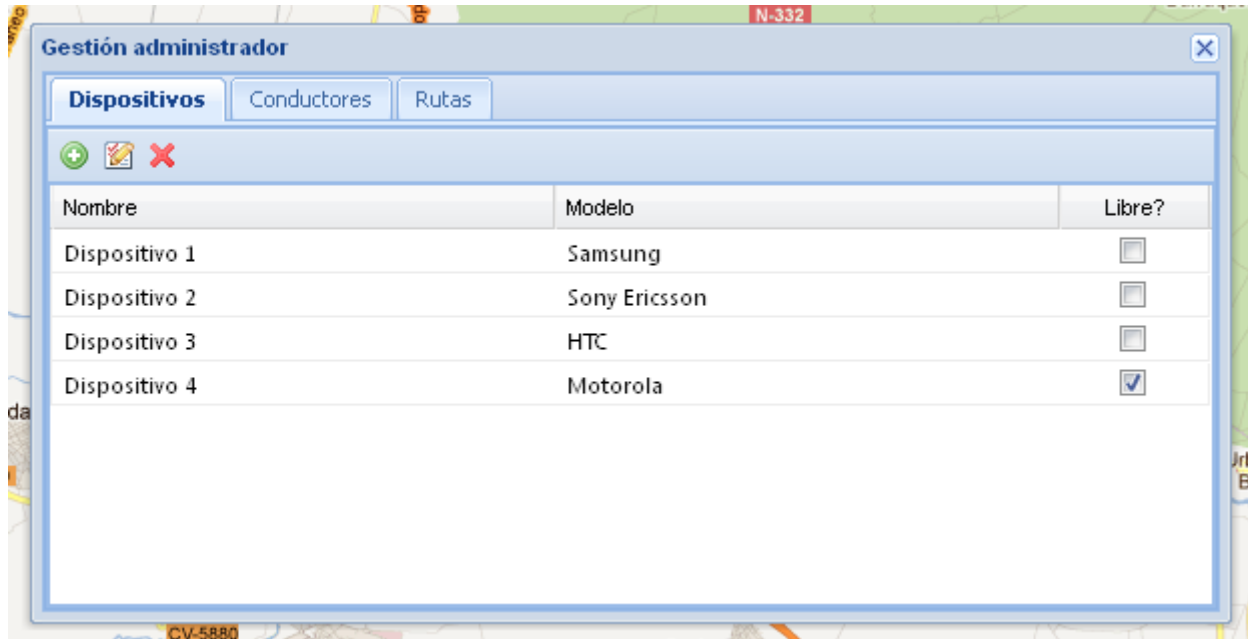


Ilustración 42 Aplicación web lista dispositivos.

Cuando se añade un nuevo dispositivo automáticamente se pone libre para que cuando se añada un nuevo vehículo el sistema le asigne el primer dispositivo libre. Pulsando el botón modificar o haciendo doble clic sobre el dispositivo podemos modificar los datos del dispositivo.

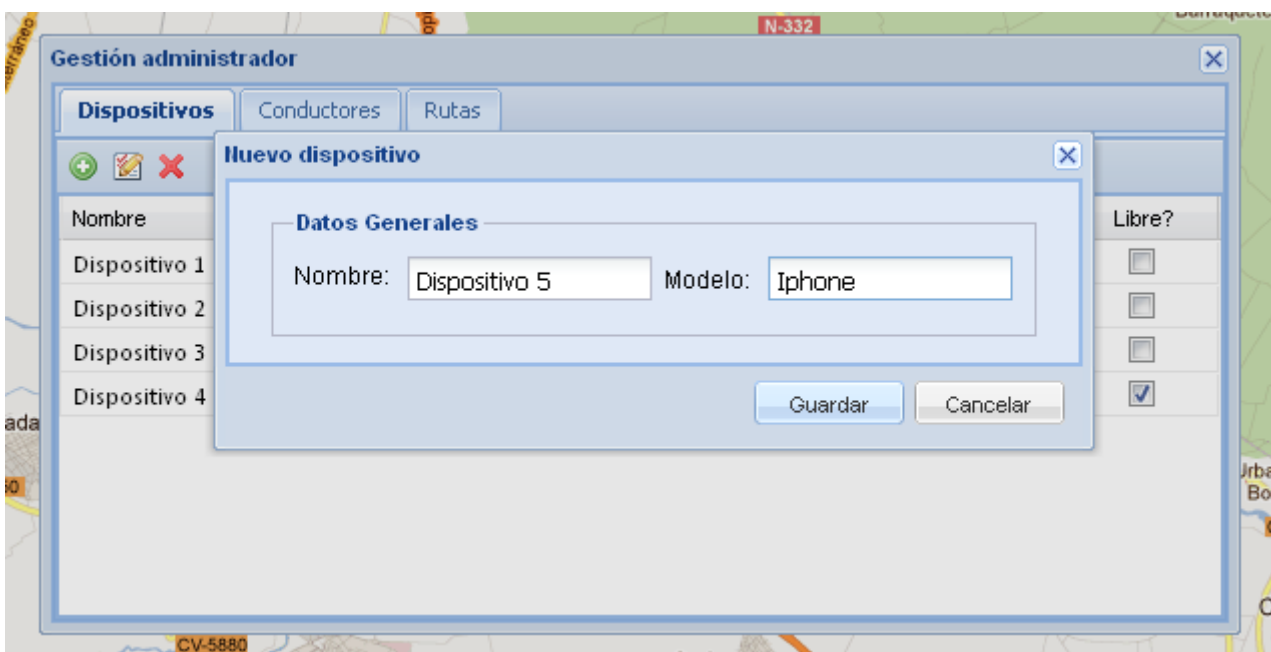


Ilustración 43 Aplicación web formulario dispositivo.

Si pulsas el botón borrar puedes eliminar un dispositivo siempre y cuando no esté ocupado.

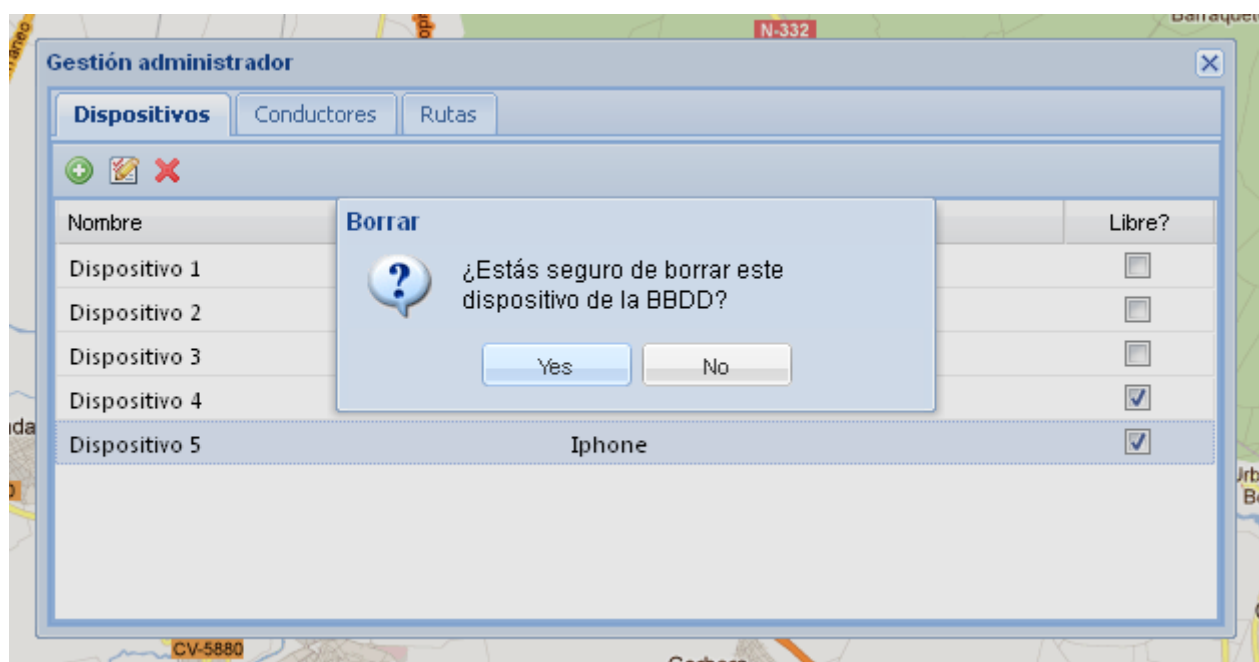


Ilustración 44 Aplicación web confirmación borrado dispositivo.

Lista de los conductores disponibles

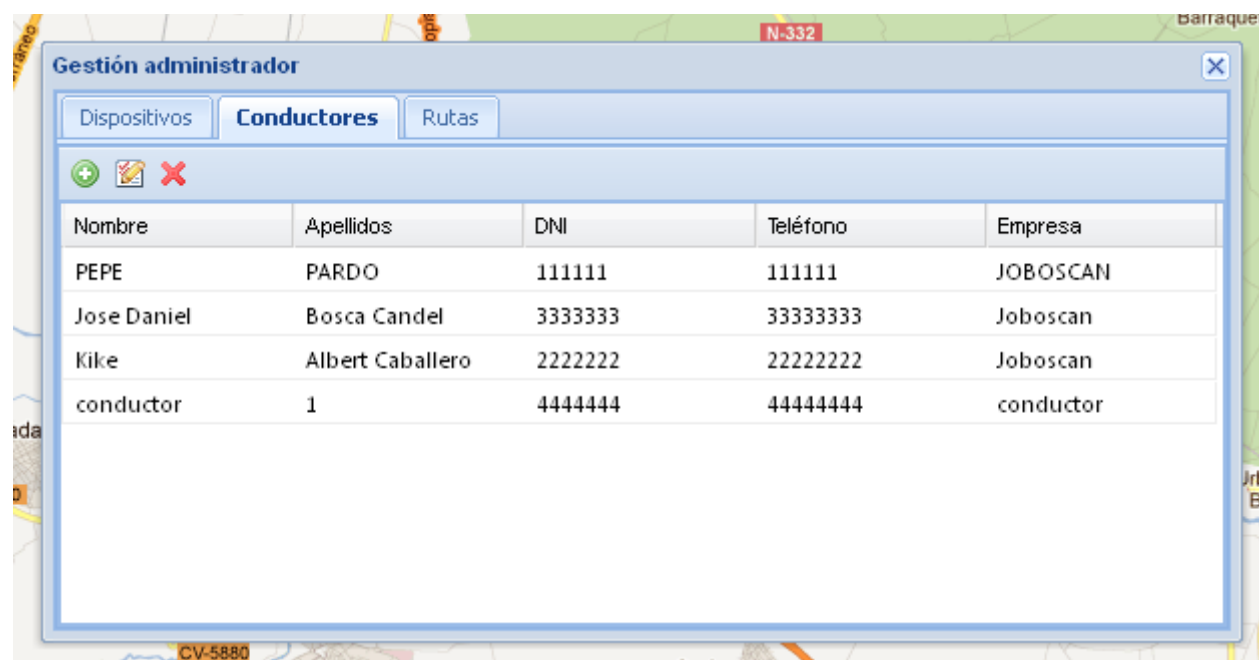


Ilustración 45 Aplicación web lista conductores.

Si haces clic en el botón añadir conductor aparece un formulario donde introduces los datos del conductor. Si haces doble clic sobre un conductor o pulsas el botón modificar también aparece el formulario para modificar los datos.

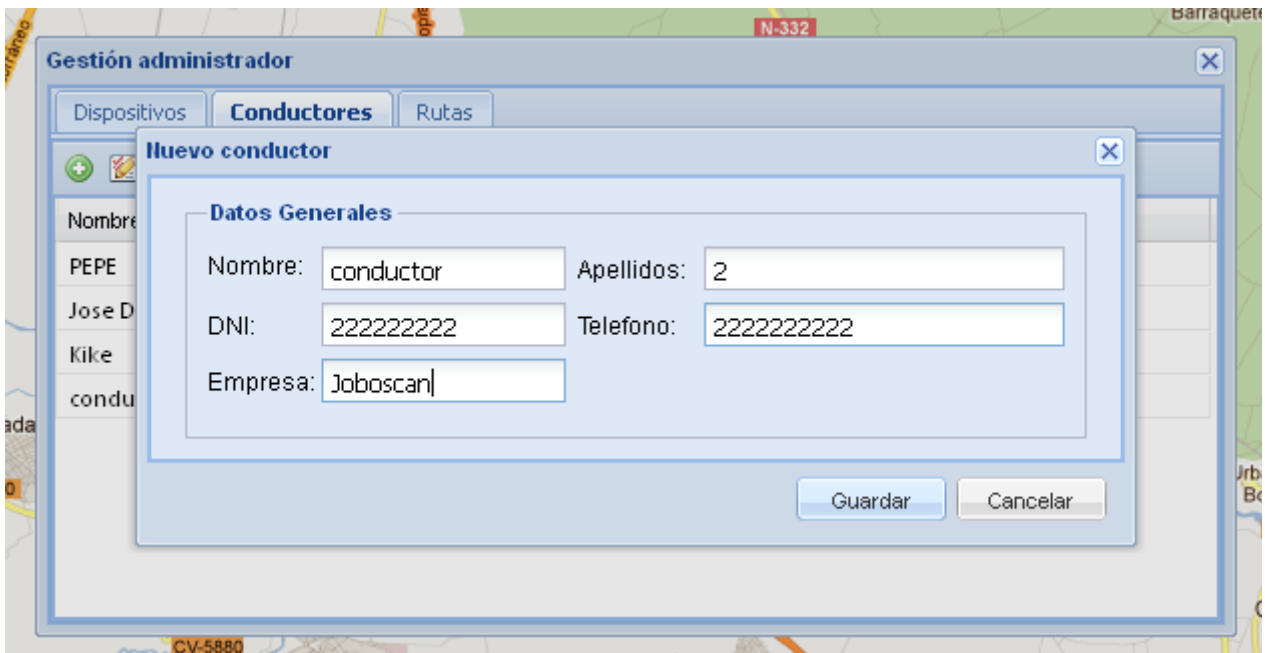


Ilustración 46 Aplicación web formulario conductor.

Si pulsas el botón borrar puedes eliminar un conductor siempre y cuando no esté asignado a un vehículo.

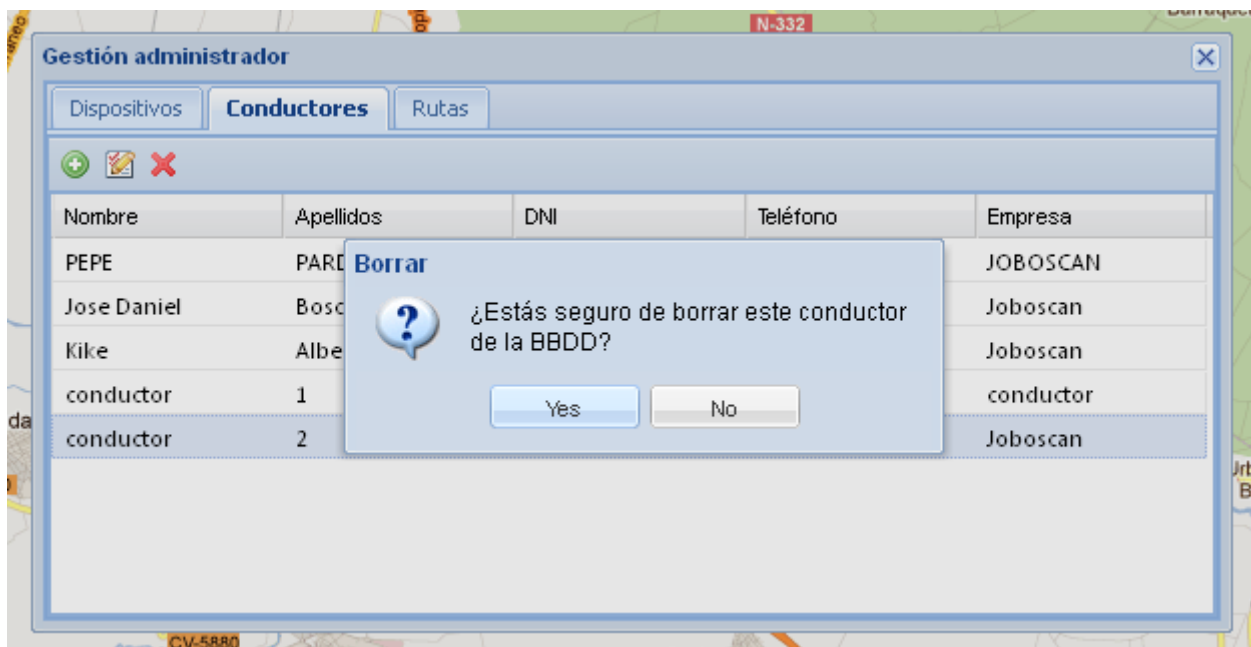
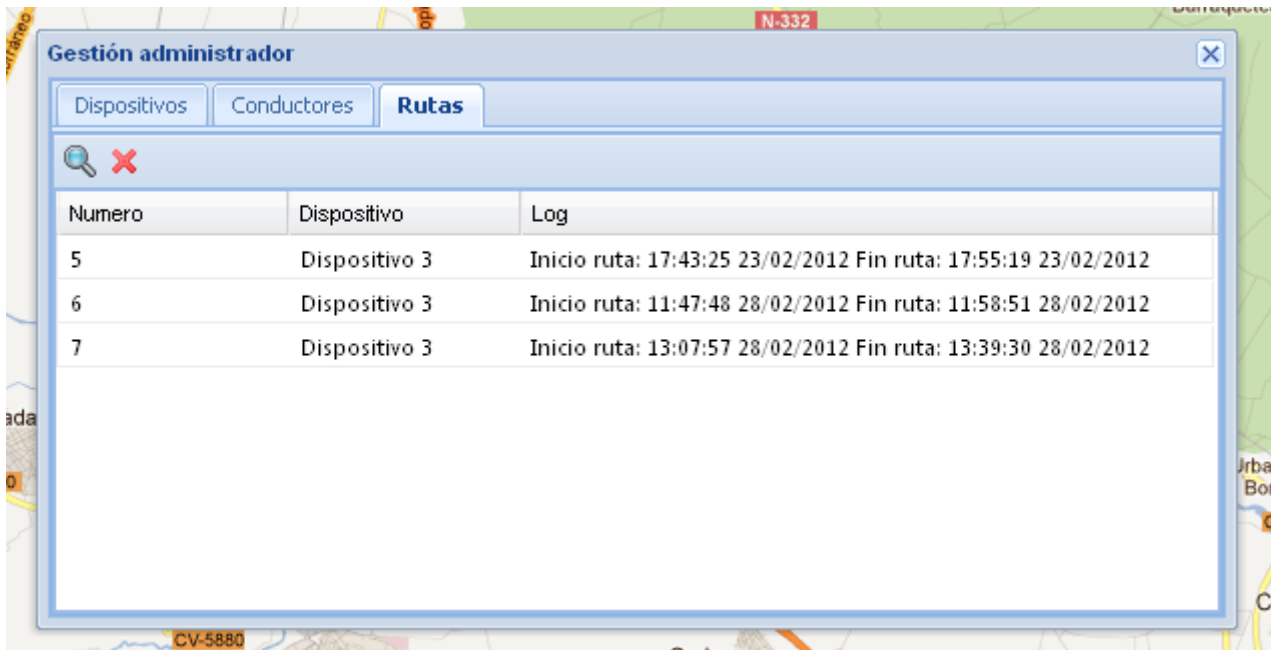


Ilustración 47 Aplicación web confirmación borrado conductor.

Listado de las rutas de todos los dispositivos. Podemos eliminar rutas y ver el log de las rutas.



Numero	Dispositivo	Log
5	Dispositivo 3	Inicio ruta: 17:43:25 23/02/2012 Fin ruta: 17:55:19 23/02/2012
6	Dispositivo 3	Inicio ruta: 11:47:48 28/02/2012 Fin ruta: 11:58:51 28/02/2012
7	Dispositivo 3	Inicio ruta: 13:07:57 28/02/2012 Fin ruta: 13:39:30 28/02/2012

Ilustración 48 Aplicación web lista rutas administrador.

7. Etapas de la aplicación

7.1 Fase previa

El primer paso de la fase previa es la elección de las herramientas para la elaboración del proyecto como el entorno de programación, la base de datos, la conexión con el servidor, el lenguaje de programación, dispositivos, etc...

También hay que investigar los productos existentes y como funcionan. Saber la cantidad de productos relacionados que existen y la aceptación del mercado que tiene.

7.2 Primera fase

Una vez elegidas las herramientas, llega el proceso de adaptación y documentación de las herramientas. Las herramientas elegidas son:

- Entorno de programación: Eclipse SDK Versión: 3.7.1
- Lenguaje de programación: GWT y Android (ambos JAVA).
- Interfaz de usuario: EXT GWT.
- Conexión servidor: Sockets.
- Bases de datos: MySQL.
- Conexión a la base de datos: Hibernate.
- Tratamiento de datos: Spring.

Elaboración de la especificación de requisitos del cliente y del desarrollador informático.

7.3 Segunda fase

Con la especificación de requisitos en mano se pasa a hacer toda la parte de análisis del proyecto como diagramas de clases, casos de uso, diagramas de secuencias, etc...

7.4 Tercera fase

Según las pautas del análisis hay que pasar al diseño de la bases de datos. Diseñar los posibles formularios en un papel para la posterior implementación.

7.5 Cuarta fase

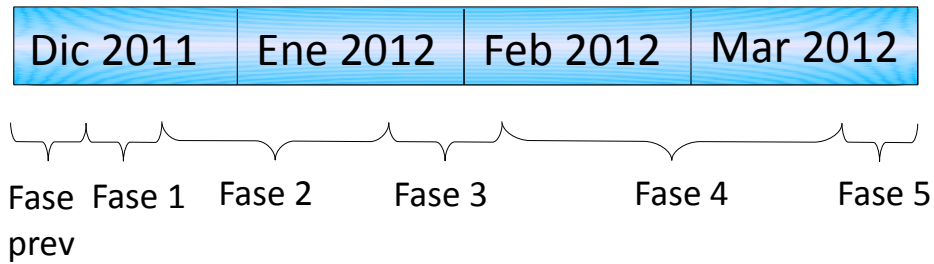
Empieza la codificación de la aplicación web y de la aplicación Android. Para ello utilizaremos los plugins correspondientes para Eclipse. Para la construcción de un proyecto web utilizamos el plugin de GWT, para la interfaz gráfica el plugin Designer de GWT, el mapa que utilizamos es la versión 2 de google maps con la api para GWT y para la aplicación Android utilizamos el plugin de Android con la máquina virtual para probar las aplicaciones. Mediante un plugin de Hibernate para Eclipse crearemos las clases necesarias para la aplicación.

Siguiendo las pautas del análisis implementaremos toda la funcionalidad de la aplicación web y la aplicación móvil.

7.6 Quinta fase

Después de implementar las aplicaciones, ponemos en marcha las aplicaciones y pasamos a la etapa de pruebas donde comprobaremos si las aplicaciones funcionan correctamente.

7.7 Cronología



8. Conclusiones

8.1 Conclusiones

Para verificar que el proyecto se ha realizado de acuerdo a lo esperado, basta con comprobar que se han alcanzado con éxito los objetivos que se marcaron al inicio del proyecto.

Los administradores disponen de una aplicación Web en la que puedan crear, gestionar, borrar y modificar rutas, vehículos, conductores y dispositivos de una manera fácil usando interfaces tan simples como la selección del objeto con una lista diferenciando claramente cada uno de los campos. Este objetivo ha sido superado con éxito, ya que después de pasar las pruebas, no se ha detectado ninguna anomalía.

El usuario y el administrador dispondrán de una aplicación Web en la que podrá visualizar las rutas existentes en la base de datos. Este objetivo ha sido superado con éxito, ya que después de pasar las pruebas, no se ha detectado ninguna anomalía.

El conductor dispondrá de una aplicación Android con la que automáticamente se enviará las coordenadas del GPS a la base de datos a través del servidor Web con el objetivo de tener el seguimiento a tiempo real y la ruta que esté realizando. Este objetivo ha sido superado con éxito, ya que después de pasar las pruebas reales con un dispositivo móvil, no se ha detectado ninguna anomalía y las coordenadas se han guardado en la base de datos correctamente.

8.2 Futuras ampliaciones

Las futuras ampliaciones de este proyecto tienen relación con los futuros cambios, actualizaciones de las APIs utilizadas en este proyecto, ampliaciones de las aplicaciones y mejoras de rendimiento.

- Actualmente el mapa es la versión 2, se puede migrar a la versión 3 pero hoy por hoy aun no hay una API estable para GWT.
- La aplicación móvil puede desarrollarse también para la plataforma iOS y así funcionar también en los dispositivos iPhone.
- El dispositivo también puede enviar la velocidad y así poder estimar el combustible consumido por el vehículo.
- Introducir puntos de interés en el mapa.

9. Referencias

- **Wikipedia** → <http://es.wikipedia.org/wiki/Wikipedia:Portada>
- **Mapa para GWT** → <http://code.google.com/p/gwt-google-apis/wiki/MapsGettingStarted>
- **Proyecto GWT** → <http://code.google.com/intl/es-ES/webtoolkit/>
- **Api EXT GWT** → <http://www.sencha.com/gxtdocs/>
- **Hibernate** → <http://www.hibernate.org/>
- **Spring** → <http://www.springsource.org/>
- **Android** → <http://developer.android.com/index.html>

10. Apéndice

10.1 Plantillas casos de uso

<u>Caso de uso</u>	Añadir vehículo
Actores	Usuario
Propósito	Añadir un vehículo en el sistema para localizarlo.
Resumen	El usuario pulsa el botón añadir vehículo. Aparece un formulario para rellenar los datos del vehículo. Una vez guardado, el nuevo vehículo aparecerá en la tabla de vehículos disponibles del usuario.
Pre-condiciones	El usuario se ha identificado en el sistema.
Post-condiciones	El vehículo se almacena en el sistema.
Usuario	Sistema
1. El usuario pulsa el botón añadir vehículo.	
	2. El sistema muestra el formulario añadir vehículo.
3. El usuario rellena los datos del vehículo.	
	4. El sistema comprueba si existe o hay algún error.
	5. El sistema muestra una ventana de confirmación.
6. El usuario confirma	
	7. El sistema almacena el nuevo vehículo y le asigna un dispositivo libre.

<u>Caso de uso</u>	Modificar vehículo
Actores	Usuario
Propósito	Modificar los datos de un vehículo del sistema.
Resumen	El usuario pulsa el botón modificar vehículo. Aparece un formulario con los datos rellenos del vehículo para poder modificar los datos necesarios por el usuario. Una vez guardado, el vehículo modificado aparecerá en la tabla de vehículos disponibles del usuario.
Pre-condiciones	El usuario se ha identificado en el sistema.
Post-condiciones	El vehículo se almacena en el sistema.
Incluye	Buscar vehículo.
Usuario	Sistema
1. El usuario selecciona el vehículo.	
2. El usuario pulsa el botón modificar vehículo.	3. El sistema muestra el formulario modificar vehículo con los datos rellenos.
4. El usuario modifica los datos del vehículo.	
	5. El sistema comprueba si hay algún error.
	6. El sistema muestra una ventana de confirmación.
7. El usuario confirma.	
	8. El sistema almacena el vehículo modificado.

<u>Caso de uso</u>	Eliminar vehículo
Actores	Usuario
Propósito	Eliminar un vehículo del sistema.
Resumen	El usuario pulsa el botón eliminar vehículo. Aparece una confirmación para eliminar. Una vez eliminado, el vehículo desaparecerá de la tabla de vehículos disponibles del usuario.
Pre-condiciones	El usuario se ha identificado en el sistema.
Post-condiciones	El vehículo se elimina del sistema.
Incluye	Buscar vehículo.
Usuario	Sistema
1. El usuario selecciona el vehículo.	
2. El usuario pulsa el botón eliminar vehículo.	3. El sistema muestra una ventana de confirmación.
4. El usuario confirma.	
	5. El sistema elimina el vehículo seleccionado.

<u>Caso de uso</u>	Modificar usuario
Actores	Usuario
Propósito	Modificar los datos del usuario.
Resumen	El usuario pulsa sobre su nombre. Aparece un formulario con los datos rellenos del usuario para poder modificar los datos necesarios por el usuario.
Pre-condiciones	El usuario se ha identificado en el sistema.
Post-condiciones	El usuario se almacena en el sistema.
Usuario	Sistema
1. El usuario pulsa sobre su nombre.	3. El sistema muestra el formulario modificar usuario con los datos rellenos.
4. El usuario modifica los datos necesarios.	
	5. El sistema comprueba si hay algún error.
	6. El sistema muestra una ventana de confirmación.
7. El usuario confirma.	
	8. El sistema almacena el usuario modificado.

<u>Caso de uso</u>	Localizar vehículo
Actores	Usuario
Propósito	Localizar un vehículo del sistema.
Resumen	El usuario pulsa el botón localizar vehículo. En el mapa aparecerá un marcador indicando en el sitio exacto donde se encuentra el vehículo seleccionado.
Pre-condiciones	El usuario se ha identificado en el sistema.
Incluye	Seleccionar vehículo.
Usuario	Sistema
1. El usuario selecciona el vehículo.	
2. El usuario pulsa el botón localizar vehículo.	3. El sistema obtiene las coordenada del dispositivo asignado al vehículo.
	4. El sistema coloca un marcador en el mapa y refresca la posición cada 5 segundos.

<u>Caso de uso</u>	Ver rutas vehículo
Actores	Usuario
Propósito	El usuario puede ver las rutas de un vehículo del sistema.
Resumen	El usuario pulsa el botón ver ruta vehículo. Aparece una ventana con las rutas realizadas por el vehículo. Al seleccionar una en el mapa aparecerá una línea que indica los lugares por donde se ha desplazado el vehículo .
Pre-condiciones	El usuario se ha identificado en el sistema.
Incluye	Seleccionar vehículo.
Usuario	Sistema
1. El usuario selecciona el vehículo.	
2. El usuario pulsa el botón ver ruta vehículo.	3. El sistema muestra una ventana con todas las rutas realizadas por el vehículo.
4. El usuario selecciona la ruta deseada.	5. El sistema dibuja la ruta siguiendo las coordenadas cronológicamente.

<u>Caso de uso</u>	Iniciar sesión
Actores	Usuario
Propósito	El usuario accede al sistema.
Resumen	El usuario accede al sistema para utilizar el programa.
Post-condiciones	El usuario se ha identificado en el sistema.
Usuario	Sistema
1. El usuario introduce la url de la aplicación.	
	2. El sistema muestra un formulario de usuario y contraseña.
3. El usuario introduce su nombre y contraseña.	4. El sistema comprueba si el usuario existe.
	5. El sistema muestra la página principal de la aplicación.

<u>Caso de uso</u>	Cerrar sesión
Actores	Usuario
Propósito	El usuario sale del sistema.
Resumen	El usuario sale del sistema porque ha terminado.
Pre-condiciones	El usuario se ha identificado en el sistema.
Usuario	Sistema
1. El usuario pulsa el botón salir.	
	2. El sistema muestra un formulario de confirmación.
3. El usuario confirma.	4. El sistema muestra el formulario de login y contraseña.

<u>Caso de uso</u>	Iniciar sesión
Actores	Administrador
Propósito	El administrador accede al sistema.
Resumen	El administrador accede al sistema para utilizar el programa.
Post-condiciones	El administrador se ha identificado en el sistema.
Usuario	Sistema
1. El administrador introduce la url de la aplicación.	
	2. El sistema muestra un formulario de login y contraseña.
3. El administrador introduce su nombre y contraseña.	4. El sistema comprueba si el administrador existe.
	5. El sistema muestra la página principal de la aplicación.

<u>Caso de uso</u>	Cerrar sesión
Actores	Administrador
Propósito	El administrador sale del sistema.
Resumen	El administrador sale del sistema porque ha terminado.
Pre-condiciones	El administrador se ha identificado en el sistema.
Usuario	Sistema
1. El administrador pulsa el botón salir.	
	2. El sistema muestra un formulario de confirmación.
3. El administrador confirma.	4. El sistema muestra el formulario de login y contraseña.

<u>Caso de uso</u>	Modificar administrador
Actores	Administrador
Propósito	Modificar los datos del administrador.
Resumen	El administrador pulsa el botón modificar administrador. Aparece un formulario con los datos rellenos del administrador para poder modificar los datos necesario por el administrador.
Pre-condiciones	El administrador se ha identificado en el sistema.
Post-condiciones	El administrador se almacena en el sistema.
Usuario	Sistema
1. El administrador pulsa el botón modificar administrador.	3. El sistema muestra el formulario modificar administrador con los datos rellenos.
4. El administrador modifica los datos necesarios.	
	5. El sistema comprueba si hay algún error.
	6. El sistema muestra una ventana de confirmación.
7. El administrador confirma.	
	8. El sistema almacena el administrador modificado.

<u>Caso de uso</u>	Añadir dispositivo
Actores	Administrador
Propósito	Añadir un dispositivo en el sistema para asignarlo a un vehículo.
Resumen	El administrador pulsa el botón añadir dispositivo. Aparece un formulario para rellenar los datos del dispositivo. Una vez guardado, el nuevo dispositivo aparecerá en la tabla de dispositivos disponibles del administrador.
Pre-condiciones	El administrador se ha identificado en el sistema.
Post-condiciones	El dispositivo se almacena en el sistema.
Usuario	Sistema
1. El administrador pulsa el botón añadir dispositivo.	
	2. El sistema muestra el formulario añadir dispositivo.
3. El administrador rellena los datos del dispositivo.	
	4. El sistema comprueba si existe o hay algún error.
	5. El sistema muestra una ventana de confirmación.
6. El administrador confirma	
	7. El sistema almacena el nuevo dispositivo.

<u>Caso de uso</u>	Modificar dispositivo
Actores	Administrador
Propósito	Modificar los datos de un dispositivo del sistema.
Resumen	El administrador pulsa el botón modificar dispositivo. Aparece un formulario con los datos rellenos del dispositivo para poder modificar los datos necesarios por el administrador. Una vez guardado, el dispositivo modificado aparecerá en la tabla de dispositivos disponibles del administrador.
Pre-condiciones	El administrador se ha identificado en el sistema.
Post-condiciones	El dispositivo se almacena en el sistema.
Incluye	Buscar dispositivo.
Usuario	Sistema
1. El administrador selecciona el dispositivo.	
2. El administrador pulsa el botón modificar dispositivo.	3. El sistema muestra el formulario modificar dispositivo con los datos rellenos.
4. El administrador modifica los datos del dispositivo.	
	5. El sistema comprueba si hay algún error.
	6. El sistema muestra una ventana de confirmación.
7. El administrador confirma.	
	8. El sistema almacena el dispositivo modificado.

<u>Caso de uso</u>	Eliminar dispositivo
Actores	Administrador
Propósito	Eliminar un dispositivo del sistema.
Resumen	El administrador pulsa el botón eliminar dispositivo. Si el dispositivo esta libre aparece una confirmación para eliminar. Una vez eliminado, el dispositivo desaparecerá de la tabla de dispositivos disponibles del administrador.
Pre-condiciones	El administrador se ha identificado en el sistema.
Post-condiciones	El dispositivo se elimina del sistema.
Incluye	Buscar dispositivo.
Usuario	Sistema
1. El administrador selecciona el dispositivo.	
2. El administrador pulsa el botón eliminar dispositivo.	3. El sistema comprueba si esta libre y muestra una ventana de confirmación.
4. El administrador confirma.	
	5. El sistema elimina el dispositivo seleccionado.

<u>Caso de uso</u>	Añadir conductor
Actores	Administrador
Propósito	Añadir un conductor en el sistema.
Resumen	El administrador pulsa el botón añadir conductor. Aparece un formulario para rellenar los datos del conductor. Una vez guardado, el nuevo conductor aparecerá en la tabla de conductores disponibles del administrador.
Pre-condiciones	El administrador se ha identificado en el sistema.
Post-condiciones	El conductor se almacena en el sistema.
Usuario	Sistema
1. El administrador pulsa el botón añadir conductor.	
	2. El sistema muestra el formulario añadir conductor.
3. El administrador rellena los datos del conductor.	
	4. El sistema comprueba si existe o hay algún error.
	5. El sistema muestra una ventana de confirmación.
6. El administrador confirma	
	7. El sistema almacena el nuevo conductor.

<u>Caso de uso</u>	Modificar conductor
Actores	Administrador
Propósito	Modificar los datos de un conductor del sistema.
Resumen	El administrador pulsa el botón modificar conductor. Aparece un formulario con los datos rellenos del conductor para poder modificar los datos necesarios por el administrador.
Pre-condiciones	El administrador se ha identificado en el sistema.
Post-condiciones	El conductor se almacena en el sistema.
Incluye	Buscar conductor.
Usuario	Sistema
1. El administrador selecciona el conductor.	
2. El administrador pulsa el botón modificar conductor.	3. El sistema muestra el formulario modificar conductor con los datos rellenos.
4. El administrador modifica los datos del conductor.	
	5. El sistema comprueba si hay algún error.
	6. El sistema muestra una ventana de confirmación.
7. El administrador confirma.	
	8. El sistema almacena el conductor modificado.

<u>Caso de uso</u>	Eliminar conductor
Actores	Administrador
Propósito	Eliminar un conductor del sistema.
Resumen	El administrador pulsa el botón eliminar conductor. Aparece una confirmación para eliminar. Una vez eliminado, el conductor desaparecerá de la base de datos.
Pre-condiciones	El administrador se ha identificado en el sistema.
Post-condiciones	El conductor se elimina del sistema.
Incluye	Buscar conductor.
Usuario	Sistema
1. El administrador selecciona el conductor.	
2. El administrador pulsa el botón eliminar conductor.	3. El sistema comprueba si esta libre y muestra una ventana de confirmación.
4. El administrador confirma.	
	5. El sistema elimina el conductor seleccionado.

<u>Caso de uso</u>	Leer log ruta
Actores	Administrador,
Propósito	Leer el archivo log que genera la ruta.
Resumen	Cuando el dispositivo es activado, automáticamente genera un log donde se almacena la fecha de inicio de la ruta. Cuando es desactivado envía el archivo al servidor con el número de coordenadas enviadas y la fecha de fin de la ruta.
Pre-condiciones	El administrador se ha identificado en el sistema.
Post-condiciones	El log se almacena en el sistema.
Administrador	Sistema
1. El administrador selecciona la ruta.	
2. El administrador pulsa el botón leer log.	3. El sistema muestra una ventana con los datos del log.

<u>Caso de uso</u>	Leer log ruta
Actores	Usuario,
Propósito	Leer el archivo log que genera la ruta.
Resumen	Cuando el dispositivo es activado, automáticamente genera un log donde se almacena la fecha de inicio de la ruta. Cuando es desactivado envía el archivo al servidor con el número de coordenadas enviadas y la fecha de fin de la ruta.
Pre-condiciones	El usuario se ha identificado en el sistema.
Post-condiciones	El log se almacena en el sistema.
Usuario	Sistema
1. El usuario selecciona la ruta.	
2. El usuario pulsa el botón leer log.	3. El sistema muestra una ventana con los datos del log.

<u>Caso de uso</u>	Activar dispositivo
Actores	Conductor
Propósito	Activar el dispositivo para enviar las coordenadas.
Resumen	Cuando un conductor se dispone a realizar una ruta, debe activar el dispositivo para poder transmitir las coordenadas al servidor.
Pre-condiciones	El dispositivo tiene que tener activado el GPS y los datos.
Usuario	Sistema
1. El conductor inicia la aplicación móvil.	
	2. El sistema muestra la pantalla principal.
3. El conductor pulsa el botón iniciar.	
	4. El sistema crea una ruta y envía al servidor las coordenadas del módulo GPS.

<u>Caso de uso</u>	Desactivar dispositivo
Actores	Conductor
Propósito	Desactivar el dispositivo.
Resumen	Cuando un conductor termina la ruta debe desactivar el dispositivo. Al finalizar este envía un log con el número de coordenadas enviadas, la fecha de inicio y fin de la ruta.
Pre-condiciones	El dispositivo debe estar activado.
Post-condiciones	El dispositivo envía el archivo log.
Usuario	Sistema
1. El conductor finaliza la aplicación móvil.	
	3. El sistema envía un archivo log con el número de coordenadas enviadas, la fecha de inicio y fin de la ruta.

<u>Caso de uso</u>	Añadir vehículo
Actores	Administrador
Propósito	Añadir un vehículo en el sistema para localizarlo.
Resumen	El administrador pulsa el botón añadir vehículo. Aparece un formulario para rellenar los datos del vehículo. Una vez guardado, el nuevo vehículo aparecerá en la tabla de vehículos disponibles del usuario.
Pre-condiciones	El administrador se ha identificado en el sistema.
Post-condiciones	El vehículo se almacena en el sistema.
Administrador	Sistema
1. El administrador pulsa el botón añadir vehículo.	
	2. El sistema muestra el formulario añadir vehículo.
3. El administrador rellena los datos del vehículo.	
	4. El sistema comprueba si existe o hay algún error.
	5. El sistema muestra una ventana de confirmación.
6. El administrador confirma	
	7. El sistema almacena el nuevo vehículo y le asigna un dispositivo libre.

<u>Caso de uso</u>	Modificar vehículo
Actores	Administrador
Propósito	Modificar los datos de un vehículo del sistema.
Resumen	El administrador pulsa el botón modificar vehículo. Aparece un formulario con los datos rellenos del vehículo para poder modificar los datos necesarios por el administrador. Una vez guardado, el vehículo modificado aparecerá en la tabla de vehículos disponibles del administrador.
Pre-condiciones	El administrador se ha identificado en el sistema.
Post-condiciones	El vehículo se almacena en el sistema.
Incluye	Buscar vehículo.
Administrador	Sistema
1. El administrador selecciona el vehículo.	
2. El administrador pulsa el botón modificar vehículo.	3. El sistema muestra el formulario modificar vehículo con los datos rellenos.
4. El administrador modifica los datos del vehículo.	
	5. El sistema comprueba si hay algún error.
	6. El sistema muestra una ventana de confirmación.
7. El administrador confirma.	
	8. El sistema almacena el vehículo modificado.

<u>Caso de uso</u>	Eliminar vehículo
Actores	Administrador
Propósito	Eliminar un vehículo del sistema.
Resumen	El administrador pulsa el botón eliminar vehículo. Aparece una confirmación para eliminar. Una vez eliminado, el vehículo desaparecerá de la tabla de vehículos disponibles del administrador.
Pre-condiciones	El administrador se ha identificado en el sistema.
Post-condiciones	El vehículo se elimina del sistema.
Incluye	Buscar vehículo.
Administrador	Sistema
1. El administrador selecciona el vehículo.	
2. El administrador pulsa el botón eliminar vehículo.	3. El sistema muestra una ventana de confirmación.
4. El administrador confirma.	
	5. El sistema elimina el vehículo seleccionado.

<u>Caso de uso</u>	Modificar usuario
Actores	Usuario
Propósito	Modificar los datos del usuario.
Resumen	El usuario pulsa sobre su nombre. Aparece un formulario con los datos rellenos del usuario para poder modificar los datos necesarios por el usuario.
Pre-condiciones	El usuario se ha identificado en el sistema.
Post-condiciones	El usuario se almacena en el sistema.
Usuario	Sistema
1. El usuario pulsa sobre su nombre.	3. El sistema muestra el formulario modificar usuario con los datos rellenos.
4. El usuario modifica los datos necesarios.	
	5. El sistema comprueba si hay algún error.
	6. El sistema muestra una ventana de confirmación.
7. El usuario confirma.	
	8. El sistema almacena el usuario modificado.

<u>Caso de uso</u>	Eliminar ruta
Actores	Administrador
Propósito	Eliminar un ruta del sistema.
Resumen	El administrador pulsa el botón eliminar ruta. Aparece una confirmación para eliminar. Una vez eliminado, el ruta desaparecerá de la base de datos.
Pre-condiciones	El administrador se ha identificado en el sistema.
Post-condiciones	El ruta se elimina del sistema.
Incluye	Buscar ruta.
Administrador	Sistema
1. El administrador selecciona el ruta.	
2. El administrador pulsa el botón eliminar ruta.	3. El sistema muestra una ventana de confirmación.
4. El administrador confirma.	
	5. El sistema elimina el ruta seleccionada.

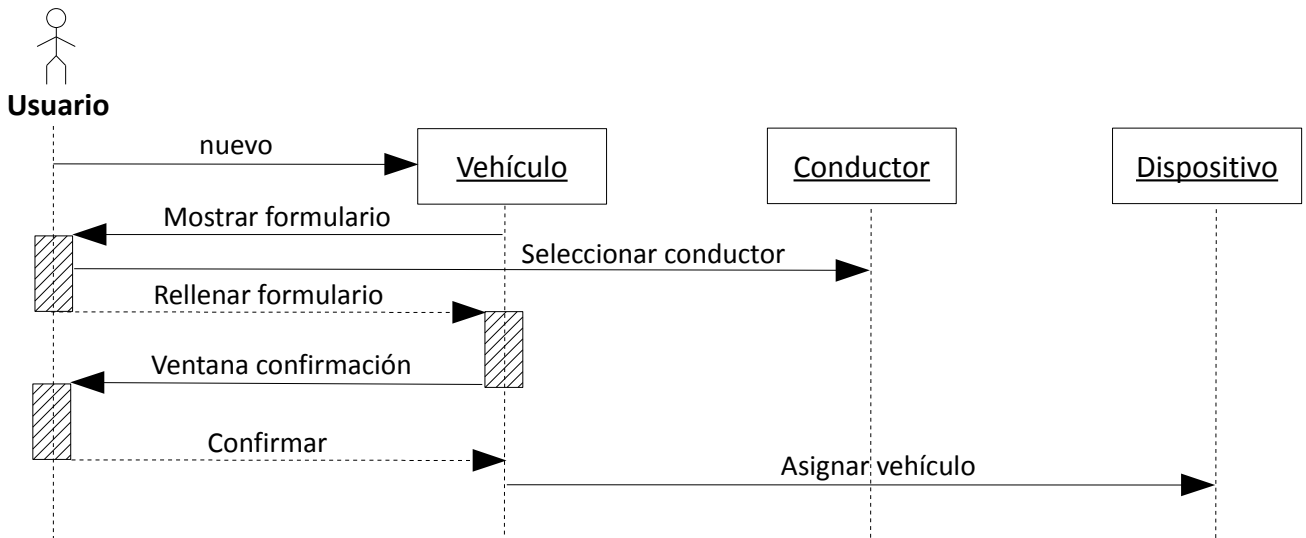
<u>Caso de uso</u>	Eliminar ruta
Actores	Usuario
Propósito	Eliminar un ruta del sistema.
Resumen	El usuario pulsa el botón eliminar ruta. Aparece una confirmación para eliminar. Una vez eliminado, el ruta desaparecerá de la base de datos.
Pre-condiciones	El usuario se ha identificado en el sistema.
Post-condiciones	El ruta se elimina del sistema.
Incluye	Buscar ruta.
Usuario	Sistema
1. El usuario selecciona el ruta.	
2. El usuario pulsa el botón eliminar ruta.	3. El sistema muestra una ventana de confirmación.
4. El usuario confirma.	
	5. El sistema elimina el ruta seleccionada.

<u>Caso de uso</u>	Alta usuario
Actores	Usuario
Propósito	Darse de alta en el sistema
Resumen	El usuario pulsa el botón registro. Aparece un formulario de alta. Una vez aceptado, el sistema envía un E-Mail al usuario.
Pre-condiciones	El usuario introduce la url en el navegador.
Post-condiciones	El usuario puede acceder a la aplicación.
Incluye	
Usuario	Sistema
1. El usuario pulsa el botón registro.	
	2. El sistema muestra un formulario de alta .
3 El usuario rellena los campos.	4. El sistema comprueba los campos, si son correctos envía un E-Mail al usuario.
	5. El sistema almacena al usuario en la base de datos.

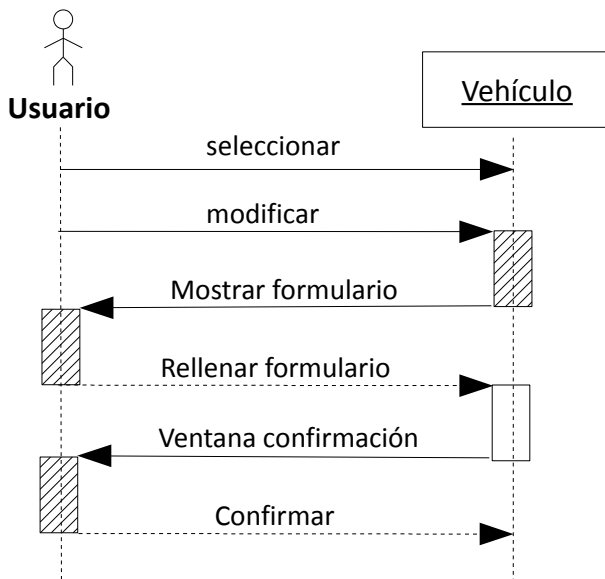
<u>Caso de uso</u>	Alta administrador
Actores	Administrador
Propósito	Darse de alta en el sistema
Resumen	El administrador pulsa el botón registro. Aparece un formulario de alta. Una vez aceptado, el sistema envía un E-Mail al administrador.
Pre-condiciones	No pueden haber usuarios en la aplicación.
Post-condiciones	El administrador puede acceder a la aplicación.
Incluye	
Administrador	Sistema
1. El administrador pulsa el botón registro.	
	2. El sistema muestra un formulario de alta .
3 El administrador rellena los campos.	4. El sistema comprueba los campos, si son correctos comprueba si es el único usuario, si es correcto le asigna derechos de administrador y envía un E-Mail al administrador.
	5. El sistema almacena al administrador en la base de datos.

10.2 Diagramas de secuencia

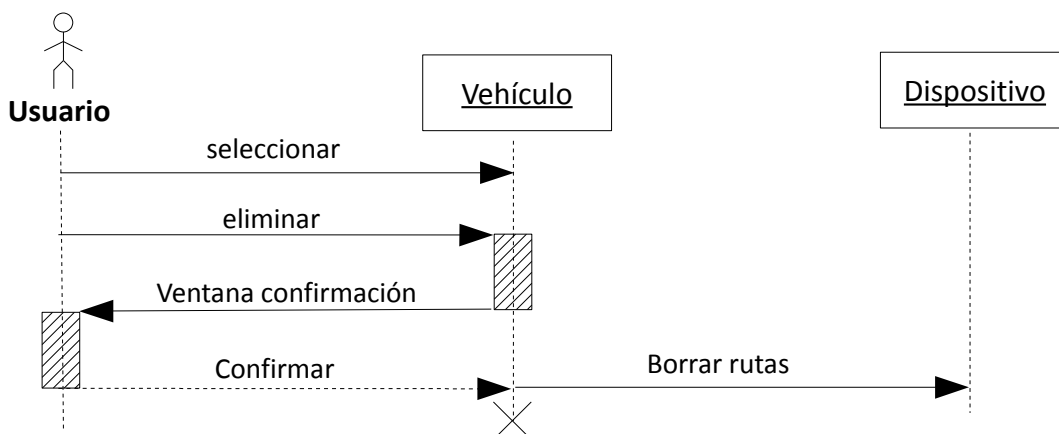
Añadir vehículo



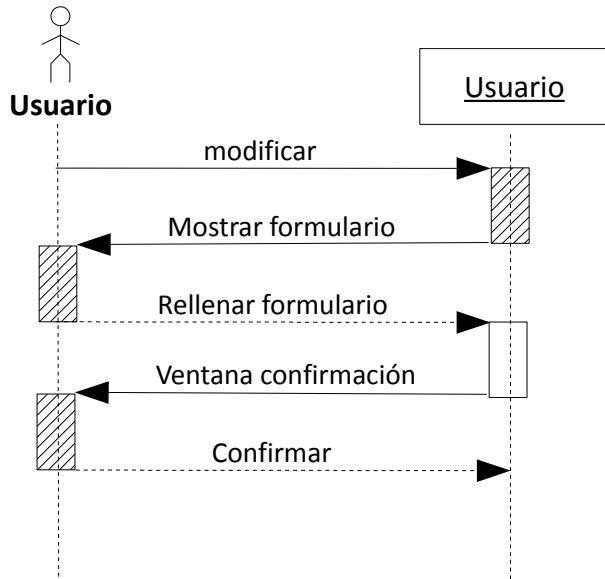
Modificar vehículo



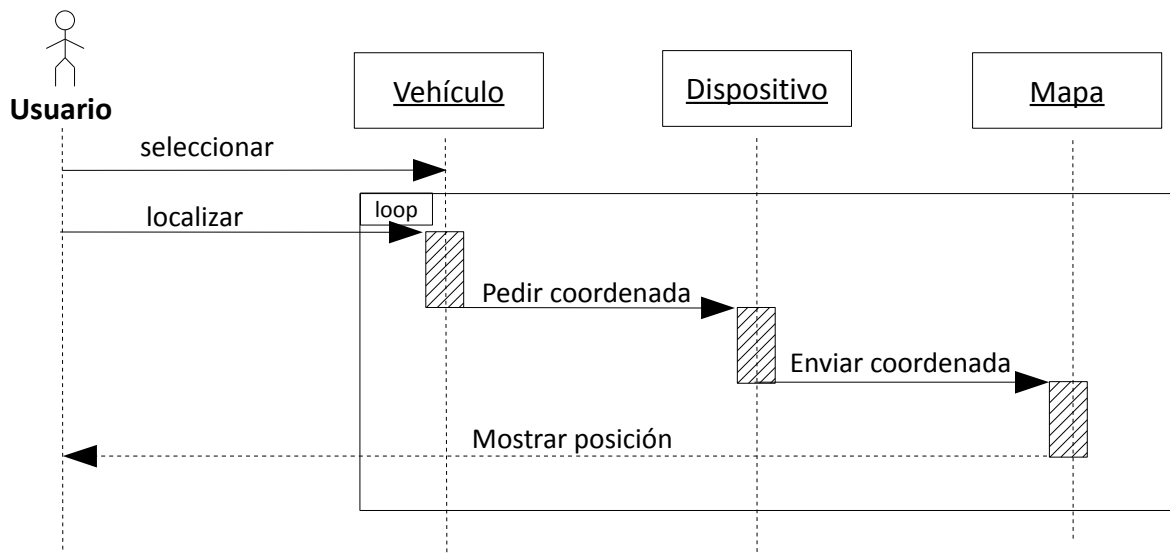
Eliminar vehículo



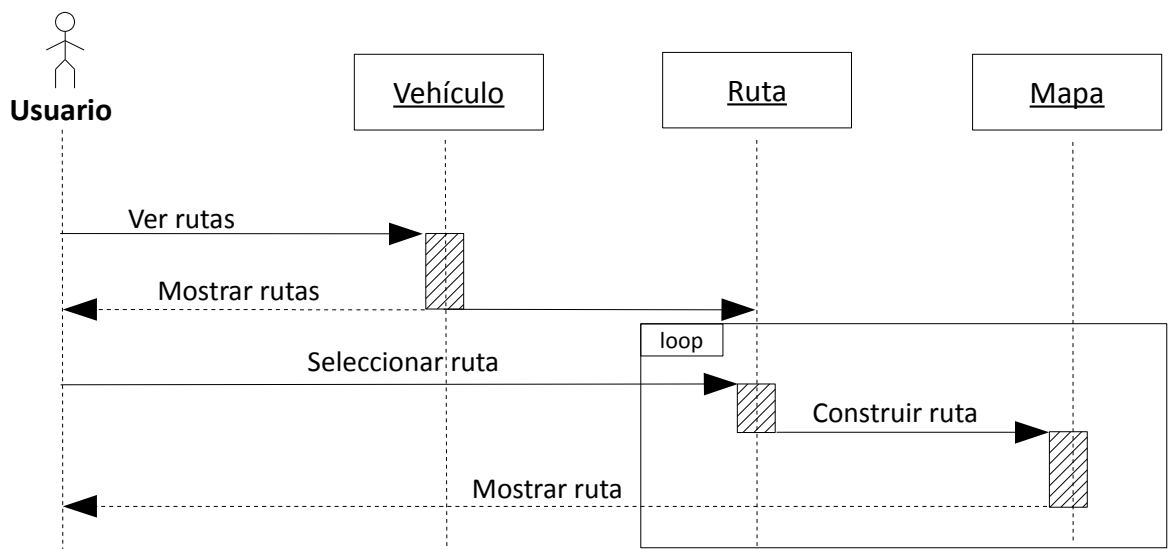
Modificar usuario



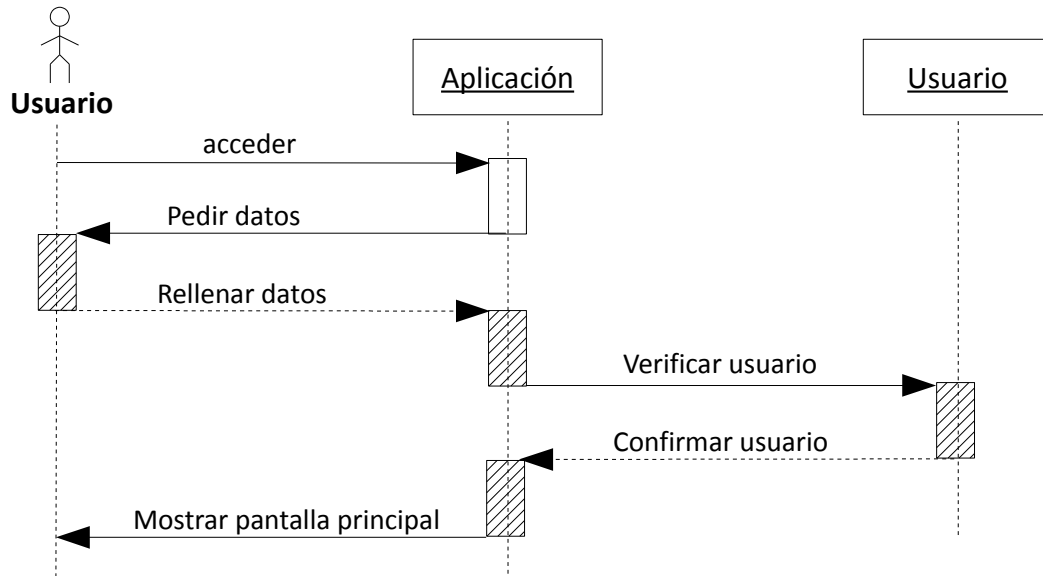
Localizar vehículo



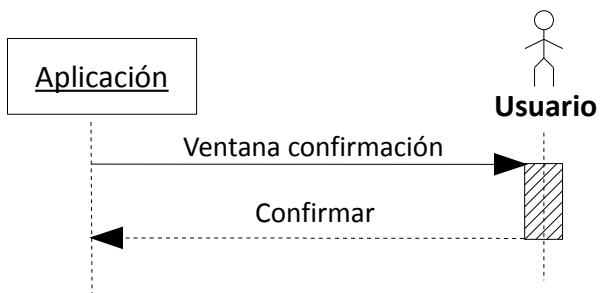
Ver ruta vehículo



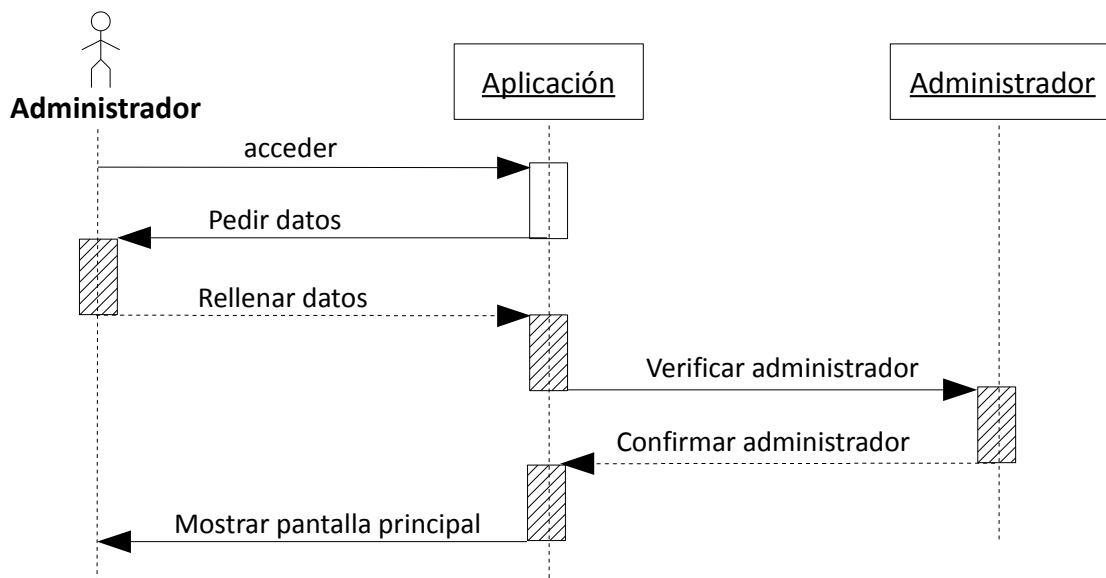
Iniciar sesión



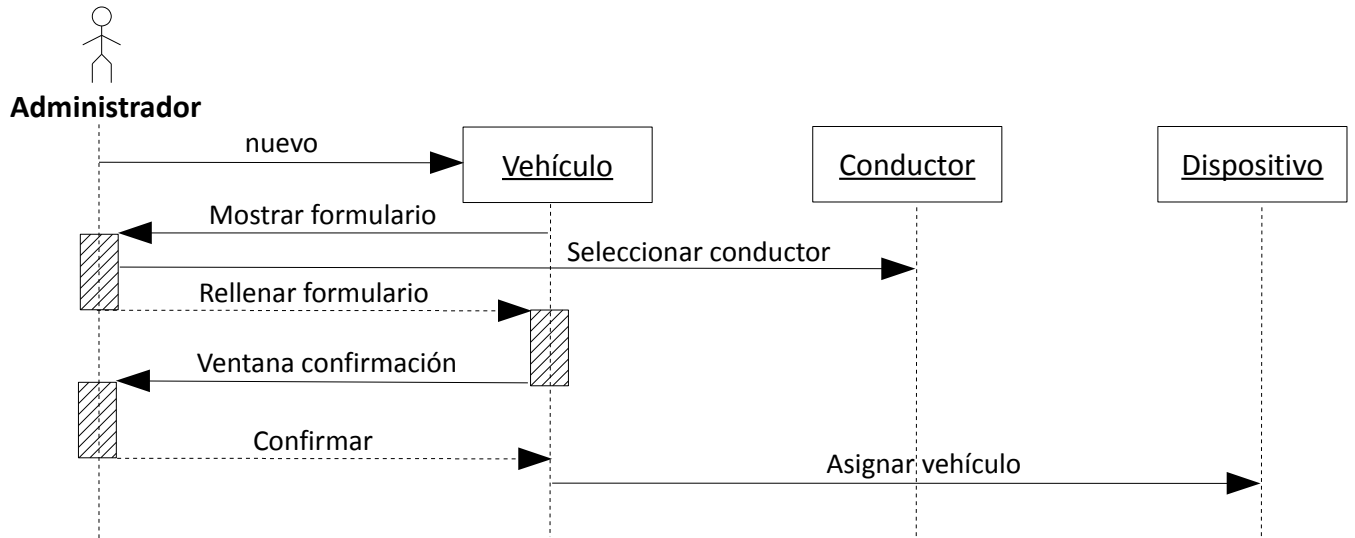
Cerrar sesión



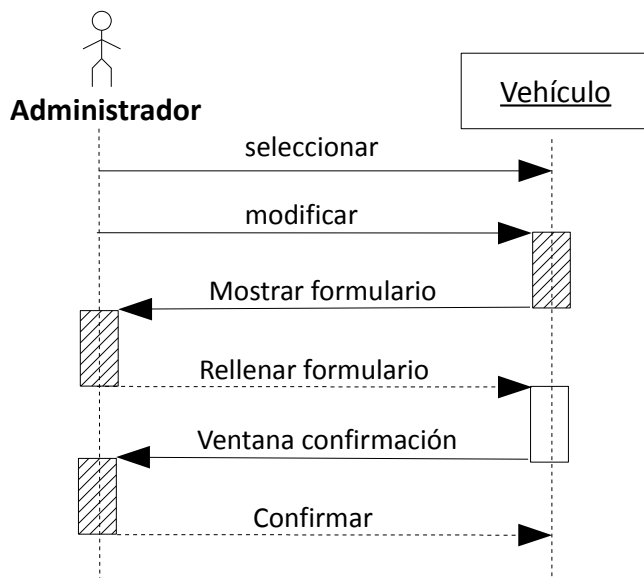
Iniciar sesión



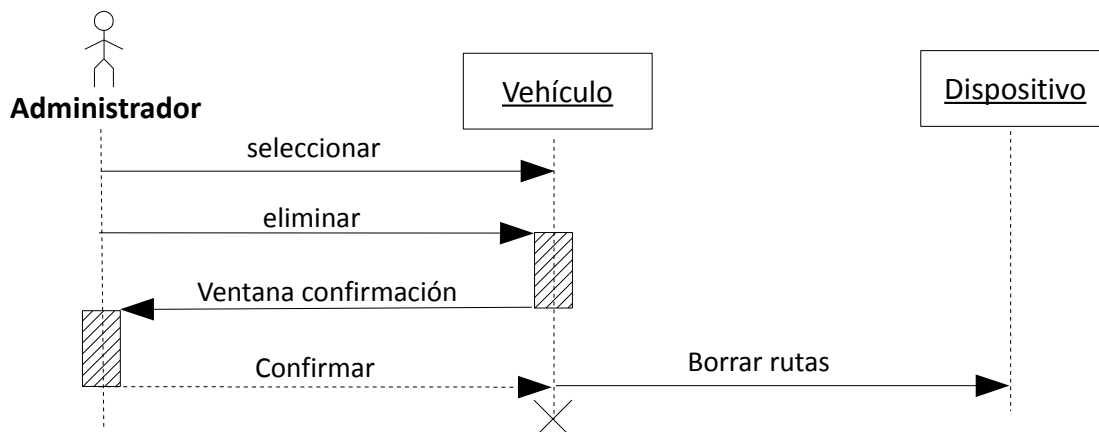
Añadir vehículo



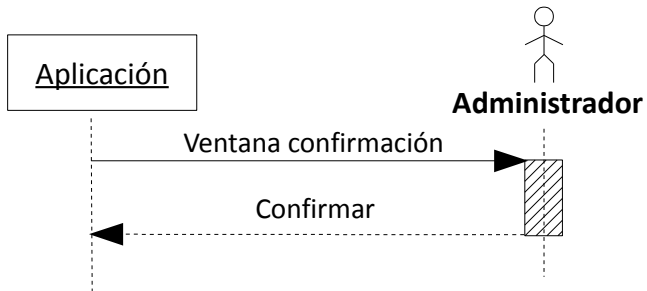
Modificar vehículo



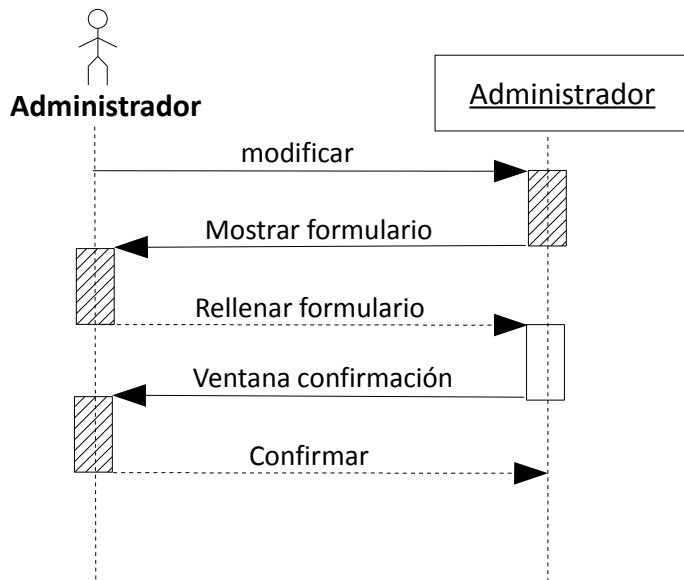
Eliminar vehículo



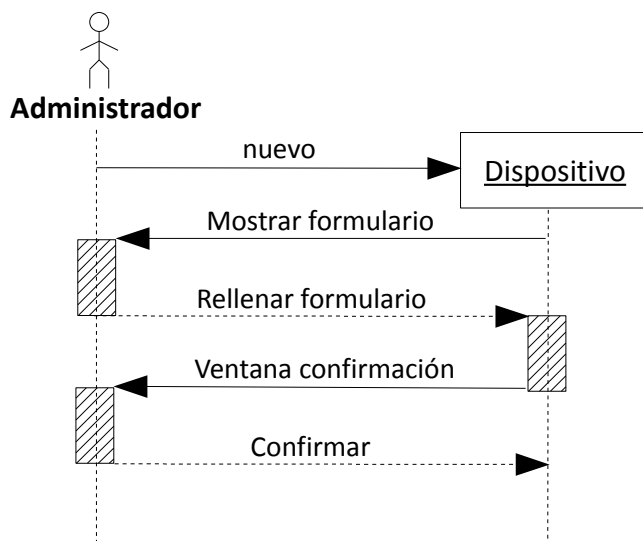
Cerrar sesión



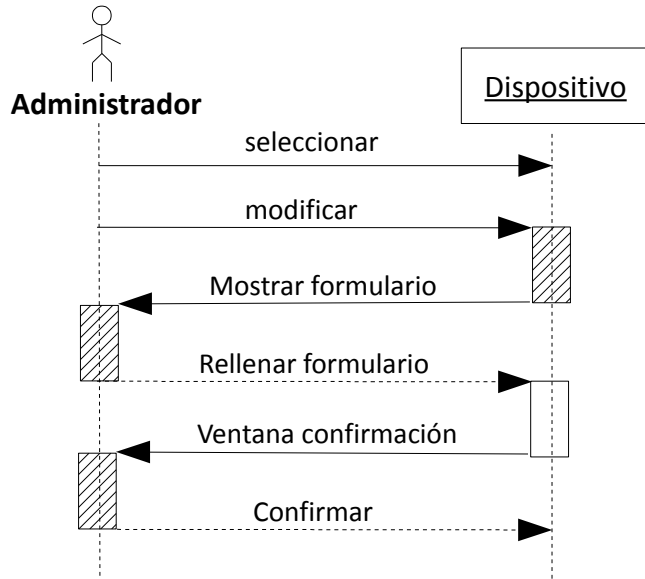
Modificar administrador



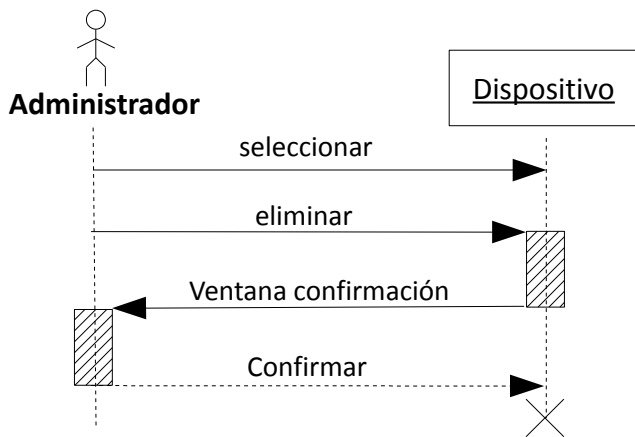
Añadir dispositivo



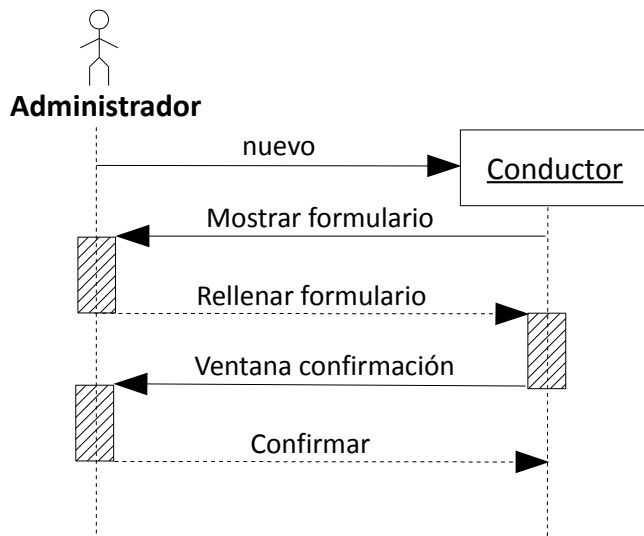
Modificar dispositivo



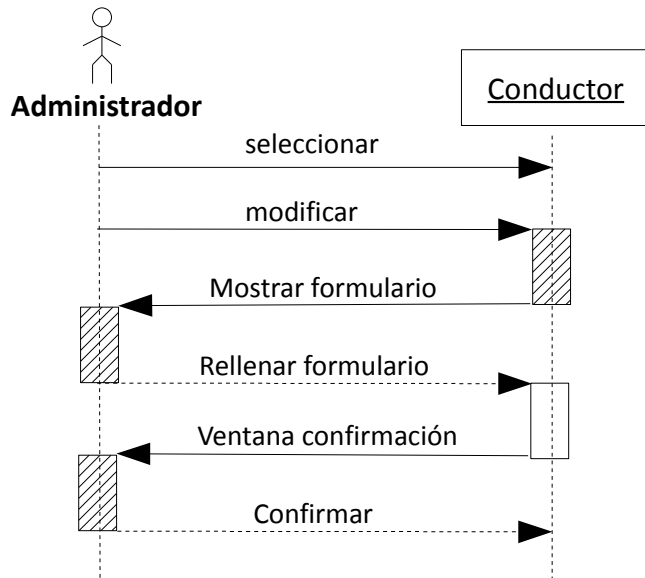
Eliminar dispositivo



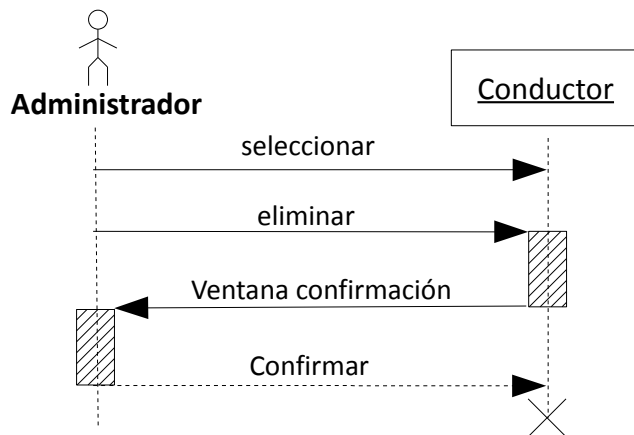
Añadir conductor



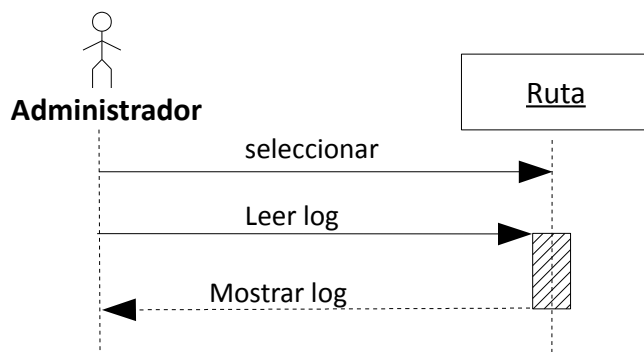
Modificar conductor



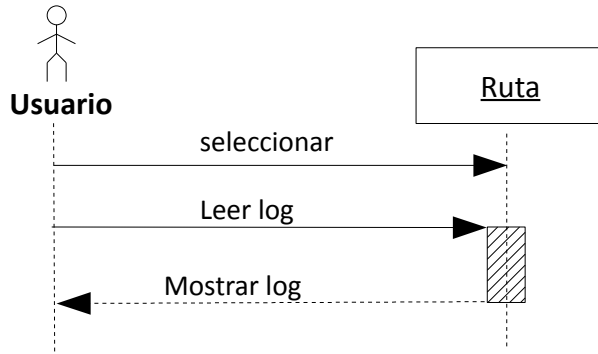
Eliminar conductor



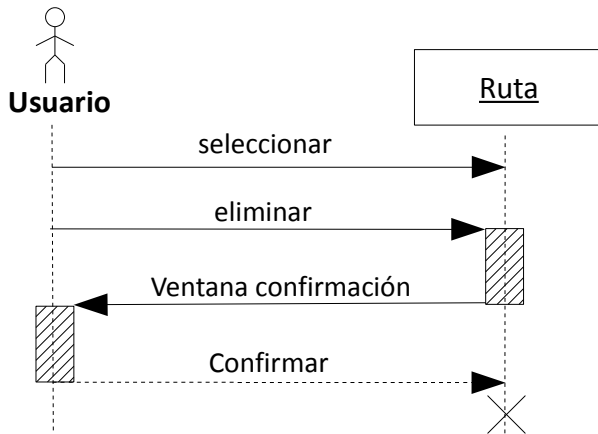
Leer log ruta



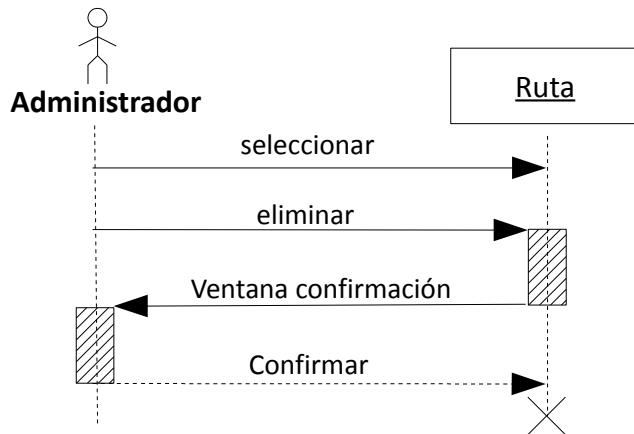
Leer log ruta



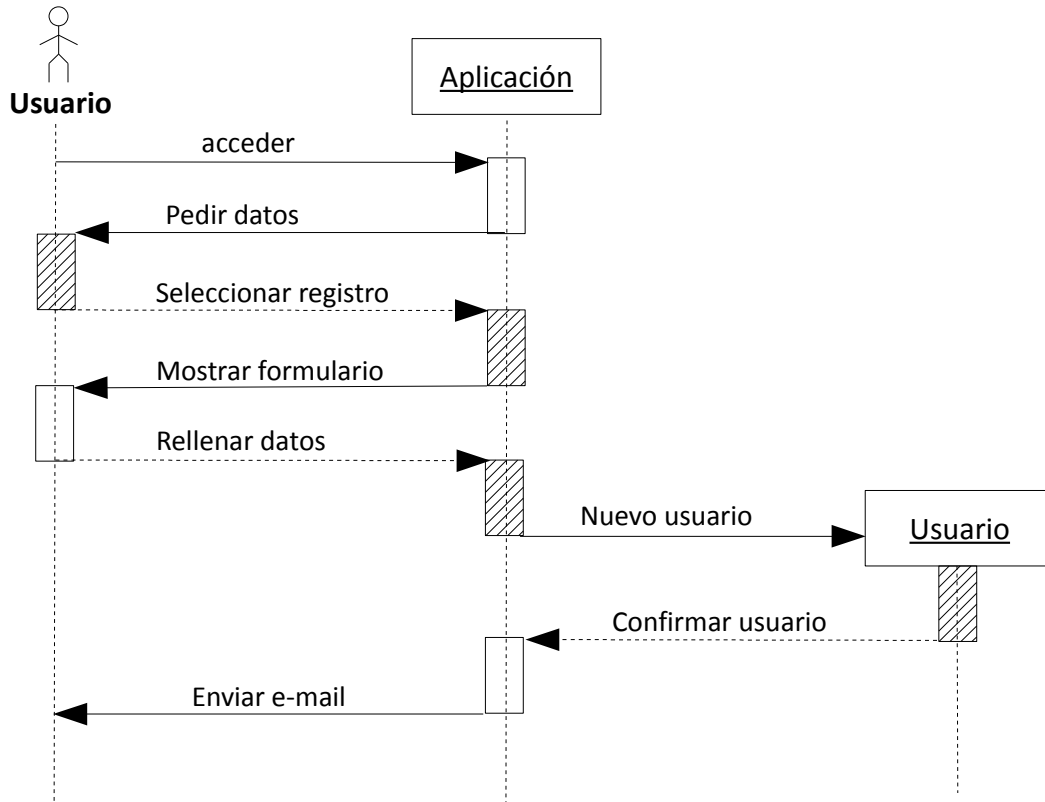
Eliminar ruta



Eliminar ruta



Alta usuario



Alta administrador

