

**Universidad de las Ciencias Informáticas**  
**Facultad 7**



**Título: Sistema para el control, posicionamiento  
y direccionamiento de vehículos.**

Trabajo de Diploma para optar por el título de  
Ingeniero en Ciencias Informáticas.

**Autores:** Alejandro Mario Velázquez Carralero.  
José Alejandro Segura Roque.

**Tutor:** Lic. Eduardo Solis Céspedes.

Ciudad de La Habana, Junio de 2007.

## DECLARACIÓN DE AUTORÍA

Nosotros: **José Alejandro Segura Roque y Alejandro Mario Velázquez Carralero**, nos declaramos como únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) a que hagan el uso del mismo de la manera en que estimen conveniente.

Y para que así conste firmamos la presente a los 21 días del mes de junio del 2007.

---

Alejandro Mario Velázquez Carralero

---

Lic. Eduardo Solis Céspedes

---

José Alejandro Segura Roque

*“Todos somos aficionados. La vida es tan corta que no da para más.”*

**Charles Chaplin**

# Agradecimientos

A mis padres, por educarme en el valor al estudio y a la superación personal, por enseñarme que los logros se forjan con una gran cuota de esfuerzo y voluntad, pero, sobre todas las cosas, por su amor.

A mi hermano, mi ejemplo y mi estímulo para seguir.

A mis profesores y amigos de la Universidad de Camagüey, en especial a Jorge Recio.

A mis amigos de la UCI, por su confianza y apoyo.

A mi tutor y profesores que revisaron el trabajo.

A mi novia, por su apoyo y comprensión.

Alejandro Mario Velázquez Carralero.

A mi mamá, por haberme apoyado en todo momento, por sus consejos, por sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada por su amor.

A mi papá, por enseñarme que la perseverancia y el esfuerzo son el camino para lograr objetivos.

A Cary, a Fernando, a Griffith y a Ana, por haberme aceptado y haberme tratado como un hijo.

A mis amigos de toda la vida, a Juli, a Roly y a Marlon, por estar conmigo en las buenas y en las malas.

A mis amigos y compañeros de la UCI, por su ejemplo y su apoyo.

A mi tutor y a todos los profesores que de una forma o de otra han influido en mi formación profesional.

José Alejandro Segura Roque.

# Dedicatoria

A mis padres, por su educación, amor y sacrificio...

A mi hermano, por su guía constante...

A mi tía Gloria, por ser una madre para mí...

A toda mi familia y amigos...

A Kariné...

Alejandro Mario Velázquez Carralero.

A mi mamá, por ser mi ejemplo, mi guía y mi luz a seguir...

A mi papá, por sus consejos y su apoyo incondicional...

A mi abuelita Ada, por su cariño y su ternura...

A Yadira, por su amor y su comprensión...

A toda mi familia y amigos....

José Alejandro Segura Roque.

# Resumen

El posicionamiento y direccionamiento a través de GPS ha cobrado vital importancia en el mundo actual. Sus aplicaciones en la gestión de emergencias y control de recursos han contribuido al mejoramiento de la calidad de vida.

En el presente trabajo se expone la implementación de un sistema de software para el posicionamiento y direccionamiento de vehículos mediante la información brindada por GPS. El mismo se basa en el empleo de estándares internacionales para la compresión, almacenamiento y transmisión de imágenes satelitales, así como para la representación de rutas, puntos de interés y recorridos en el tiempo.

Para su desarrollo se han utilizado librerías gratuitas y de código abierto compatibles con los estándares elegidos, y el marco de trabajo (*framework*) .Net de Microsoft.

Se pretende crear una herramienta que permita a los conductores conocer su posición actual y lograr direccionamientos hacia puntos de su interés. Las entidades que decidieran usarlo, tendrían un instrumento de gestión y control, que les permitiría no solo posicionar y direccionar sus vehículos, sino tener un registro de los recorridos de los mismos en el tiempo.

# Índice

AGRADECIMIENTOS.....	I
DEDICATORIA.....	II
RESUMEN.....	III
ÍNDICE.....	IV
INTRODUCCIÓN.....	1
CAPÍTULO I FUNDAMENTACIÓN DEL TEMA.....	4
1.1 SISTEMAS DE INFORMACIÓN GEOGRÁFICA.....	4
1.2 GPS.....	5
1.2.1 SEGMENTOS Y COMPONENTES DEL SISTEMA GPS.....	5
1.2.1.1 SEGMENTO ESPACIAL.....	6
1.2.1.2 SEGMENTO DE CONTROL.....	7
1.2.1.3 SEGMENTO DEL USUARIO.....	8
1.2.2 GPX.....	8
1.3 PRINCIPALES SISTEMAS DE DIRECCIONAMIENTO EN EL MUNDO.....	9
1.4 TECNOLOGÍAS UTILIZADAS PARA LA CONFECCIÓN DE NUESTRA SOLUCIÓN.....	14
1.4.1 WPF.....	14
1.4.2 WCF.....	15
1.4.3 C#.....	16
1.4.4 MICROSOFT CACHING APPLICATION BLOCK.....	16
1.4.5 NETWORK LOAD BALANCING (NLB).....	17
1.5 CONCLUSIONES.....	17
CAPÍTULO II CARACTERÍSTICAS DEL SISTEMA.....	18
2.1 PROPUESTA DE SISTEMA.....	18
2.2 MODELO DEL DOMINIO.....	23
2.3 REQUERIMIENTOS FUNCIONALES DEL SISTEMA.....	23
2.4 REQUERIMIENTOS NO FUNCIONALES DEL SISTEMA.....	24
2.5 DEFINICION DE LOS CASOS DE USO.....	26
2.5.1 DEFINICIÓN DE LOS ACTORES.....	26
2.5.2 DESCRIPCIÓN DE LOS CASOS DE USOS DEL SISTEMA.....	26
2.5.3 DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	28
2.5.4 CASOS DE USOS EXPANDIDOS.....	29
2.6 CONCLUSIONES.....	30
CAPÍTULO III ANÁLISIS Y DISEÑO DEL SISTEMA.....	31
3.1 ANÁLISIS.....	31
3.2 DISEÑO.....	32
3.2.1 DIAGRAMAS DE INTERACCIÓN.....	32

3.2.2 DIAGRAMAS DE CLASES.....	36
3.2.3 DESCRIPCIÓN DE LAS CLASES. ....	39
3.3 CONCLUSIONES.....	45
CAPÍTULO IV IMPLEMENTACIÓN .....	46
4.1 MODELO DE IMPLEMENTACIÓN.....	47
4.1.1 DIAGRAMA DE DESPLIEGUE.....	47
4.1.2 DIAGRAMA DE COMPONENTES. ....	48
4.2 CONCLUSIONES.....	49
CONCLUSIONES.....	50
RECOMENDACIONES .....	51
REFERENCIAS BIBLIOGRÁFICAS .....	52
BIBLIOGRAFÍA.....	53
GLOSARIO DE TÉRMINOS .....	535

# Introducción

En las entidades estatales de nuestro país existen varios problemas relacionados con la imposibilidad de la ubicación exacta de vehículos, así como del registro en el tiempo de su recorrido. En ocasiones, los conductores de ambulancias no saben dónde se encuentra una dirección determinada a la cual deben llegar con urgencia, además la central de ambulancias no es capaz de determinar con exactitud cuál es la que se encuentra más cerca del lugar donde se necesita. Este tipo de situaciones pueden provocar que al paciente se le deteriore aún más su estado de salud y en algunos casos llegue a perder la vida.

En una situación similar, se encuentran las patrullas de la PNR con las cuales se podría lograr una mejor gestión de emergencias si se conociera el lugar, el camino y la distancia exacta que los separan de un lugar indicado. En otro ámbito tenemos al turismo, en el cual sería igualmente aplicable un sistema de este tipo. De forma general, en ninguna de nuestras entidades estatales se puede conocer el recorrido realizado por los conductores de sus vehículos en un período de tiempo, posibilitando la utilización de los recursos del estado en beneficio personal.

Luego de analizar la situación descrita anteriormente, se formula el siguiente **problema** ¿cómo, utilizando la información brindada por los GPS, se pudiera lograr el control, direccionamiento y posicionamiento de vehículos en las entidades estatales de nuestro país?

Para dar solución al **problema** se enmarca el **objeto de estudio** en los procesos de gestión de los vehículos en las entidades estatales de nuestro país.

El estudio delimita como **campo de acción** los procesos de control, direccionamiento y posicionamiento de vehículos en las entidades estatales de nuestro país.

Para guiar la investigación se plantea la siguiente **idea a defender**: si se desarrolla un Sistema de Control, Posicionamiento y Direccionamiento de Vehículos, utilizando la información brindada por los GPS (Sistema de Posicionamiento Global), mejorará la gestión de los vehículos por parte de las entidades estatales de nuestro país, así como el servicio que brindan las mismas a la población.

La investigación se orientó a través de las siguientes **tareas**:

1. Investigar sobre el estado del arte de los Sistemas de Direccionamiento existentes en el mundo.
2. Proponer una arquitectura acorde con la situación tecnológica actual de las entidades estatales de nuestro país.
3. Estudiar los formatos de compresión de imágenes.
4. Aplicar el formato más adecuado para nuestra solución.
5. Estudiar los estándares para la representación de rutas y puntos de interés.
6. Aplicar el formato más adecuado para nuestro país.
7. Desarrollar los métodos y algoritmos de direccionamiento.
8. Investigar sobre el Sistema de Posicionamiento Global (GPS) y dispositivos afines.
9. Estudiar las nuevas tecnologías que se usarán en la implementación del software.
10. Diseñar el sistema.
11. Implementar el sistema.

Para realizar las tareas se emplearon los siguientes **métodos**:

**Métodos teóricos:**

Analítico - sintético:

Nos ha permitido analizar las teorías y los documentos referentes al objetivo de nuestra investigación, permitiéndonos de esta forma la extracción de los elementos más importantes relacionados con nuestro objeto de estudio. Además de permitirnos construir nuestra teoría y nuestro camino a seguir, a partir del análisis detallado de cada uno de los documentos previamente mencionados.

Análisis histórico-lógico:

Este método nos ha ayudado a comprender como han evolucionado en el mundo los Sistemas de Direccionamiento, permitiéndonos conocer la esencia del objeto de estudio, investigando las leyes generales y primordiales de su funcionamiento.

Modelación:

Muy útil nos ha resultado este método pues para la realización de nuestro sistema hemos utilizado RUP como metodología de desarrollo de software, con el cual se hace muy necesaria la creación

de varios modelos, ya que estos nos permiten una reproducción ampliada de la realidad, además de permitirnos descubrir y estudiar nuevas relaciones y cualidades de nuestro objeto de estudio.

### **Métodos Empíricos:**

Observación:

Éste es el único método empírico que hemos usado para la realización de nuestro sistema debido a que todavía no contamos con un cliente. Nos ha resultado muy útil, pues casi todo lo que hemos podido crear, lo hemos hecho a través del registro visual de lo que ocurre en la situación real que nos ocupa.

El presente documento consta de cuatro capítulos:

El **Capítulo 1** describe algunas de las características de los Sistemas de Direccionamiento y de los proyectos existentes semejantes al nuestro en el resto del mundo. Además hace referencia al estado del arte de las técnicas, tecnologías, metodologías y software usados para la realización de nuestro sistema.

El **Capítulo 2** da a conocer las principales características de nuestro sistema, haciendo comparaciones con otros softwares de este tipo existentes en el mundo. Además les ofrecemos el modelo de dominio de nuestra aplicación conjuntamente con la especificación de los requisitos tanto funcionales como no funcionales, y el modelo de casos de uso del sistema

El **Capítulo 3** muestra los resultados obtenidos en el desarrollo de los procesos de análisis y diseño del sistema, así como los diagramas que fueron necesarios para obtener una mayor claridad a la hora de elaborar la solución que se propone.

Y finalmente, en el **Capítulo 4** le mostramos el modelo de implementación como resultado del análisis y el diseño estando compuesto este por su respectivo diagrama de despliegue, y por su diagrama de componentes.

# Capítulo I

## FUNDAMENTACIÓN DEL TEMA.

En este capítulo se expondrán de manera general los aspectos relacionados con los sistemas de direccionamiento e información geográfica, se presentan sus conceptos y características. Se muestra además, el estado en que se encuentran en el mundo las técnicas y tecnologías que pueden ser adecuadas para la construcción del sistema a desarrollar.

### 1.1 SISTEMAS DE INFORMACIÓN GEOGRÁFICA.

El término GIS proviene del acrónimo en Inglés de Sistema de Información Geográfica (*Geographic Information System*).

Técnicamente se puede definir un GIS como una tecnología de manejo de información geográfica formada por equipos electrónicos (*hardware*) programados adecuadamente (*software*) que permiten manejar una serie de datos espaciales (información geográfica) y realizar análisis complejos con éstos, siguiendo los criterios impuestos por el equipo científico (personal). [1]

Son por tanto cuatro los elementos constitutivos de un sistema de estas características:

1. Hardware.
2. Software.
3. Datos geográficos.
4. Equipo humano

Aunque todos ellos han de cumplir con su cometido para que el sistema sea funcional, existen diferencias en cuanto a su importancia relativa. A lo largo del tiempo, el peso de cada uno de los elementos dentro de un proyecto GIS ha ido cambiando mostrando una clara tendencia: mientras los equipos informáticos condicionan cada vez menos los proyectos GIS por el

abaratamiento de la tecnología, los datos geográficos se hacen cada vez más necesarios y son los que actualmente consumen la mayor parte de las inversiones en términos económicos y de tiempo.

Hoy día, el condicionante principal a la hora de afrontar cualquier proyecto basado en GIS lo constituye la disponibilidad de datos geográficos del territorio a estudiar, mientras que hace diez años lo era la disponibilidad de ordenadores potentes que permitieran afrontar los procesos de cálculo involucrados en el análisis de datos territoriales.

Pero además de ser un factor limitante, la información geográfica es a su vez el elemento diferenciador de un Sistema de Información Geográfica frente a otro tipo de Sistemas de Información. La particular naturaleza de este tipo de información contiene dos vertientes diferentes: por un lado está la vertiente espacial y por otro la vertiente temática de los datos.

## 1.2 GPS.

Una de las maneras más utilizadas para capturar información geográfica e incorporarla a un GIS lo constituye el sistema GPS. El término GPS procede del acrónimo de la expresión inglesa 'Global Positioning System' (Sistema de Posicionamiento Global). Se trata de un sistema que permite calcular las coordenadas de cualquier punto de la superficie terrestre a partir de la recepción de señales emitidas desde una constelación de satélites en órbita.

Básicamente, su principal funcionalidad es que permite al usuario conocer, mediante un receptor, su posición en cualquier parte del planeta. Los diferentes métodos e instrumentos existentes condicionan la precisión de las mediciones realizadas, existiendo un amplio abanico de posibilidades en cuanto a resolución.

### 1.2.1 SEGMENTOS Y COMPONENTES DEL SISTEMA GPS.

El fundamento del sistema GPS consiste en la recepción de entre cuatro y ocho señales de radio de otros tantos satélites de los cuales se conoce de forma muy exacta su posición orbital con respecto a la tierra; a la vez, se conoce muy bien el tiempo que han tardado las señales en recorrer el camino entre el satélite y el receptor. Conociendo la posición de los satélites, la velocidad de propagación de sus señales y el tiempo empleado en recorrer el camino hasta el usuario, por trilateración se puede establecer la posición en términos absolutos del receptor. [2]

Para entender el sistema GPS se hace necesario conocer los elementos que lo forman. Dentro del sistema GPS existen tres conjuntos de componentes denominados segmentos:

- Segmento Espacial.
- Segmento de control
- Segmento del usuario.

Veamos a continuación las principales características de cada uno de ellos.

### 1.2.1.1 SEGMENTO ESPACIAL

El Segmento Espacial está constituido por los satélites que soportan el sistema y las señales de radio que emiten. Estos satélites conforman la llamada constelación NAVSTAR (*Navigation Satellite Timing and Ranging*), constituida por 24 satélites operativos más cuatro de reserva, mantenidos por la fuerza aérea estadounidense. No hay que olvidar, que el origen de este sistema es militar y su financiación corre íntegramente a cargo del gobierno de los Estados Unidos.

Existe también una versión rusa del sistema de posicionamiento global. Se trata de un intento incompleto que inició el gobierno ruso (Constelación Glonass), pero que acabó abandonando por falta de financiación. Esta constelación incompleta de satélites Glonass sólo se usa ocasionalmente como complemento al sistema GPS norteamericano en algunas aplicaciones de precisión.

Por otro lado, también existe en proyecto una versión europea modernizada del GPS que actualmente se encuentra en fase de elaboración. Liderado por la Agencia Espacial Europea, el Proyecto Galileo espera tener operativo todo su sistema a partir del 2008, si no hay problemas políticos o de financiación.

En medio de este escenario, el único sistema de posicionamiento global totalmente operativo hoy en día es el norteamericano, con varias generaciones de satélites ya en órbita y funcionando, y con otra nueva generación de aparatos actualmente en fase de desarrollo que prometen mejorar considerablemente las prestaciones para antes de 2008.

Los 24 satélites y sus 4 de reserva de la constelación NAVSTAR (Fig. 1), circundan la tierra en órbitas a una altura alrededor de los 20.200 Km. de la superficie (puede ser algo más o algo menos, dependiendo del satélite) y distribuidos de tal manera que en cada punto de la superficie terrestre se tiene posibilidad de leer la señal de al menos cuatro satélites. Esto es muy importante,

porque se necesitan al menos cuatro satélites para conocer la posición del observador, y que éstos se dispongan con un ángulo de elevación sobre el horizonte superior a 15°; no obstante, casi siempre son más de cuatro los satélites 'visibles'.

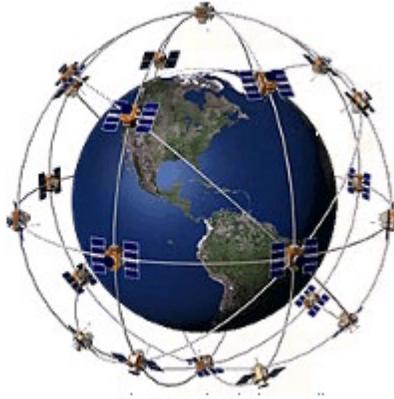


Fig. 1 Segmento espacial.

### 1.2.1.2 SEGMENTO DE CONTROL.

El segmento de control son todas las infraestructuras en tierra necesarias para el control de la constelación de satélites, mantenidas por la fuerza aérea estadounidense. Dichas infraestructuras tienen coordenadas terrestres de muy alta precisión y consisten en cinco grupos de instalaciones repartidas por todo el planeta, para tener un control homogéneo de toda la constelación de satélites (Fig. 2).

Estas infraestructuras realizan un seguimiento continuo de los satélites que pasan por su región del cielo, acumulando los datos necesarios para el cálculo preciso de sus órbitas. Dichas órbitas son muy predecibles, dado que no existe fricción atmosférica en el entorno donde se mueven los satélites; a las predicciones de las órbitas de los satélites para el futuro se les conoce con el nombre de Almanagues, cuyo cálculo depende también del segmento de control.

Sin embargo, aunque muy predecibles, las órbitas también tienen una degradación debido a una serie de factores: desigual densidad de la gravedad terrestre, mareas gravitatorias provocadas por el alineamiento de la luna y los planetas, viento solar, etc. Todos estos factores conllevan pequeñas degradaciones sobre las órbitas que hay que tener en cuenta para que el sistema GPS sea preciso. Por ello, aquellas estaciones del segmento de control que están dotadas de antenas de referencia tienen también la función de subir a los satélites las correcciones de órbita para sus sistemas de navegación.



Fig. 2 Segmento de Control.

### 1.2.1.3 SEGMENTO DEL USUARIO.

El segmento del usuario está constituido por el hardware (equipos de recepción) y el software que se utilizan para captar y procesar las señales de los satélites. Es quizá la parte que más nos interesa a nosotros como usuarios del sistema GPS, puesto que del tipo de instrumental y métodos utilizados depende la precisión alcanzada.

### 1.2.2 GPX.

GPX, formato de Intercambio GPS (*GPS eXchange Format*), es un esquema basado en XML pensado para transferir y almacenar datos GPS entre aplicaciones y servicios web (*web services*) en Internet. Se puede usar para describir puntos de interés, recorridos y rutas. GPX fue diseñado desde el inicio para ser el estándar de intercambio debido a que hereda todos los beneficios de XML de interoperabilidad y extensibilidad. Es suficientemente simple para ser aprendido en una hora pero también suficientemente poderoso para describir entidades geográficas complejas de cualquier tipo. Le permite a los desarrolladores definir su propio conjunto de elementos y atributos. Está pensado para que se desarrolle en el tiempo, basado en una comunidad donde las nuevas extensiones al lenguaje son revisadas. [3]

### 1.3 PRINCIPALES SISTEMAS DE DIRECCIONAMIENTO EN EL MUNDO.

#### Google Earth Pro:

Aunque en su versión estándar sólo provee las imágenes y puntos de interés, tiene una versión profesional que permite la localización por GPS y el direccionamiento. Permite al usuario navegar por imágenes de satélite de toda la Tierra y observar millones de datos geográficos y relacionados con servicios de ciudades (hospitales, colegios, restaurantes). También dispone de información en 3 dimensiones de algunas localidades de EEUU. La resolución de estas imágenes de satélite varía en función de la zona donde nos encontremos. Ciudades como Madrid, Barcelona, México DF, Lima o Santiago de Chile están con 70 cm. por píxel. Otras (todas estadounidenses) como Boston o Chicago, con 30.5 cm. /píxel, y otro grupo como Washington, Nueva York o Los Ángeles, con 15.2 cm. /píxel. También hay zonas, en las cuales no hay apenas poblaciones, en las que la resolución es bastante baja para de esta forma optimizar el volumen de información.

Las imágenes que muestra 'Google Earth' son obtenidas de la compañía DigitalGlobe, la cual dispone de un satélite llamado QuickBird (lanzado al espacio en octubre de 2001), y cuyas imágenes vende a cualquier cliente del mundo.[4]

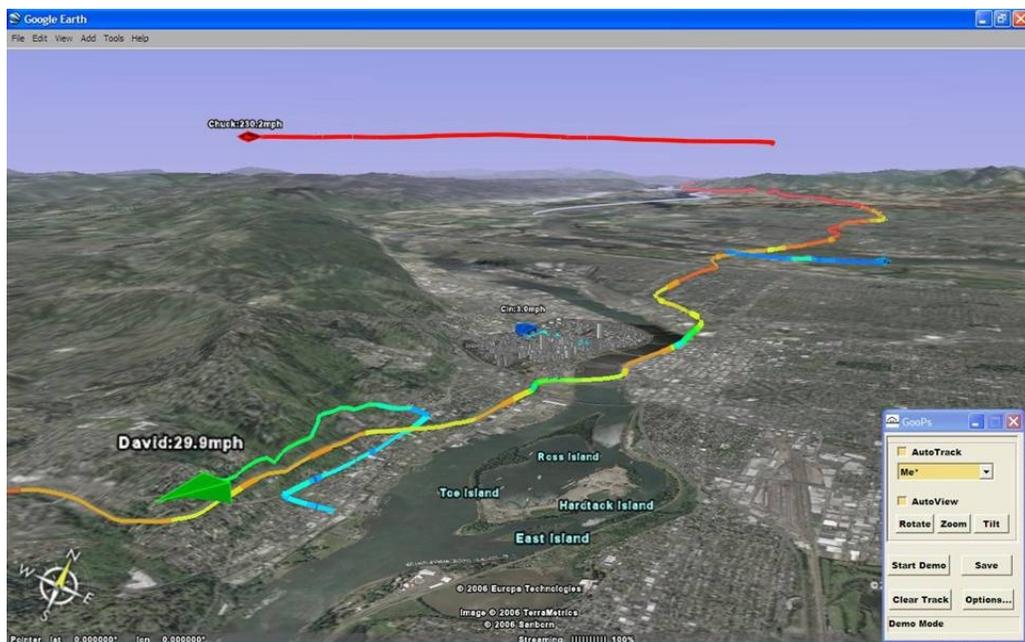


Fig. 3 Google Earth

## Sync:

Ford y Microsoft han presentado el sistema Sync, desarrollado conjuntamente por las dos compañías. Se trata de un conjunto de funcionalidades para las comunicaciones y el entretenimiento en el coche.

Lo cierto es que no presenta elementos excesivamente novedosos respecto a las tecnologías que ya conocemos con móviles y Bluetooth, pero sí la mejora de algunos aspectos. El sistema Sync incluye las siguientes funcionalidades:

- **Llamadas telefónicas** “manos libres” pulsando el botón de *teléfono* en el volante y diciendo su nombre.
- **Uso de GPS.** Permitiendo realizar un viaje virtual, y guiándonos hacia donde queramos ir.
- **Conexiones continuas:** no es necesario cortar la llamada si entramos en el coche, pulsando el botón teléfono se realiza la conexión Bluetooth.
- **Lectura de mensajes** (SMS, mail). Sync convierte el texto a voz y lee el mensaje, proporcionando además 20 respuestas predefinidas.
- **Servicios avanzados de llamada:** identificador de llamada, llamada en espera, conferencia, registro de llamadas, lista de contactos, estado de la cobertura y batería, se pueden consultar en el display del coche.
- **Música activada por voz:** podemos pedir por voz “música rock”, una pista en concreto o un artista, de los que tengamos disponibles en el reproductor conectado (iPod, Zune, etc).
- **Tonos de llamada personalizados,** incluso para cada número.
- **Copia automática de la agenda** de contactos del móvil al coche.
- **Multilinguaje:** Sync habla inglés, francés y español. [5]

## Volkswagen, Google, NVidia:

Volkswagen, Google y NVidia desde el mes de octubre de 2005 se encontraban trabajando en un nuevo sistema de navegación para autos. En enero del 2006 en las Vegas apareció el primer prototipo de este producto (Fig. 4) que promete dar un vuelco en el mundo de los sistemas de información geográfica. [6]

El elemento central es una interfaz touch-screen conectado con Google Earth. Contiene mapas tridimensionales, tráfico en tiempo real, guías de carreteras actualizadas, y facilidades de búsquedas a la hora de localizar algún punto de relativa importancia, por ejemplo Hoteles, hospitales, restaurantes, etc.

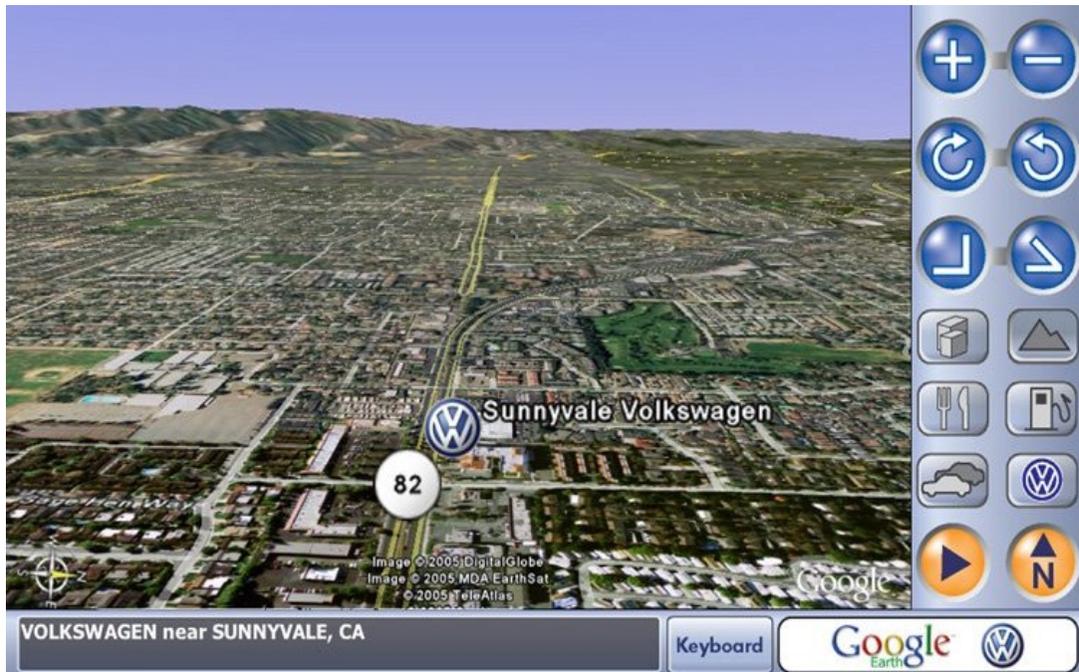


Fig. 4 VolksWagen, Google, NVidia.

### Tele Atlas:

Brinda sus soluciones de mapas digitales y tecnologías de contenido geográfico a compañías tan disímiles como Ferrari Maserati Group, el Departamento de transportación de los EEUU, el estado de Tennessee, la Reserva Federal de los EEUU, y tienen varias patentes registradas en tecnologías de mapas, algoritmos de enrutamiento y navegación.

### Route 66:

Provee soluciones de direccionamiento para Reino Unido, Irlanda y EEUU. Sus productos poseen una completa solución de navegación GPS, información de tráfico, re-enrutamientos en caso de desviaciones de la ruta original. Instrucciones de navegación en tiempo real, interfaz de botones de grandes dimensiones para el uso de tecnologías touch-screen (Fig. 5).



Fig. 5 Route 66.

### Navman:

Provee equipos PND (*Personal Navigation Device*) con mapas 3D a nivel de calle con los últimos datos de Tele Atlas (Fig. 6).



Fig. 6 Navman.

## Destinator:

En el caso de Destinator no sólo tiene versiones para PNDs (*Personal Navigation Device*) sino que también ofrece versiones para smartphones, Pocket PC y Palms.

## VIAMOTO y TeleNav

En el caso de VIAMOTO y TeleNav se dedican fundamentalmente al sector de los móviles en asociación con compañías como Motorola.



Fig. 6 Viamoto y TeleNav.

## 1.4 TECNOLOGÍAS UTILIZADAS PARA EL DESARROLLO DE NUESTRA SOLUCIÓN.

### 1.4.1 WPF (*Windows Presentation Foundation*).

Por definición, es el personal técnico quien se preocupa más por la tecnología. El interés de muchos profesionales de software se centra, principalmente, en el modo de funcionamiento de las aplicaciones y no tanto en la forma de interacción que pueda darse entre éstas y los usuarios. Sin embargo, los usuarios dan gran importancia a las interfaces. La interfaz de una aplicación constituye una parte fundamental de la experiencia global del usuario con el software particular. En lo que respecta a los usuarios, la aplicación es la experiencia. La experiencia mejorada de los usuarios mediante una interfaz optimizada puede contribuir al incremento de la productividad, a la generación de clientes leales y a una ampliación de las ventas en línea, entre muchas otras ventajas.

Los usuarios, que un día se conformaban con interfaces basadas en caracteres, ahora se han acostumbrado a las interfaces gráficas. No obstante, los requisitos que deben cumplir estas interfaces siguen aumentando. El uso de gráficos y componentes multimedia se ha hecho más generalizado. Además, la Web ha condicionado a una generación de usuarios que esperan obtener una interacción con software sencilla. La relevancia de las interfaces de las aplicaciones crece con el aumento del tiempo que los usuarios dedican a interactuar con las mismas. Para satisfacer las crecientes expectativas, la tecnología de creación de interfaces de usuario debe avanzar a la par. [7]

El objetivo de WPF es proporcionar estos avances en el entorno de Windows. WPF se incluye en la versión 3.0 de Microsoft .NET Framework y permite crear interfaces que incorporan documentos, componentes multimedia, gráficos bidimensionales y tridimensionales, animaciones, características tipo Web, etc.

## 1.4.2 WCF

WCF (*Windows Communication Foundation*) es la nueva plataforma de Microsoft para crear aplicaciones distribuidas. Podríamos definirlo también como el modelo de programación unificado para crear aplicaciones distribuidas.

Las aplicaciones distribuidas se han vuelto cada día más complejas, en el mejor de los casos saber construirlas requiere de leer cinco libros de cinco tecnologías distintas. Además la globalización informática requiere la interacción y la conectividad entre diferentes plataformas e incontables dispositivos. WCF fue concebido con el objetivo de simplificar el desarrollo de aplicaciones distribuidas.

Una de sus principales características es que cuenta con un modelo de programación unificado. Es la unificación de numerosas capacidades que antes podíamos encontrar en distintas tecnologías. Esto nos libra de tener que estar utilizando más de una tecnología para cumplir satisfactoriamente los requerimientos. De este modo, los desarrolladores harán una tarea de una única manera.

Las comunicaciones se enriquecen notablemente, ya que esta tecnología da la libertad al desarrollador de utilizar múltiples transportes, distintos tipos de formatos de mensajes y diversos patrones de mensajes.

La interoperabilidad es el plato fuerte de WCF, ya que la comunicación es totalmente independiente de la plataforma. Los servicios utilizados son sobres de SOAP, cuyo contenido es nada menos que XML. Está basado en la arquitectura de Web Services y utiliza los estándares establecidos.

El transporte es otra de sus cualidades fuertes, ya que éste se define con un protocolo neutral, de transporte neutral y formato neutral. Esto quiere decir que los servicios pueden utilizar tanto HTTP como TCP u otro mecanismo de transporte. Los desarrolladores tienen la posibilidad de agregar nuevos proveedores de transporte, y esto es indiferente al servicio. WCF separa el “código” del “transporte”, y de esta manera soporta distintos métodos de comunicación, sin representar esto una carga de trabajo extra para el desarrollador. [8].

### 1.4.3 C#.

C# es un lenguaje de programación simple pero eficaz, diseñado para escribir aplicaciones empresariales. Este toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo.

Es el único lenguaje que ha sido diseñado específicamente para ser utilizado en la plataforma .NET. Programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes soportados. Por esta razón, se suele decir que es el lenguaje nativo de .NET.

C# presenta entre otras características:

- Sencillez.
- Modernidad.
- Orientación a objetos.
- Orientación a componentes.
- Gestión automática de memoria.
- Seguridad de tipos.
- Instrucciones seguras.
- Sistema de tipos unificado.
- Extensibilidad de tipos básicos.
- Extensibilidad de operadores.
- Extensibilidad de modificadores.
- Eficiencia.
- Compatibilidad.

### 1.4.4 Microsoft Caching Application Block.

Por las características de nuestra aplicación el tema de la caché de los datos se hace de vital importancia, para esto elegimos uno de los Application Blocks que forma parte del Enterprise library 3.0 desarrollada por la comunidad de Pattern and Practices, el Caching Application Block.

Este Application Block permite a los desarrolladores incorporar una caché local en sus aplicaciones. Incluye soporte de caché en memoria y en almacenamiento persistente (base de

datos, *Isolate Storage*, etc.). Provee toda la funcionalidad necesaria para obtener, agregar y remover los datos de la caché y políticas configurables de expiración. [9]

#### 1.4.5 Network Load Balancing (NLB).

La tecnología de equilibrio de carga en la red (NLB) actúa como un clúster de cliente que distribuye el tráfico IP entrante a través de un clúster de servidores. Esta tecnología resulta ideal para habilitar la escalabilidad incremental y proporciona un alto nivel de disponibilidad para los sitios Web de comercio electrónico.

Asimismo, permite la conexión de hasta 32 servidores empresariales de Windows .NET para compartir una sola dirección IP virtual. NLB mejora la escalabilidad distribuyendo las solicitudes de cliente a través de varios servidores dentro del clúster. A medida que el tráfico aumenta, se pueden agregar servidores adicionales al clúster; cada clúster puede incluir un máximo de 32 servidores. El alto nivel de disponibilidad que aporta NLB se debe en parte a la detección automática de errores en los servidores y a la distribución del tráfico de clientes entre el resto de servidores en menos de 10 segundos, sin que ello interrumpa el servicio a los usuarios. [10]

### 1.5 CONCLUSIONES

En este capítulo se han expuesto las principales características de los Sistemas de Información Geográfica, de los Sistemas de posicionamiento y direccionamiento, así como de el Sistema de Posicionamiento Global. Hemos abordado los diferentes sistemas de posicionamiento y direccionamiento existentes en el mundo, además de dar algunas de las principales cualidades que poseen las tecnologías que han sido usadas para la construcción del sistema.

# Capítulo II

## CARACTERÍSTICAS DEL SISTEMA.

En este capítulo se darán a conocer las principales características de nuestro sistema, haciendo comparaciones con otros softwares de este tipo existentes en el mundo. Además les ofrecemos el modelo de dominio de nuestra aplicación conjuntamente con la especificación de los requisitos tanto funcionales como no funcionales, y el modelo de casos de uso del sistema.

### 2.1 PROPUESTA DE SISTEMA.

El sistema propuesto constará de 3 partes fundamentales: un cliente inteligente, la aplicación controladora y el servidor.

El cliente inteligente se encontrará ubicado en el vehículo, permitiéndole al conductor conocer su posición en tiempo real y además el direccionamiento hacia un punto geográfico determinado por él o asignado por otra persona en funciones de controlador.

La aplicación controladora, permitirá visualizar la posición en tiempo real de aquellos vehículos que se encuentran en el “dominio” del controlador, posibilitará además interactuar con cada uno de los conductores, asignarles destinos, así como visualizar sus posibles rutas y recorridos en un período de tiempo.

El servidor por su parte, almacenará y gestionará todos los datos necesarios para el buen funcionamiento del sistema, desde el repositorio de imágenes hasta los datos de cada uno de los vehículos en su dominio.

El algoritmo de enrutamiento se diseñó para que fuera configurable a las necesidades y gustos de los posibles clientes. Mientras que en su versión básica se vale de estimaciones heurísticas de los mejores recorridos basados en modelos matemáticos probados internacionalmente (Manhattan, Chebyshev, Geometría del Taxi, etc.), en una versión más avanzada, podría utilizar heurísticas basadas además en el estado de las carreteras, número de semáforos, velocidad promedio en el recorrido, información en tiempo real de embotellamientos, etc.

Estas heurísticas vienen a servir de soporte para la aplicación del algoritmo A\* (*A Star*), que es en fin de cuentas el que usamos para determinar las rutas óptimas. El problema de algunos algoritmos de búsqueda en grafos informados, es que se guían exclusivamente por la función heurística, la cual puede no indicar el camino de coste más bajo, o por el coste real de desplazarse de un nodo a otro, pudiéndose dar el caso de que sea necesario realizar un movimiento de coste mayor para alcanzar la solución. Es por ello bastante intuitivo el hecho de que un buen algoritmo de búsqueda informada debería tener en cuenta ambos factores, el valor heurístico de los nodos y el coste real del recorrido. [11]

Así, el algoritmo A\* utiliza una función de evaluación  $f(n) = g(n) + h(n)$ , donde  $h(n)$  representa el valor heurístico del nodo a evaluar, y  $g(n)$ , el coste real del camino recorrido para llegar a dicho nodo. A\* mantiene dos estructuras de datos auxiliares, que podemos denominar abiertos, implementado como una cola de prioridad (ordenada por el valor  $f(n)$  de cada nodo), y cerrados, donde se guarda la información de los nodos que ya han sido visitados. En cada paso del algoritmo, se expande el nodo que esté primero en abiertos, y en caso de que no sea un nodo objetivo, calcula la  $f(n)$  de todos sus hijos, los inserta en abiertos, y pasa el nodo evaluado a cerrados. El algoritmo es una combinación entre búsquedas del tipo “primero en anchura” con “primero en profundidad”: mientras que  $h(n)$  tiende a “primero en profundidad”,  $g(n)$  tiende a “primero en anchura”. De este modo, se cambia de camino de búsqueda cada vez que existen nodos más prometedores.

Para alimentar este grafo se le deben introducir solamente los puntos que sean significativos para el direccionamiento, a saber, los inicios y fines de las calles y las intersecciones entre ellas. Todos los demás puntos interiores, que definen a las calles geodésicamente, no son significativos, pues, si se puede ir desde un inicio hasta un fin, quiere decir que también se puede transitar hasta cualquier punto interior a éstos. Las intersecciones se determinan por puntos que están repetidos en dos o más calles, sin embargo, la posibilidad de errores humanos a la hora de introducir los datos, provoca que existan puntos repetidos sin que estos sean una intersección. Todo esto implica un trabajo de discriminación para determinar las intersecciones verdaderas. Después de hecho todo esto es que se puede proceder a introducir todos estos puntos al grafo y establecer las relaciones entre ellos.

En el caso de las imágenes, el formato de compresión y almacenamiento elegido fue ECW (*Enhanced Compressed Wavelet*), que se ha ido convirtiendo poco a poco en el estándar de facto en este tipo de aplicaciones, a pesar de ser de los últimos formatos en aparecer.

ECW comprime transformando las imágenes al espacio wavelet usando Transformaciones Wavelet Discretas (DWT) multi-nivel. El siguiente paso es la cuantización, que reduce la cantidad de información de la imagen, seguido por la fase de codificación, que es la que realmente comprime las imágenes del espacio wavelet. La imagen comprimida ECW se procesa línea a línea directamente a partir de la imagen original. Por tanto, la técnica de compresión ECW puede comprimir imágenes de cualquier tamaño usando relativamente poca memoria RAM, siendo extremadamente útil para la compresión de imágenes muy grandes. El ECW usa una técnica de algoritmo "pipeline" recursivo que no requiere el uso de almacenamiento en disco mientras se realiza las transformaciones. [12]

Las imágenes comprimidas ECW tienen un archivo de cabecera .ers estándar con información sobre la imagen, por lo que la información completa de georreferencia, proyección y datum (modelo matemático de la Tierra obtenido a partir de, y usado para, cálculos geodésicos) se mantienen. Las imágenes comprimidas encajarán automáticamente con otros datos raster o vectoriales al igual que los archivos no comprimidos. Esto le permite a nuestro sistema que se le puedan adicionar nuevas imágenes ECW de mayor definición en "caliente" y éstas encajarían automáticamente en el lugar que le corresponde, sin que nuestro sistema tenga que ser recompilado para esto.

Uno de los aspectos más relevantes que debería tener en consideración un usuario de GPS, es el hecho de que una misma posición real, contendrá valores de coordenadas diferentes si se expresan en dos datum diferentes. El GPS hace posible tener un datum mundial como el World Geodetic System de 1984 (WGS84).

Todas estas características que provee ECW permiten realizar una sustitución de imágenes por capas, donde cada imagen de una capa se sustituye por cuatro imágenes de la capa posterior, logrando que la resolución aumente el doble por cada sustitución. Este es el mismo sistema usado por muchos servidores internacionales, por lo que nuestro sistema estaría preparado para sustituir por capas provistas por estos servidores siempre y cuando usen para la georeferenciación el mismo datum estándar que la nuestra (WGS84).

Por último, una aplicación de este tipo, tiene que lidiar con estructuras de datos que en el mejor de los casos cuentan con cientos de miles de instancias, datos de éstas y sus relaciones. Además, el usuario común espera tiempos de respuesta muy cortos, sin reparar quizás en los cálculos que implican una sencilla operación de direccionamiento. Todo esto conlleva a que el tema del rendimiento se convierta en algo crucial.

Por lo tanto, toda estrategia o algoritmo tenía que pasar primero por un test de rendimiento, y el desarrollo en muchas ocasiones tuvo que buscar vías alternativas en pos de mejores tiempos de ejecución. Para lograr este objetivo nos basamos esencialmente en técnicas de indexación múltiple y hashing, de modo que, no se hiciera ni una sola búsqueda secuencial sino que, el dato que fuera, estuviera al alcance de una sola operación.

Segundo, utilizamos lo que denominamos después “cargas inteligentes” de datos por niveles, de modo que, en cada nivel solo se analizarían aquellos datos que tenían sentido en el mismo, por ejemplo, un direccionamiento a nivel de país solo tendría (y cargaría por tanto) datos de carreteras interprovinciales o interestatales según fuera el caso, y, a medida que fuera aumentando el nivel de detalle se irían cargando datos más específicos hasta llegar a cargar y representar todas las calles en un sector determinado.

Tercero, una aplicación de este tipo tiene que tener una gran prioridad para el hilo principal encargado de mantener actualizada la interfaz de usuario y por tanto, de la renderización de todas las imágenes y datos. Sin embargo, todas las operaciones que involucren búsquedas de imágenes y/o datos al servidor, tienen que realizarse en un segundo plano y por tanto, en hilos de ejecución paralelos al principal. Ahora, para que la interfaz de usuario se actualice de manera más eficiente, estos datos deben tener la mayor granularidad posible, esto implica que la cantidad de hilos de ejecución aumente excesivamente.

Por tanto, se hizo necesaria la implementación de un “administrador” de hilos de ejecución que llamamos *ThreadManager*, que permitiría definir políticas referentes a la cantidad máxima de hilos concurrentes activos. Además mantendría en una cola con prioridad los hilos en espera, para que, en caso de la llegada de nuevos hilos de ejecución con más urgencia, estos pudieran ser ubicados en correspondencia con su nivel de prioridad. Como resultado, se logró mantener la interfaz de usuario con un nivel de respuesta adecuado y al mismo tiempo tener un control absoluto de la cantidad y prioridad de hilos a ejecutar.

Cuarto, teniendo en cuenta que la transmisión de imágenes se hace desde un servicio web, que la serialización a XML de datos binarios es extremadamente ineficiente, y que la implementación básica que traen los servicios web de .Net de los estándares de SOAP 1.1 no incluye ninguna de las especificaciones WS-\*, usamos en principio los WSE 3.0 (*Web Services Enhancements*) y la implementación de los mismos de MTOM (*Message Transmission Optimization Mechanism*); lográndose que la transmisión de imágenes desde el servicio web fuera mediante serialización binaria y por lo tanto más eficiente. Pero, al migrar nuestro servidor a WCF

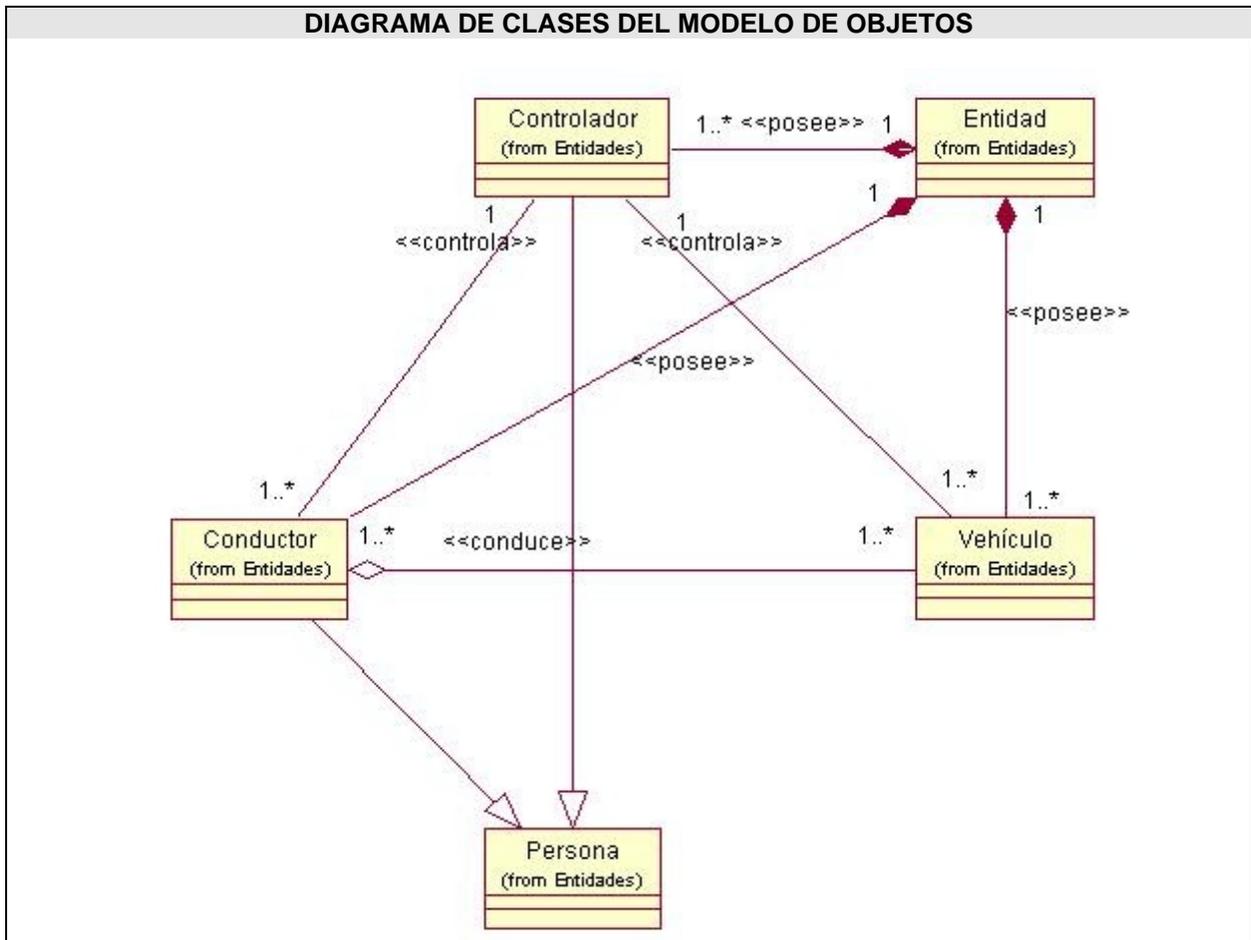
(*Windows Communication Foundation*) pudimos lograr esto de una manera más simple y declarativa.

Quinto, con el fin de reducir la transmisión innecesaria de imágenes establecimos tres niveles de caché, el primero en la memoria del dispositivo o computadora cliente donde estarían las imágenes usadas más recientemente, el segundo en el dispositivo de almacenamiento del cliente, que podría ser tarjetas de memorias Flash (por ejemplo memorias SD), discos duros, etc. Y el tercero estaría en la memoria de los servidores.

Todas estas características permitirán que nuestro sistema pueda llegar a brindar prestaciones semejantes a la de los sistemas de este tipo a nivel mundial, además de proporcionar el recorrido realizado por un vehículo en un período de tiempo determinado, la cual constituye una funcionalidad no muy común en este tipo de software, todo esto sin mencionar que una solución tan a la medida como esta le costaría varios millones de dólares al país.

Por lo tanto, nuestro objetivo fue construir una solución siguiendo estándares internacionales, que pueda ser comercializable en cualquier parte del mundo, aplicable en cualquier institución estatal de nuestro país, en ambulancias, patrullas policiales, bomberos, tractores, etc.

## 2.2 MODELO DEL DOMINIO.



## 2.3 REQUERIMIENTOS FUNCIONALES DEL SISTEMA

Requerimientos funcionales de la aplicación cliente:

R1 Visualizar la posición actual del vehículo.

R2 Visualizar la posición de los puntos de interés en el mapa.

- Destacar con estereotipos los diferentes puntos de interés.

R3 Direcccionar al conductor hacia un sitio determinado.

## Requerimientos funcionales de la aplicación controladora:

R4 Visualizar los vehículos pertenecientes al dominio.

R5 Visualizar la posición de los puntos de interés en el mapa.

- Destacar con estereotipos los diferentes puntos de interés.

R6 Conocer el recorrido de un vehículo en un período de tiempo determinado

R7 Enviar destino a un vehículo determinado.

## 2.4 REQUERIMIENTOS NO FUNCIONALES DEL SISTEMA

Apariencia o interfaz externa:

- Diseño amigable y que sea compatible con la posible aplicación de tecnologías touch-screen.

Usabilidad:

- El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora.

Rendimiento:

- Tiempos de respuestas rápidos al igual que la velocidad de procesamiento de la información

Soporte:

- Utilización de más de un servidor de aplicaciones que permitan mediante el uso de tecnologías de NLB (Network Load Balancing) soportar la carga de una aplicación de este tipo.

Seguridad:

- Garantizar el acceso a la aplicación controladora de personal autorizado.
- Garantizar la seguridad sobre los datos del recorrido de los vehículos.

- Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos

Confiabilidad:

- La utilización de las tecnologías de clustering y NLB en este tipo de aplicaciones garantizará la disponibilidad y confiabilidad de sistema.

Funcionalidad:

- Guardar en caché las imágenes y datos usados frecuentemente.

Software utilizados:

- .Net Framework 3.5.
- Caching Application Block
- SharpMap Library
- ECW ActiveX Controls.
- Application Center 2000.

## 2.5 DEFINICION DE LOS CASOS DE USO.

### 2.5.1 DEFINICIÓN DE LOS ACTORES

<b>Actores</b>	<b>Justificación</b>
Conductor	Es el usuario principal del sistema, el cual interactúa con el mismo desde el vehículo.
Controlador	Se llama controlador a aquel trabajador encargado de coordinar el trabajo de cada uno de los vehículos de su dominio.
Usuario	Es una generalización de los actores conductor y controlador.

### 2.5.2 DESCRIPCIÓN DE LOS CASOS DE USOS DEL SISTEMA.

CU-1	Conocer Posición actual del vehículo.
<b>Actor</b>	Conductor
<b>Descripción</b>	El conductor visualiza en todo momento su posición geográfica.
<b>Referencia</b>	R1

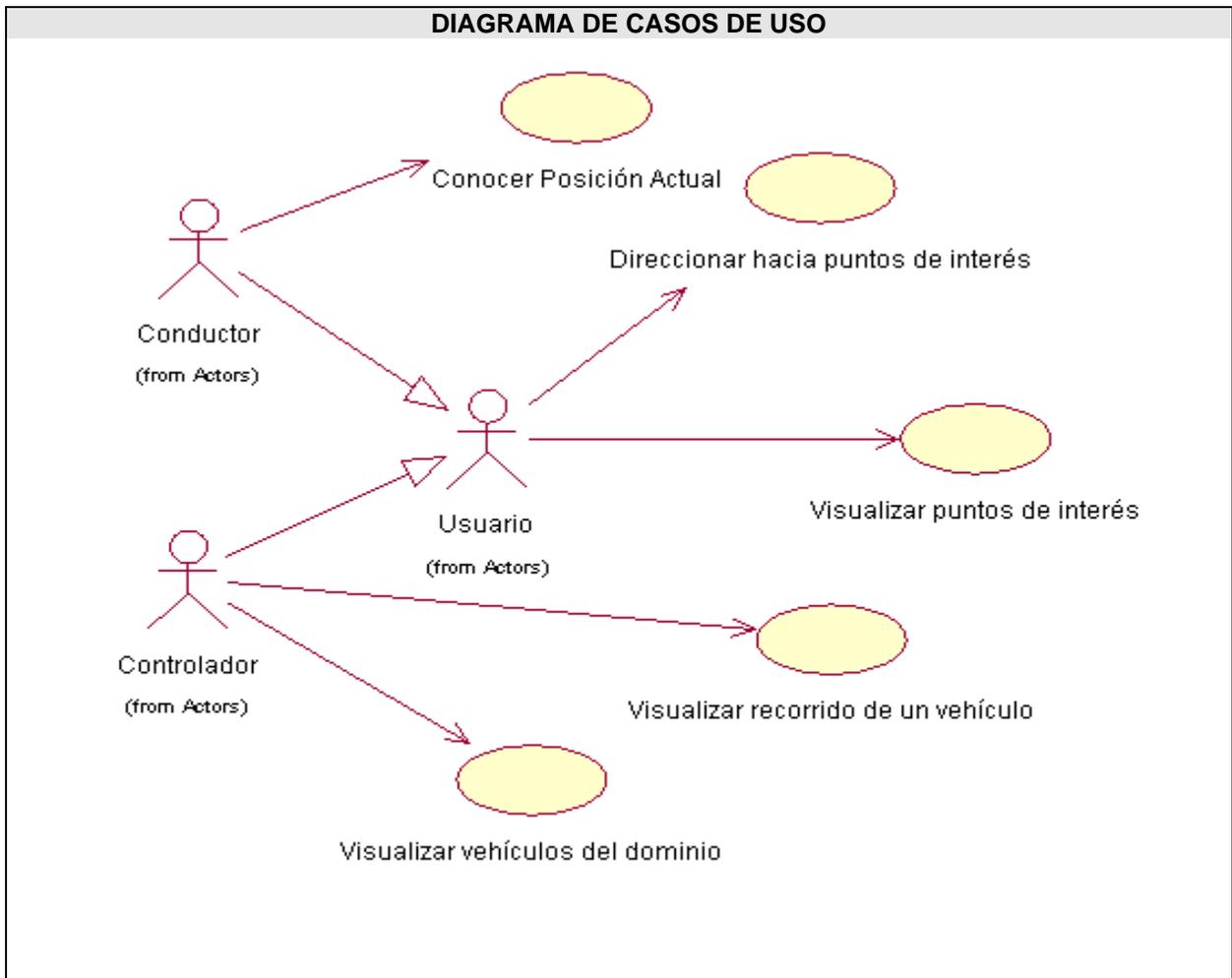
CU-2	Visualizar puntos de interés
<b>Actor</b>	Usuario
<b>Descripción</b>	El usuario puede solicitar la visualización de los puntos de interés, en un área y de un tipo determinado.
<b>Referencia</b>	R2,R5

CU-3	Direccionar hacia un punto de interés
<b>Actor</b>	Usuario
<b>Descripción</b>	El usuario puede direccionar desde un punto inicial hacia un punto final.
<b>Referencia</b>	R3,R7

CU-4	Visualizar vehículos del dominio
<b>Actor</b>	Controlador
<b>Descripción</b>	El controlador puede conocer la posición geográfica de cada uno de los vehículos pertenecientes en su dominio.
<b>Referencia</b>	R4

CU-5	Visualizar recorrido de un vehículo
<b>Actor</b>	Controlador
<b>Descripción</b>	El controlador visualiza el recorrido de un vehículo en un periodo de tiempo determinado
<b>Referencia</b>	R6

### 2.5.3 DIAGRAMA DE CASOS DE USO DEL SISTEMA.



## 2.5.4 CASOS DE USOS EXPANDIDOS

Caso de uso	
<b>CU-1</b>	Conocer Posición actual del vehículo.
<b>Propósito</b>	Mostrar gráficamente la posición geográfica del vehículo.
<b>Actores:</b> Conductor	
<b>Resumen:</b> El caso de uso se inicia cuando el conductor inicializa la aplicación y esta automáticamente le muestra su posición actual en el mapa. La aplicación lo hará en cualquier momento en que el conductor lo estime necesario y este se encuentre visualizando cualquier otra región del mapa que no lo incluya.	
<b>Referencias</b>	R1
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
Inicializar la aplicación	Muestra en la pantalla y en el mapa la posición actual del vehículo
<b>Flujo alternativo</b>	
Flujo alternativo en caso de que el actor ya halla inicializado la aplicación.	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
El conductor pide al sistema mostrar su posición actual	Muestra en la pantalla y en el mapa la posición actual del vehículo

Caso de uso	
<b>CU-2</b>	Visualizar puntos de interés.
<b>Propósito</b>	Mostrar gráficamente la posición geográfica de los diferentes puntos de interés.
<b>Actores:</b> Usuario	
<b>Resumen:</b> El caso de uso se inicia cuando el usuario le pide al sistema que se muestre en pantalla los puntos de interés de un tipo determinado por él que se hallan en la región que actualmente muestra el sistema.	
<b>Referencias</b>	R2
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
Selecciona un tipo de punto de interés a mostrar	Muestra en la pantalla los puntos de interés que se encuentren en la región que actualmente se muestra y que sean del tipo solicitado.

Caso de uso	
<b>CU-3</b>	Direccionar hacia un punto de interés
<b>Propósito</b>	Mostrar gráficamente las posibles rutas a seguir para llegar a un punto determinado por el controlador.
<b>Actores:</b> Usuario	
<b>Resumen:</b> El caso de uso se inicia cuando el usuario define un punto inicial y final al cual se quiere dirigir.	
<b>Referencias</b>	R3, R7
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
El usuario selecciona punto inicial y punto final, y elige direccionar	El sistema le muestra la ruta a seguir.

Caso de uso	
<b>CU-4</b>	Visualizar vehículos del dominio
<b>Propósito</b>	Mostrar gráficamente la posición geográfica de los diferentes vehículos pertenecientes al dominio del controlador.
<b>Actores:</b> Controlador	
<b>Resumen:</b> El caso de uso se inicia cuando el controlador solicita que se muestren en pantalla la posición de los vehículos que se encuentran en su dominio, permitiéndole restringir la visualización por un criterio determinado.	
<b>Referencias</b>	R4
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
Solicita que se muestren en pantalla los vehículos que pertenecen a su dominio	Muestra en el mapa la posición geográfica de los vehículos pertenecientes al dominio de dicho controlador.
<b>Flujo alternativo</b>	
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
El controlador pide a la aplicación que muestre en pantalla la posición geográfica de vehículos que cumplen con un criterio determinado.	Muestra en la pantalla la posición geográfica de los vehículos que cumplen con dicho criterio de búsqueda.

Caso de uso	
<b>CU-5</b>	Visualizar recorrido de un vehículo
<b>Propósito</b>	Mostrar gráficamente el recorrido realizado por un vehículo en un período de tiempo determinado.
<b>Actores:</b> Controlador	
<b>Resumen:</b> El caso de uso se inicia cuando el controlador solicita que se represente en el mapa el recorrido realizado por un vehículo en un período de tiempo determinado.	
<b>Referencias</b>	R6
<b>Acción del actor</b>	<b>Respuesta del sistema</b>
Solicita que se muestre el recorrido de un vehículo en un período de tiempo.	Muestra en el mapa el recorrido de dicho vehículo en el periodo de tiempo especificado por el actor.

## 2.6 CONCLUSIONES.

A lo largo de este capítulo, se ha querido dar una panorámica sobre las principales características con que cuenta nuestro sistema, dando a conocer los motivos que nos condujeron al desarrollo de nuestra aplicación. Se muestra el modelo de dominio, conjuntamente con la especificación de los requisitos tanto funcionales como no funcionales, y el modelo de casos de uso del sistema, en el cual nos guiamos para el desarrollo de la misma.

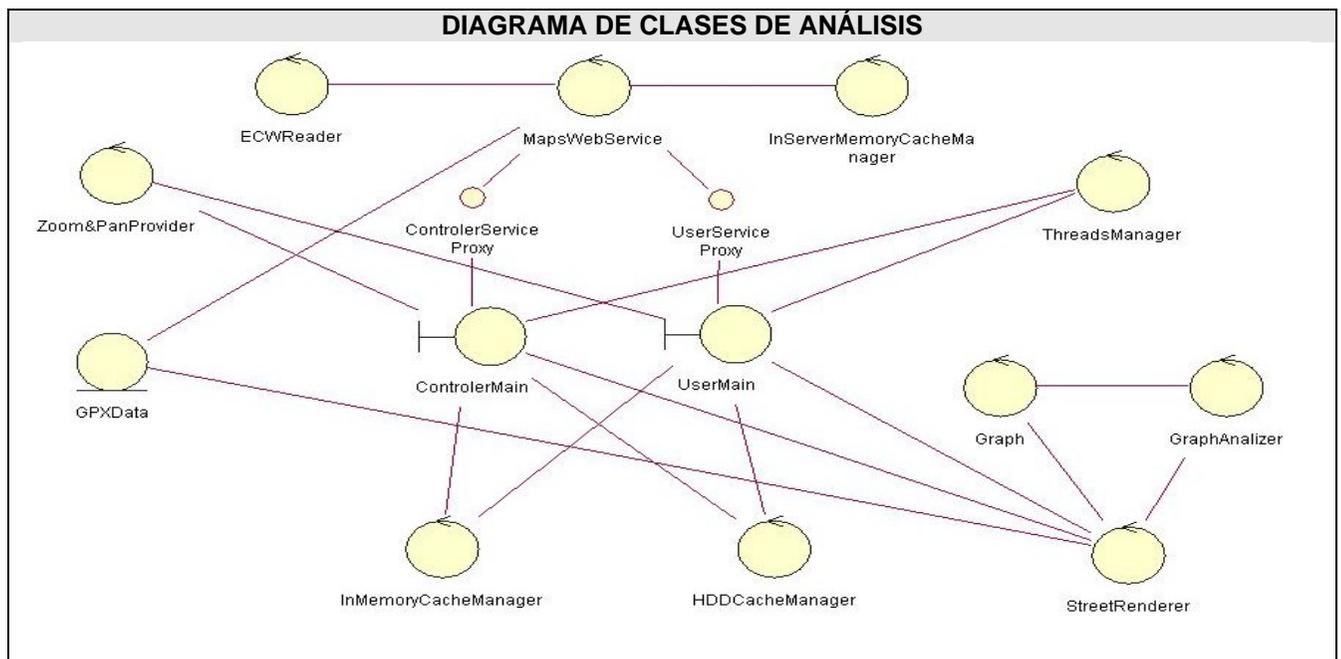
# Capítulo III

## ANÁLISIS Y DISEÑO DEL SISTEMA.

En este capítulo se muestran los resultados obtenidos en el desarrollo de los procesos de análisis y diseño del sistema, así como los diagramas que fueron necesarios para obtener una mayor claridad a la hora de elaborar la solución que se propone.

### 3.1 ANÁLISIS.

En el Diagrama de Clases del Análisis se muestran los conceptos básicos del sistema, sus partes y relaciones. Éste se realiza a través de un diagrama de clases de UML simplificado, en el cual se representan las clases y las asociaciones preliminares entre ellas.

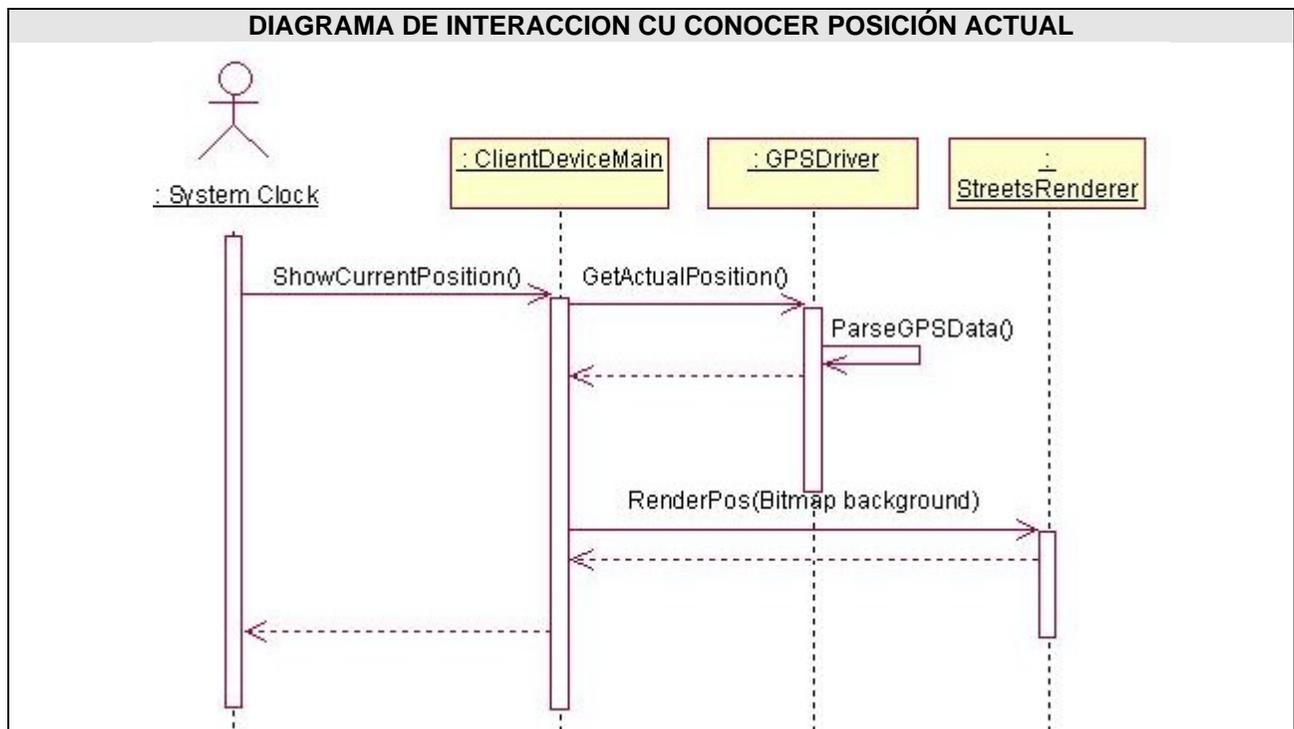


## 3.2 DISEÑO.

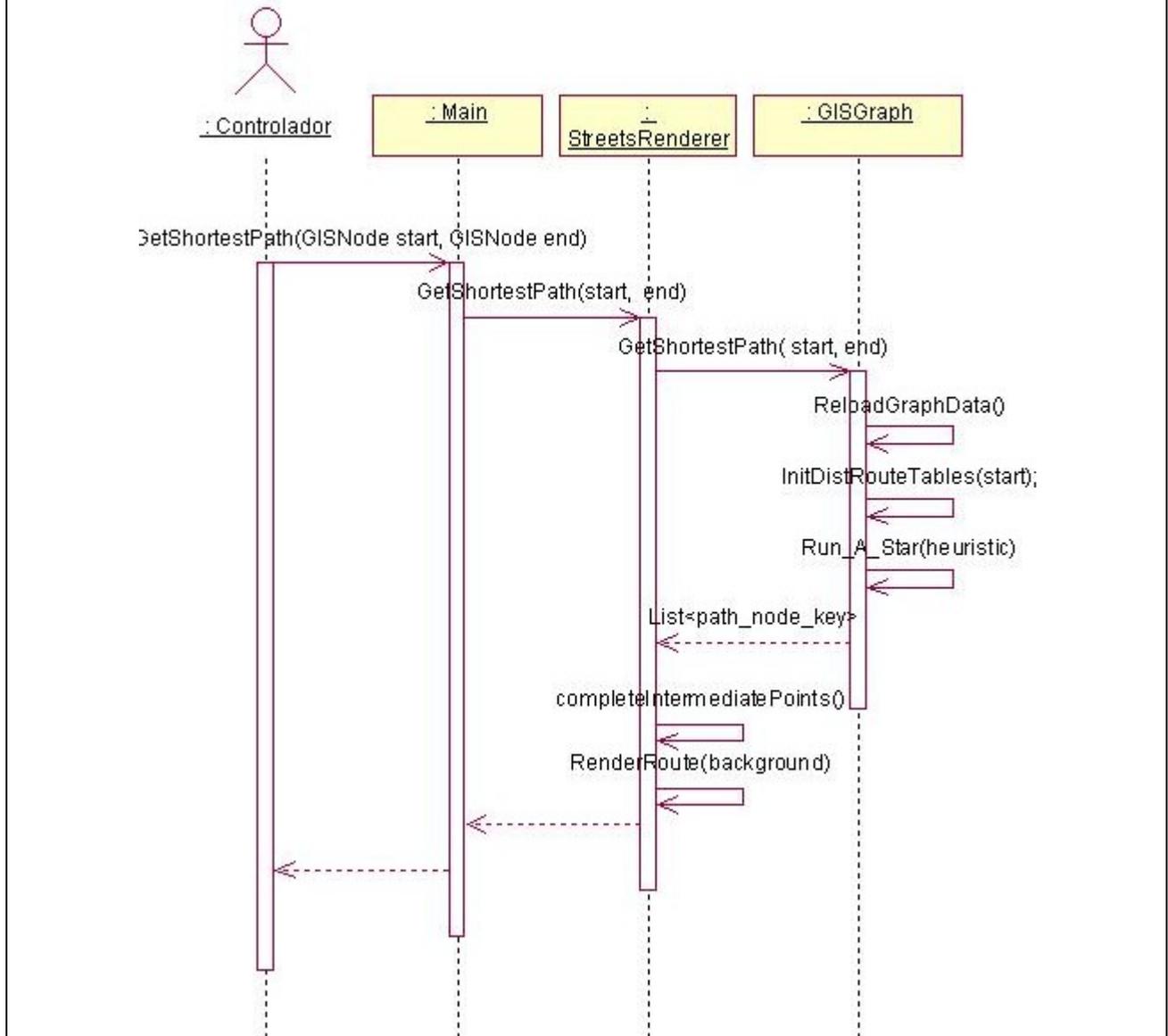
El diseño es parte fundamental del proceso de desarrollo de software. Su propósito primario es decidir cómo el sistema se llevará a cabo. Durante el diseño, se toman decisiones estratégicas y tácticas para cumplir los requerimientos funcionales y de calidad de un sistema, además se refina la visión obtenida en el Análisis.

### 3.2.1 DIAGRAMAS DE INTERACCIÓN.

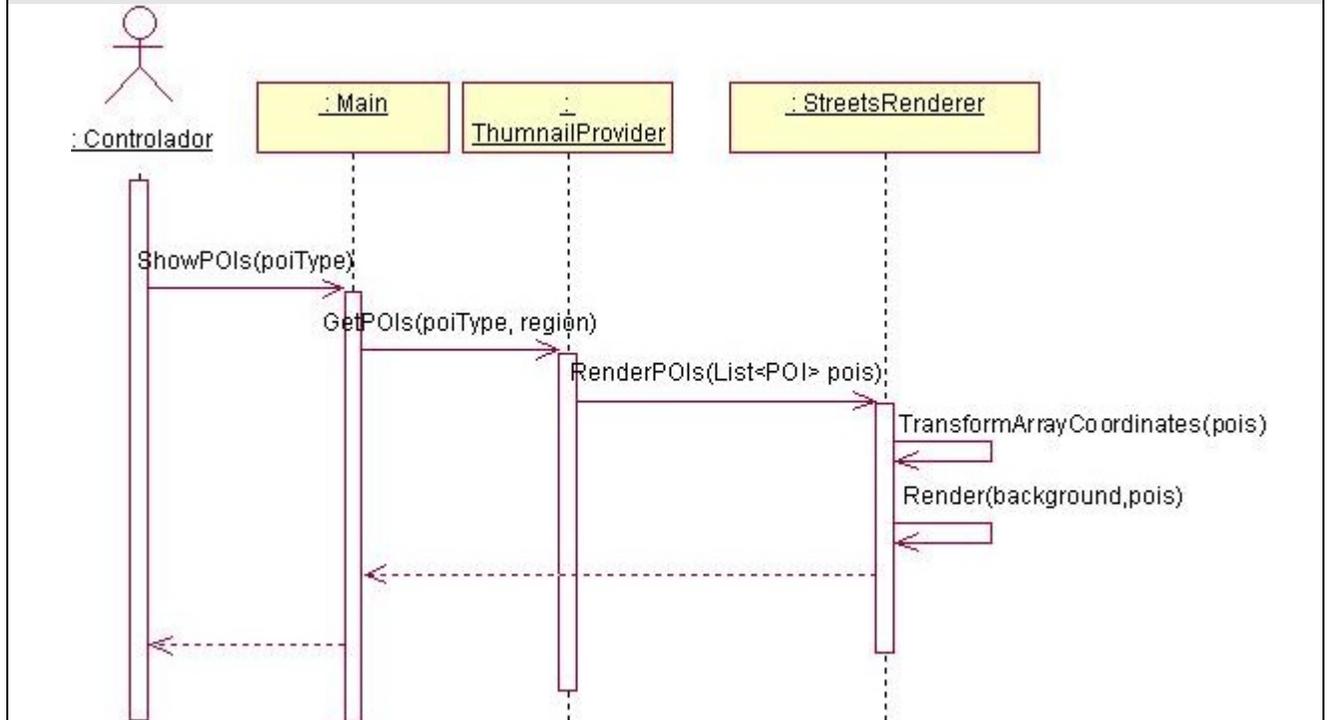
Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. La mayoría de las veces, esto implica modelar instancias concretas o prototípicas de clases, interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. Los diagramas de interacción pueden utilizarse para visualizar, especificar, construir y documentar la dinámica de una sociedad particular de objetos, o se pueden utilizar para modelar un flujo de control particular de un caso de uso.



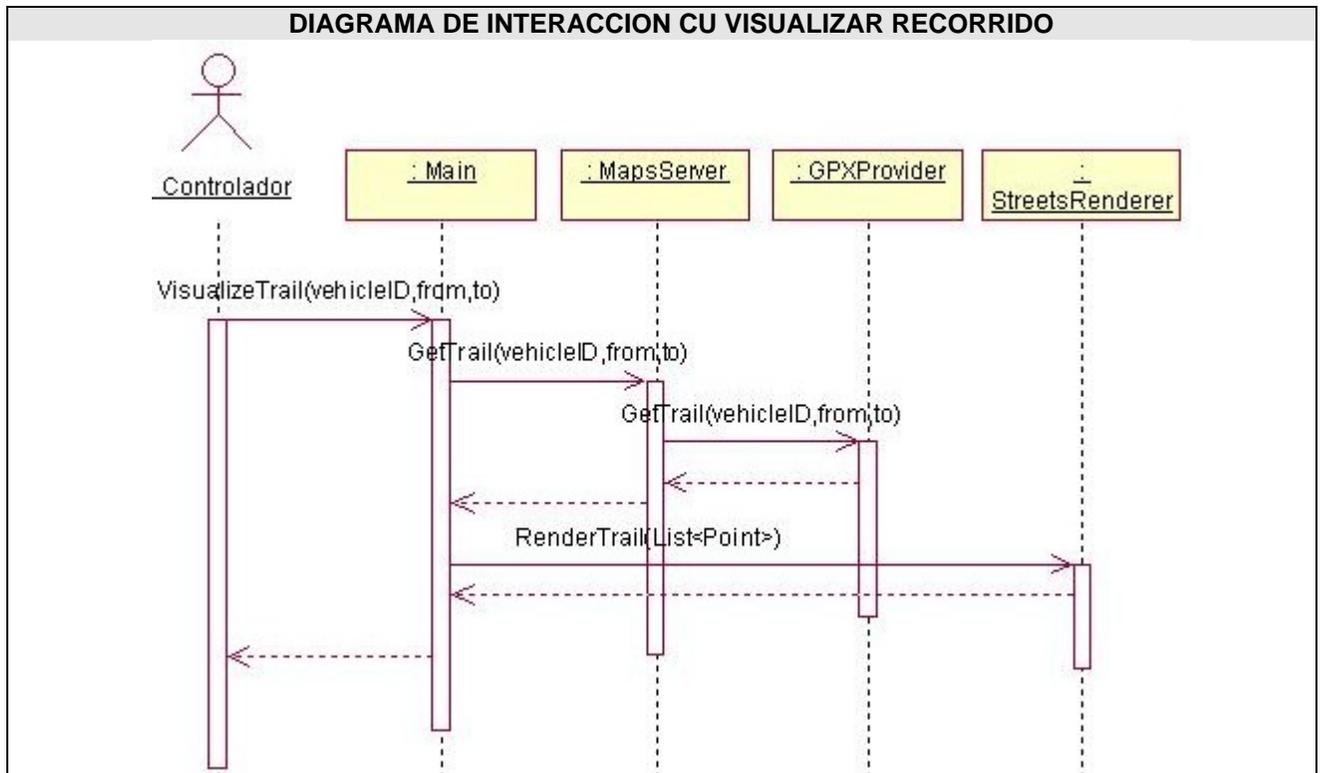
### DIAGRAMA DE INTERACCION CU DIRECCIONAR HACIA PUNTO DE INTERES



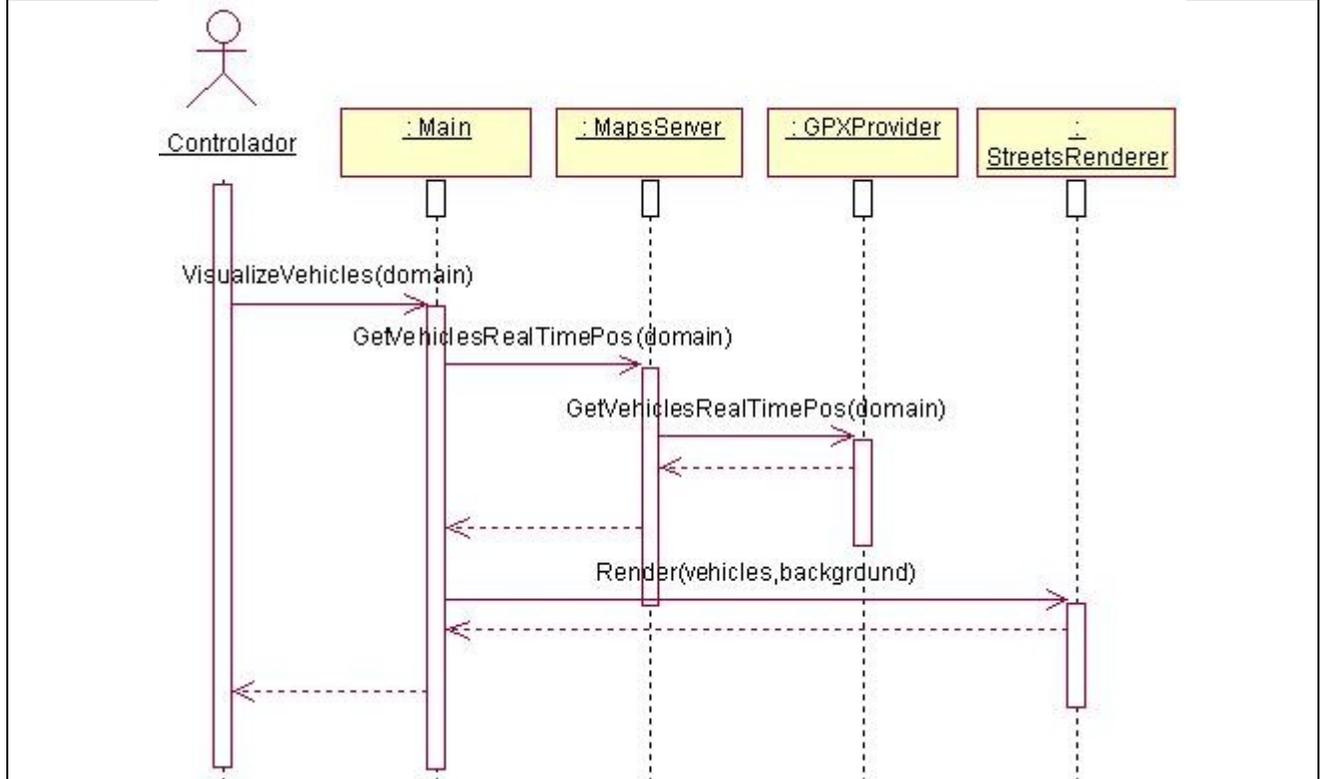
### DIAGRAMA DE INTERACCION CU VISUALIZAR PUNTOS DE INTERES.



### DIAGRAMA DE INTERACCION CU VISUALIZAR RECORRIDO

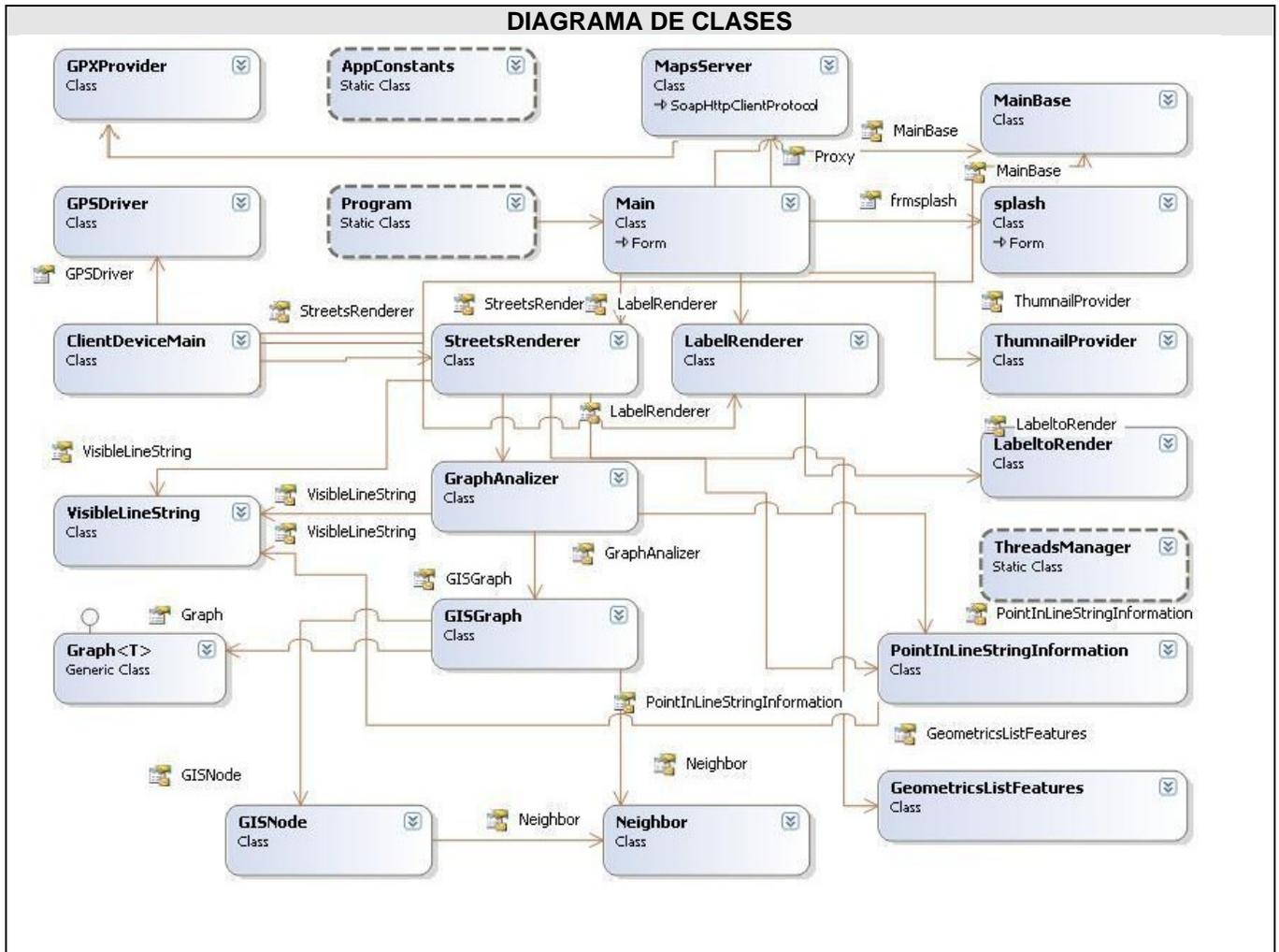


### DIAGRAMA DE INTERACCION CU VISUALIZAR VEHICULOS DOMINIO



### 3.2.2 DIAGRAMAS DE CLASES.

Los diagramas de clases son los más utilizados en el modelado de sistemas orientados a objetos. Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema, y además se obtiene como resultado del refinamiento del modelo conceptual.



## DIAGRAMA DE CLASES

**Program**  
Static Class

- Properties
  - frmMain
- Methods
  - Main

**LabelRenderer**  
Class

- Fields
- Properties
  - LabeltoRender
  - Listoflabels
- Methods
  - getLabels
  - setCanvas
  - setLabels
  - TransformArrayC...
  - TransformPointC...

**LabeltoRender**  
Class

- Fields
- Properties
  - Angle
  - LabelPoint
  - Maxvisibility
  - Minvisibility
  - Name
- Methods
  - LabeltoRender (+...

**Main**  
Class  
→ Form

- Fields
- Properties
  - frmsplash
  - LabelRenderer
  - MainBase
  - Proxy
  - StreetsRenderer
  - ThumbnailProvider
- Methods
  - btnComputeGrap...
  - btnDebug\_Click
  - buttonEnclose\_Cli...
  - close\_Button\_Click
  - close\_on\_scape\_...
  - Dispose
  - InitializeCompon...
  - Layer\_Click
  - Main
  - Main\_FormClosing
  - Main\_Load
  - Main\_Shown
  - makeWindowsEs...
  - MarkStartOrEnd
  - render\_Loop\_tim...
  - RenderRec
  - RoadsPCanvas\_M...
  - RoadsPCanvas\_M...
  - setPoints
  - ShowStreets\_But...
  - ViewsTabControl...

**GPSDriver**  
Class

- Methods
  - GetActualPosition
  - GPSDriver

**MainBase**  
Class

- Fields
- Methods
  - ByteToBoundingB...
  - CalculatePointOn...
  - CalculateViewing...
  - CleanHDDCache
  - CompleteWithSq...
  - DrawStreets
  - getFromHDDCache
  - getImage
  - getImagePath
  - getLayerNames
  - getLevel
  - HDDCacheConsta...
  - setToHDDCache
  - SetupStreetRend...
  - userMoving

**ClientMain**  
Class  
→ Form

- Fields
- Properties
- Methods
  - ClientMain
  - Dispose
  - InitializeCompon...
  - render\_Loop\_tim...
  - RoadsPCanvas\_M...
  - RoadsPCanvas\_M...
  - setPoints

**splash**  
Class  
→ Form

- Fields
- Methods
  - Dispose
  - ImageLoaderTim...
  - InitializeCompon...
  - on\_MainLoaded
  - splash
  - splash\_FormClosi...
  - splash\_Load
  - SplashCloserTime...

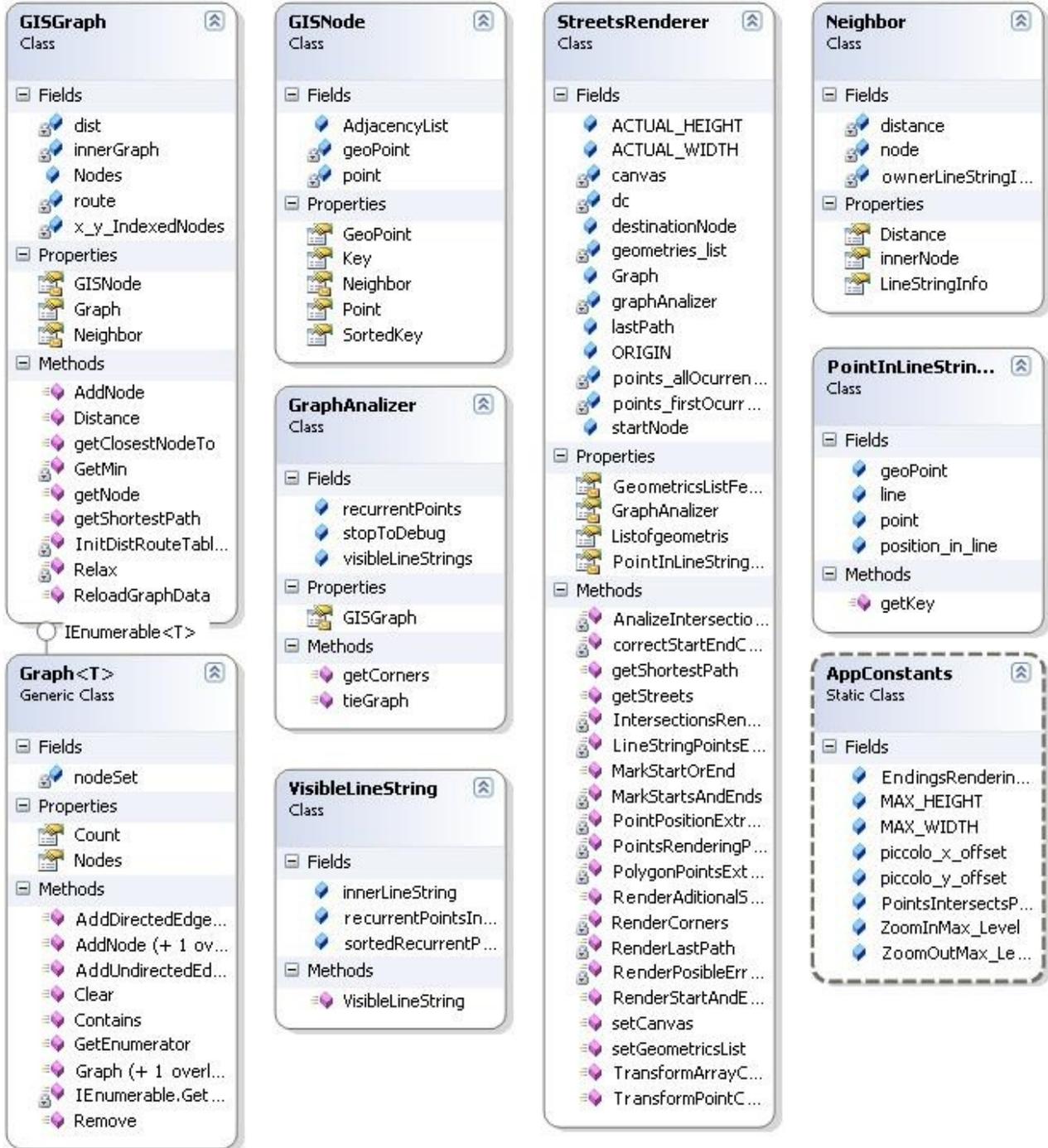
**GeometricsListF...**  
Class

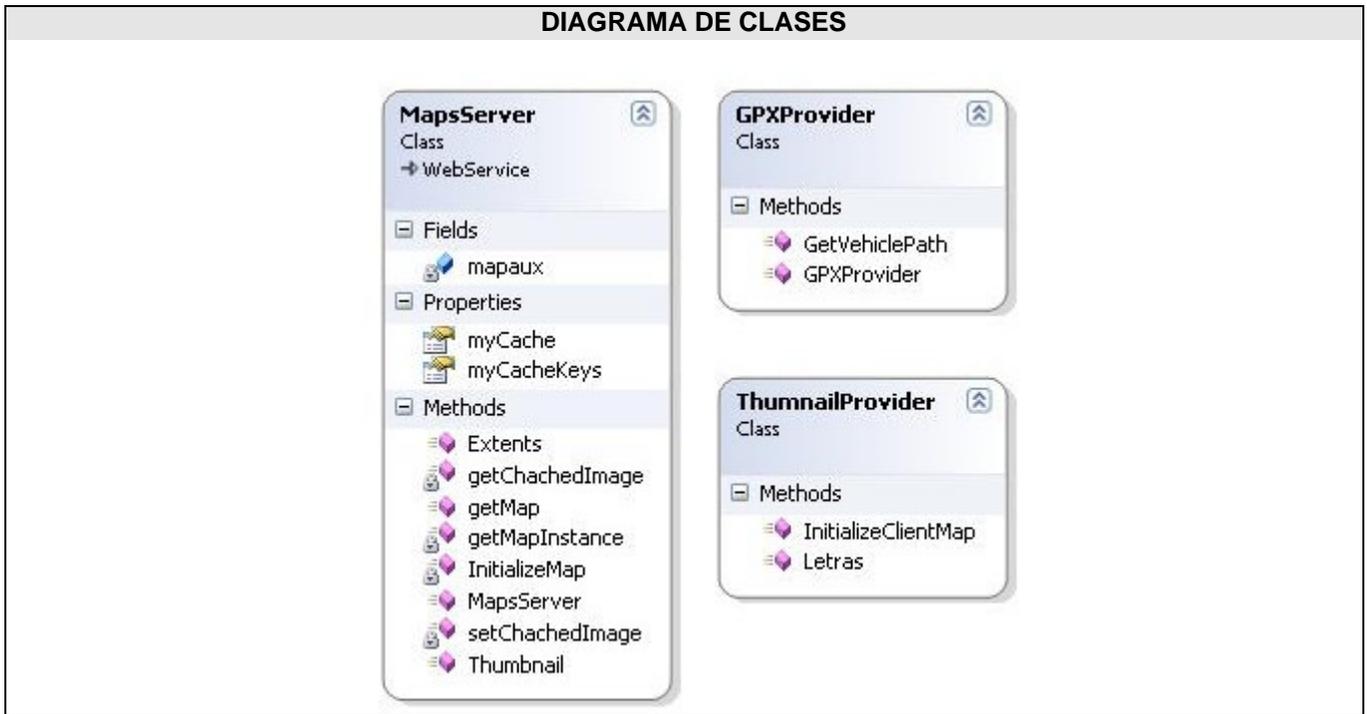
- Fields
  - geometricsList
  - geometrisIcon
  - maxvisibility
  - minvisibility
- Properties
  - GeometricsList
  - GeometrisIcon
  - Maxvisibility
  - Minvisibility
- Methods
  - GeometricsListFe...

**ThreadsManager**  
Static Class

- Fields
- Properties
  - CurrentRunningT...
- Methods
  - AddThread
  - Pop
  - ThreadsManager

## DIAGRAMA DE CLASES





### 3.2.3 DESCRIPCIÓN DE LAS CLASES.

<b>Nombre: MainBase</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	SetupStreetRenderer
Descripción:	Carga los datos necesarios y los prepara para su graficación.
Nombre:	SetToHDDCache
Descripción:	Guarda imágenes en el Isolated Storage.
Nombre:	GetFromHDDCache
Descripción:	Obtiene imágenes del Isolated Storage.
Nombre:	CalculateViewingRect
Descripción:	Calcula y define la región visible del mapa, la cual sirve para el posicionamiento o georeferenciación, de todos los elementos.
Nombre:	CompleteWithSquares
Descripción:	Se encarga de sustituir, al hacer zoom, la imagen visible por una formada por rectángulos cada uno con la misma definición de la imagen original.
Nombre:	DrawStreets
Descripción:	Es el que se encarga de invocar cada uno de los métodos del StreetsRenderer para graficar calles, puntos de interés, etc.
Nombre:	getImagePath
Descripción:	Dadas las coordenadas de una supuesta imagen y un nivel de zoom determinado, determina el "camino" de la imagen, que consiste en una consecución de Q, R, S, T, donde cada letra indica el cuadrante en que se encuentra la imagen en cada nivel de zoom.

<b>Nombre: Main</b>	
<b>Tipo de clase: Interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
Proxy	MapsServer
RoadsViewOfRoads	PIImage
HybridViewOfRoads	PIImage
street_renderer	StreetsRenderer
last_viewbounds_coordinates	PointF
cache_manager	CacheManager
<b>Para cada responsabilidad:</b>	
Nombre:	render_Loop_Timer_click
Descripción:	Es el manipulador que controla el ciclo de graficación.

<b>Nombre: ClientMain</b>	
<b>Tipo de clase: Interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
Proxy	MapsServer
RoadsViewOfRoads	PIImage
street_renderer	StreetsRenderer
last_viewbounds_coordinates	PointF
cache_manager	CacheManager
<b>Para cada responsabilidad:</b>	
Nombre:	render_Loop_Timer_click
Descripción:	Es el manipulador que controla el ciclo de graficación.
Nombre:	setPoints
Descripción:	Asigna los puntos que obtiene el GPSTDriver.

<b>Nombre: ThreadsManager</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
threadList	ArrayList
currentRunningThreads	int
<b>Para cada responsabilidad:</b>	
Nombre:	ThreadsManager
Descripción:	Constructor que se encarga de crear el threadList sincronizado
Nombre:	AddThread
Descripción:	Adiciona un nuevo hilo a la cola de hilos en espera de ejecución
Nombre:	Pop
Descripción:	Remueve un hilo de la cola y lo inicia.

<b>Nombre: MapsServer</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
map	SharpMap.Map
myCache	System.Collections.Hashtable
myCacheKeys	ArrayList
<b>Para cada responsabilidad:</b>	
Nombre:	getMap
Descripción:	Extrae dinámicamente la porción del ECW que corresponde al camino QRST indicado.
Nombre:	InitializeMap
Descripción:	Inicializa el mapa con los diferentes ECW y define a partir de que nivel se debe ver cada cual.

<b>Nombre: GPXProvider</b>	
<b>Tipo de clase : Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	GetVehiclePath
Descripción:	Devuelve la ruta seguida por un vehículo en un período de tiempo determinado

<b>Nombre: GPSDriver</b>	
<b>Tipo de clase : Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	GetActualPosition
Descripción:	Obtiene la posición actual del dispositivo GPS.

<b>Nombre: StreetsRenderer</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
ORIGIN	Sharpmap.Geometries.Point
ACTUAL_WIDTH	double
ACTUAL_HEIGHT	double
canvas	Bitmap
startNode	GISNode
destinationNode	GISNode
graphAnalyzer	GraphAnalyzer
Graph	GISGraph

<b>Para cada responsabilidad:</b>	
Nombre:	getStreets
Descripción:	Grafica las calles, puntos de interés, etc., correspondientes a la parte actualmente visible del mapa.
Nombre:	LineStringPointsExtractor
Descripción:	Extrae los puntos significativos de un LineString determinado.
Nombre:	AnalizeIntersectionsAndRenderPoints
Descripción:	Analiza los puntos significativos que puedan llegar a formar parte del grafo.
Nombre:	getShortestPath
Descripción:	Le pide a su grafo interno la ruta mínima, completa los puntos intermedios y grafica.

<b>Nombre: LabelRenderer</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
listoflabels	List<List<LabeltoRender>>
<b>Para cada responsabilidad:</b>	
Nombre:	getLabels
Descripción:	Grafica los identificadores de calles, puntos de interés, etc.

<b>Nombre: ThumbnailProvider</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo:</b>	<b>Tipo:</b>
<b>Para cada responsabilidad:</b>	
Nombre:	InitializeClientMap
Descripción:	Inicializa la información geográfica a graficar

<b>Nombre: VisibleLineString</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
innerLineString	LineString
recurrentPointsInformation	Hashtable
sortedRecurrentPointsInformation	SortedList<int, PointInLineStringInformation>
<b>Para cada responsabilidad:</b>	
Nombre:	
Descripción:	

<b>Nombre: GraphAnalyzer</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>

recurrentPoints	Hashtable
visibleLineStrings	List<VisibleLineString>
<b>Para cada responsabilidad:</b>	
Nombre:	getCorners
Descripción:	Discrimina los puntos recurrentes no significativos y devuelve aquellos que realmente los son más los inicios y fines de los LineStrings, adicionándolos al grafo.
Nombre:	tieGraph
Descripción:	Se encarga de unir los nodos del grafo con sus vecinos determinando también la distancia existente entre ellos.

<b>Nombre: LabeltoRender</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
name	string
labelPoint	SharpMap.Geometries.Point
maxvisibility	double
minvisibility	double
<b>Para cada responsabilidad:</b>	
Nombre:	
Descripción:	

<b>Nombre: GISGraph</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
dist	Hashtable
route	Hashtable
innerGraph	Graph<string>
<b>Para cada responsabilidad:</b>	
Nombre:	AddNode
Descripción:	Adiciona un nodo al grafo.
Nombre:	getNode
Descripción:	Devuelve un nodo del grafo
Nombre:	getShortestPath
Descripción:	Devuelve el camino más corto entre dos nodos del grafo dada una heurística determinada.

<b>Nombre: GISNode</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
AdjacencyList	List<Neighbor>
geopoint	SharpMap.Geometries.Point
Point	SharpMap.Geometries.Point
<b>Para cada responsabilidad:</b>	
Nombre:	
Descripción:	

<b>Nombre: Neighbor</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
node	GISNode
distance	float
ownerLineStringInfo	PointInLineStringInformation
<b>Para cada responsabilidad:</b>	
Nombre:	
Descripción:	

<b>Nombre: GeometricsListFeatures</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
geometricsList	Collection<SharpMap.Geometries.Geometry>
geometricsIcon	Icon
maxvisibility	double
minvisibility	double
<b>Para cada responsabilidad:</b>	
Nombre:	
Descripción:	

<b>Nombre: PointInLineStringInformation</b>	
<b>Tipo de clase: Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
geoPoint	SharpMap.Geometries.Point
point	SharpMap.Geometries.Point
line	VisibleLineString
position_in_line	int
<b>Para cada responsabilidad:</b>	
Nombre:	
Descripción:	

### 3.3 CONCLUSIONES.

Durante este capítulo, se muestra el desarrollo del análisis y el diseño de nuestro sistema. Primeramente se brinda el diagrama de las clases del análisis en donde se exponen los conceptos básicos del sistema, sus partes y relaciones. Luego, los diagramas de interacción por cada uno de los casos de uso y el diagrama de clases del diseño.

# Capítulo IV

## IMPLEMENTACIÓN DEL SISTEMA.

En la implementación, se parte del resultado del diseño y se desarrolla el sistema en términos de componentes, es decir, ficheros de código fuente, ficheros de código binario, ejecutables y similares.

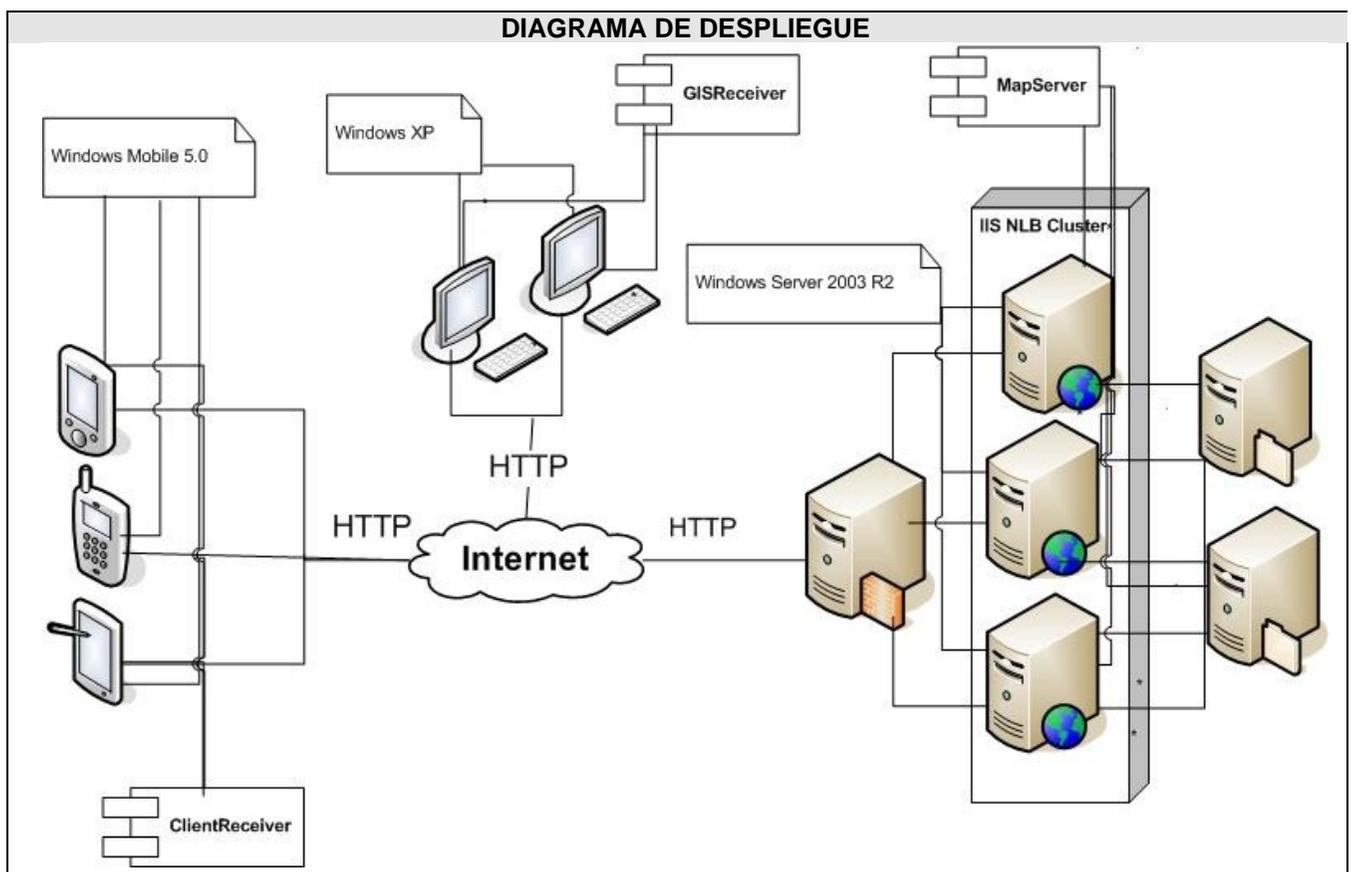
Los diagramas de despliegue y componentes conforman lo que se conoce como un modelo de implementación. Describen los componentes a construir, su organización y dependencia entre nodos físicos en los que funcionará a aplicación.

A continuación le mostramos el modelo de implementación de nuestro sistema como resultado del diseño realizado previamente.

## 4.1 MODELO DE IMPLEMENTACIÓN.

### 4.1.1 DIAGRAMA DE DESPLIEGUE.

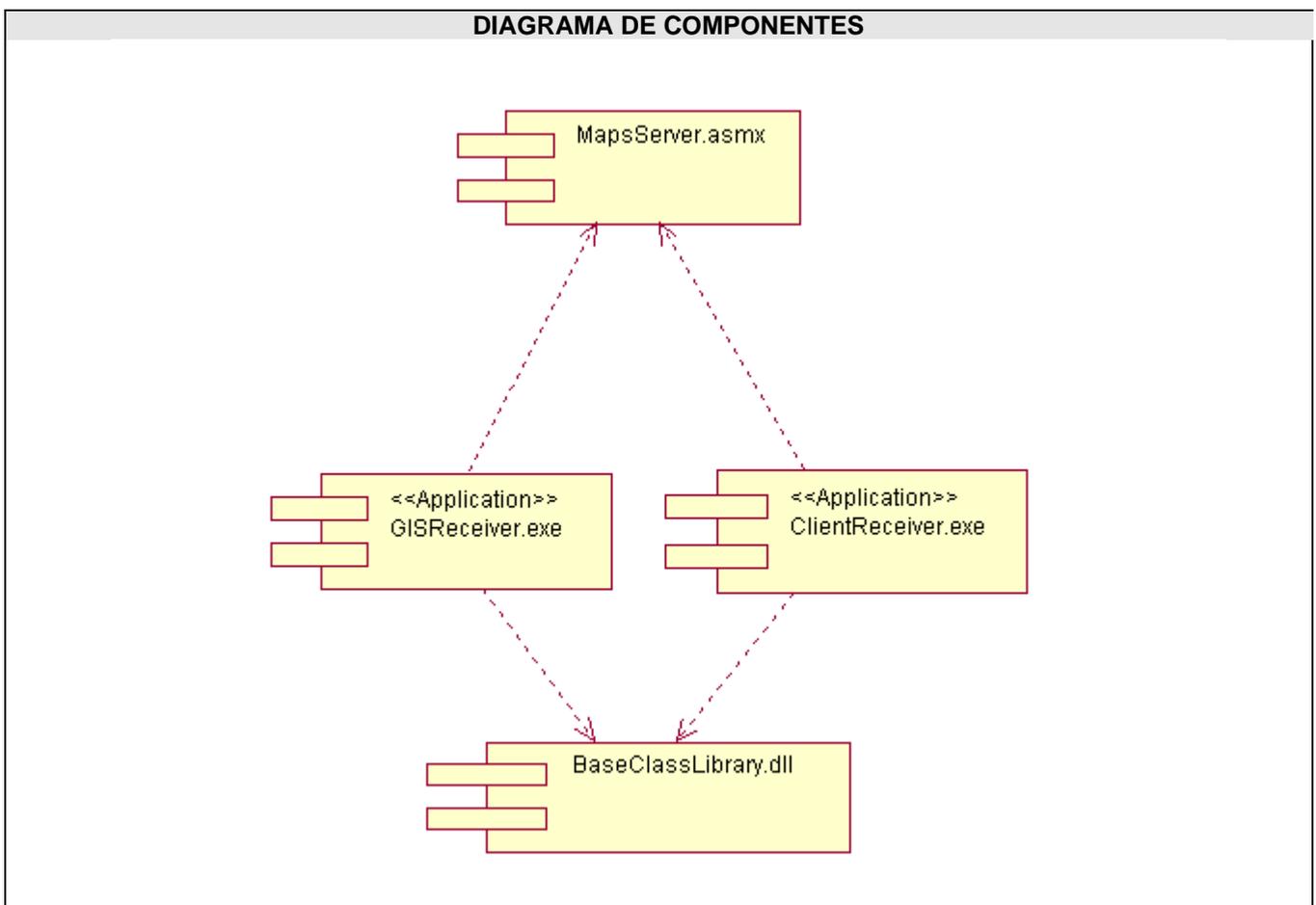
El diagrama de despliegue muestra la asignación de artefactos concretos de software, como ejecutables, a nodos computacionales, o sea, el despliegue de los mismos en la arquitectura física y la comunicación entre los elementos de esta última. Nuestra aplicación tiene tres elementos fundamentales: cliente inteligente (ClientReceiver), que sería desplegado en los dispositivos GPS que portarían los conductores, aplicación controladora (GISReceiver), desplegado en las estaciones de trabajo de los controladores y el servidor (MapServer), servicio web desplegado en la granja de servidores sobre un clúster NLB basado en Windows Server 2003. Toda la comunicación sería por el protocolo HTTP.



## 4.1.2 DIAGRAMA DE COMPONENTES.

Lo que distingue a este tipo de diagramas es su contenido: componentes, interfaces y relaciones entre ellos. También puede contener paquetes utilizados para agrupar elementos del modelo.

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, sean éstos componentes de código fuente, binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo.



## 4.2 CONCLUSIONES.

A lo largo de este capítulo se da una visión general de como quedó finalmente implementada nuestra aplicación, mostrando la misma en términos de componentes, su ubicación y utilización a la hora de la implantación final.

# Conclusiones

Con este trabajo se desarrolló un sistema informático con la capacidad de gestionar, posicionar y direccionar vehículos basado en la información de dispositivos GPS. Todos los datos que usa o genera se acogen a los estándares internacionales que rigen este tipo de sistemas, tanto para la compresión de imágenes, como para la representación de rutas y puntos de interés. Estos estándares han sido seleccionados después de un análisis exhaustivo de las tendencias y recomendaciones emitidas por las organizaciones internacionales dedicadas al tema, como el Open GIS Consortium.

Hemos propuesto una arquitectura basada en las posibilidades reales de nuestro país, y donde es posible incluso prescindir de algunas de las partes constituyentes en dependencia de presupuestos, tipos de dispositivos y conectividad. Las interfaces de usuario han sido diseñadas con un balance entre el atractivo visual y el aprovechamiento de la experiencia del usuario.

Los algoritmos de direccionamiento están abiertos a la extensión y personalización sobre la base de las necesidades específicas en un escenario de despliegue y uso determinados. Estableciendo, no obstante, una base algorítmica de direccionamiento, usando para ello los modelos de estimación heurística de recorridos más populares en el mundo en este tipo de sistemas.

Por todo lo anterior se concluye que los objetivos propuestos para el presente trabajo han sido cumplidos. Se incluyen recomendaciones que deben tenerse en cuenta para la continuación futura del trabajo.

# Recomendaciones

A lo largo del desarrollo del trabajo, han surgido ideas que podrían implementarse en un futuro, de forma que se logre una aplicación más abarcadora, para lo cual se recomienda:

1. Modificar el motor de graficación para que pueda sacar provecho del hardware gráfico instalado.
2. Ampliar el espectro de formatos soportados por la aplicación.
3. Realizar pruebas de calidad al sistema.

# Referencias Bibliográficas

- [1]. YEPES, E. L. Sistemas de Información geográfica, 2005  
<http://www.necochea.gov.ar/prensa128.htm>
- [2]. ORTIZ G. El funcionamiento del GPS: un repaso a los principales componentes, 2003  
<http://www.gabrielortiz.com/art.asp?Info=039>.
- [3]. FOSTER, D. GPX: the GPS Exchange Format, 2004  
<http://www.topografix.com/gpx.asp>
- [4]. Google Earth - la Tierra vista desde un satélite, 2007  
<http://google.dirson.com/o.a/google-earth/>
- [5]. PENALVA, J. P. Ford y Microsoft presentan Sync, 2007  
<http://xataka.com/2007/01/08-ces-2007-ford-y-microsoft-presentan-sync>
- [6]. Volkswagen and Google develop revolutionary navigation system, 2007  
[http://www.automotoportal.com/article/Volkswagen\\_and\\_Google\\_develop\\_revolutionary\\_navigation\\_system](http://www.automotoportal.com/article/Volkswagen_and_Google_develop_revolutionary_navigation_system)
- [7]. CHAPPELL, D. Windows Presentation Foundation (WPF), 2006  
<http://www.d2bnetwork.com/tecnologia/especializacion/wpf/default.aspx>
- [8]. ROJAS, N. R. Windows Communication Foundation (WCF), 2007  
<http://north18rojas.spaces.live.com/>
- [9]. AVENDAÑO, R. D. Acelerando Procesos de desarrollos, 2006  
<http://www.sistemasuni.edu.pe/AltaTecnologia/acelerandopro.html>
- [10]. MOHAN R. C. Introducción al Servicio de Cluster Server de Microsoft (MSCS) en Windows .NET Server 2003, 2002  
<http://www.microsoft.com/spanish/msdn/articulos/archivo/100103/voices/wns-introclustermcs.asp>
- [11]. Algoritmo de búsqueda A\*, 2007  
<http://es.wikipedia.org/wiki/A%2A>
- [12]. TRIGLAV J. ECW: ¿Compresión Wavelet más allá del límite?, 2000  
[http://www.mappinginteractivo.com/plantilla-ante.asp?id\\_articulo=617](http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=617)

# Bibliografía

- ANDERSON, S. J. *The Design and Implementation of Geographic Information Systems*. John Wiley and Sons, 2003. 1100 p.
- HARVEY, F. *A Primer of GIS: Fundamental Geographic and Cartographic Concepts* The Guilford Press 2007. 300 p.
- DRAGICEVIC, S. *Collaborative Geographic Information Systems*. Idea Group Publishing, 2006. 1250 p.
- SHARMA, J. *Frontiers of Geographic Information Technology*. Springer-Verlag, 2006. 980 p.
- ANDERSON, C. *Essential Windows Presentation Foundation* Addison-Wesley Professional 2007. 512 p. *Microsoft .NET Development Series*.
- ANDRADE, C. *Professional WPF Programming: .NET Development with the Windows Presentation Foundation* Wrox 2007. 480 p. *Wrox Professional Guides*.
- MACDONALD, M. *Pro WPF: Windows Presentation Foundation in .NET 3.0*. Apress 2007. 1000 p. *Pro*.
- GRADY BOOCH, J. R., IVAR JACOBSON. *The Unified Modeling Language User Guide*. Second Edition. Addison Wesley 2005. p.
- KRUCHTEN, P. *The Rational Unified Process: An Introduction*. Third Edition. Addison Wesley, 2003. p.
- LARMAN, C. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*. Third Edition. Addison Wesley, 2004. p.
- LOWY, J. *Programming WCF Services* O'Reilly Media, Inc., 2007. 634 p.
- MCMURTRY, C. *Windows Communication Foundation Unleashed* Sams 2007. 720 p. *Unleashed*.
- PEIRIS, C. *Pro WCF: Practical Microsoft SOA Implementation* Apress 2007. 500 p. *Pro*.

- YEPES, E. L. Sistemas de Información geográfica, 2005  
<http://www.necochea.gov.ar/prensa128.htm>
- ORTIZ, G. El funcionamiento del GPS: un repaso a los principales componentes, 2003  
<http://www.gabrielortiz.com/art.asp?Info=039>
- FOSTER, Dan .GPX: the GPS Exchange Format, 2004  
<http://www.topografix.com/gpx.asp>
- Google Earth - la Tierra vista desde un satélite, 2007  
<http://google.dirson.com/o.a/google-earth/>
- PENALVA, Javier P .Ford y Microsoft presentan Sync, 2007  
<http://xataka.com/2007/01/08-ces-2007-ford-y-microsoft-presentan-sync>
- Volkswagen and Google develop revolutionary navigation system, 2007  
[http://www.automotoportal.com/article/Volkswagen\\_and\\_Google\\_develop\\_revolutionary\\_navigation\\_system](http://www.automotoportal.com/article/Volkswagen_and_Google_develop_revolutionary_navigation_system)
- CHAPPELL David. Windows Presentation Foundation (WPF), 2006  
<http://www.d2bnetwork.com/tecnologia/especializacion/wpf/default.aspx>
- ROJAS, N. R. Windows Communication Foundation (WCF), 2007  
<http://north18rojas.spaces.live.com/>
- AVENDAÑO, R. D. Acelerando Procesos de desarrollos, 2006  
<http://www.sistemasuni.edu.pe/AltaTecnologia/acelerandopro.html>
- MOHAN R. C. Introducción al Servicio de Cluster Server de Microsoft (MSCS) en Windows .NET Server 2003, 2002  
<http://www.microsoft.com/spanish/msdn/articulos/archivo/100103/voices/wns-introclustermcs.asp>
- Algoritmo de búsqueda A\*, 2007  
<http://es.wikipedia.org/wiki/A%2A>
- TRIGLAV, J. ECW: ¿Compresión Wavelet más allá del límite?, 2000  
[http://www.mappinginteractivo.com/plantilla-ante.asp?id\\_articulo=617](http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=617)
- BOSWORTH, A. XML, SOAP y datos binarios, 2003  
[http://www.microsoft.com/spanish/msdn/articulos/archivo/040403/voices/infofet\\_whitepaper.asp](http://www.microsoft.com/spanish/msdn/articulos/archivo/040403/voices/infofet_whitepaper.asp)
- SOAP Message Transmission Optimization Mechanism, 2005  
<http://www.w3.org/TR/soap12-mtom/>

# Glosario de términos

**Bluetooth:** Es la norma que define un estándar global de comunicación inalámbrica, que posibilita la transmisión de voz y datos entre diferentes equipos mediante un enlace por radiofrecuencia.

**Datum:** Modelo matemático de la Tierra obtenido a partir de cálculos geodésicos..

**GPS:** Sistema de Posicionamiento Global.

**GIS:** Sistemas de información geográfica.

**GPX:** GPS Exchange format, formato para la transmisión y almacenamiento de información brindada por GPS.

**HTTP:** HyperText Transfer Protocol (Protocolo de transferencia de hipertexto). Es el protocolo usado para intercambiar archivos (texto, gráfica, imágenes, sonido, video y otros archivos multimedia) en la World Wide Web.

**iPod:** El iPod es un reproductor de música digital basado en un disco duro creado por Apple Computer. Puede reproducir archivos MP3, WAV, AAC/M4A, AIFF y Apple Lossless. La capacidad del disco duro es de hasta 60 GB (hasta 15.000 canciones o 25.000 fotos el modelo superior).

**PND:** Personal Navigation Device. Dispositivos usados para la recepción y la representación de la información GPS.

**Pocket PC:** PocketPC es un ordenador de bolsillo. Se trata de un pequeño ordenador, diseñado para ocupar el mínimo espacio y ser fácilmente transportable que ejecuta el sistema operativo Windows CE de Microsoft, el cual le proporciona capacidades similares a los PCs de escritorio.

**SMS:** El servicio de mensajes cortos o SMS (Short Message Service) es un servicio disponible en los teléfonos móviles que permite el envío de mensajes cortos (también conocidos como mensajes de texto, o más coloquialmente, textos o incluso txts o msjs) entre teléfonos móviles, teléfonos fijos y otros dispositivos de mano.

**SOAP:** SOAP (siglas de Simple Object Access Protocol) es un protocolo estándar creado por el W3C que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. SOAP es uno de los protocolos utilizados en los servicios Web.

**TCP:** Protocolo de redes, orientado a conexión y confiable, que forma parte del conjunto de protocolos de TCP/IP.

**Zune:** Un reproductor multimedia creado por Microsoft