

**PLATAFORMA WEB 2.0 PARA UN SISTEMA GPS- GSM/GPRS DE
MONITOREO EN TIEMPO REAL PARA EL TRANSPORTE PÚBLICO DE
LA CIUDAD DE PASTO**

**JOHN ALEXANDER ORDOÑEZ IMBACUAN
ADRIAN FERNANDO MARTINEZ ORTEGA**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE SISTEMAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2010**

**PLATAFORMA WEB 2.0 PARA UN SISTEMA GPS - GSM/GPRS DE
MONITOREO EN TIEMPO REAL PARA EL TRANSPORTE PÚBLICO DE
LA CIUDAD DE PASTO**

**JOHN ALEXANDER ORDOÑEZ IMBACUAN
ADRIAN FERNANDO MARTINEZ ORTEGA**

**Trabajo de grado presentado como requisito parcial para optar por el
título de Ingenieros de Sistemas**

**Director
JESUS INSUASTY PORTILLA
Ingeniero de Sistemas**

**Co – Director
ROLANDO BARAHONA CABRERA
Ingeniero Electrónico**

**UNIVERSIDAD DE NARIÑO
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE SISTEMAS
PROGRAMA DE INGENIERÍA DE SISTEMAS
SAN JUAN DE PASTO
2010**

NOTA DE ACEPTACIÓN

Firma del Presidente de Jurado

Firma del Jurado

Firma del Jurado

Firma del Asesor del Proyecto

San Juan de Pasto, noviembre 2010

“La Universidad de Nariño no se hace responsable por las opiniones o resultados obtenidos en el presente trabajo y para su publicación priman las normas sobre el derecho de autor.

Acuerdo 1, Artículo 324. Octubre 11 de 1966. Emanado del honorable Consejo Directivo de la Universidad de Nariño”

DEDICATORIA

Este trabajo está dedicado primero que nada a nuestras familias, que con su apoyo, su constancia y paciencia nos brindaron su comprensión en todo momento y en todo lugar. Llenaron con su presencia la fuerza que nos faltaba para seguir adelante en momentos difíciles, desde que empezamos nuestro camino en Ingeniería, hasta el final. No se olvidaron de apoyarnos en nuestras metas y siguieron paso a paso nuestro crecimiento como profesionales y como personas.

También, sin ser menos, dedicamos este resultado a todos los amigos que complementaron nuestras falencias y fortificaron aún más nuestras habilidades. Con ellos pasamos días y noches comprendiendo la lógica de muchas cosas sobre nuestras materias. Sin ellos hubiera sido muy difícil estar en la posición en la que nos encontramos ahora, demostrando nuestro conocimiento y nuestro profesionalismo ante el mundo.

A nuestros maestros, que confirmaron con su experiencia la realidad de la teoría. Ellos demostraron ser tan valiosos en el camino así como todas sus enseñanzas. Nos orientaron de la mejor forma e hicieron que seamos cómplices de nuestra propia confianza. Son un ejemplo a seguir como profesionales y un orgullo como compañeros en el futuro.

Dedicamos esta parte a todos nuestros demás seres queridos, nuestros seres amados, que nos motivaron a no retroceder ante las adversidades ni a ser débiles ante los problemas. Son más que una razón para seguir adelante, conquistando nuestros sueños e ilusiones desde ahora y siempre.

Todos ellos comparten sus alegrías y sus tristezas, sus triunfos y derrotas y a pesar de todo esto seguirán ahí, siendo guías en nuestro camino, un camino que apenas empieza.

Agradecemos de todo corazón que ustedes siempre estén con nosotros.

AGRADECIMIENTOS

Agradecemos muy profundamente el apoyo inmenso que nos ha brindado el Ingeniero Jesús Insuasty, como mentor de nuestro grupo de investigación Galeras.Net, así como también maestro y amigo en gran parte de nuestra formación como profesionales. Sus enseñanzas han sido causa de este triunfo para la vida.

Agradecemos la gran ayuda del Ingeniero Rolando Barahona, que con sus conocimientos y sus enseñanzas, permitió que el resultado del producto sea el mejor. No olvidaremos su dedicación y compromiso para con nuestro grupo.

Al Director de nuestro Programa de Ingeniería de Sistemas, Ingeniero Vicente Chamorro, al Ingeniero Jairo Guerrero y al grupo selecto de Docentes que pusieron a toda la disposición el espacio físico, tecnológico y cognitivo para que este proyecto cumpliera con las mejores expectativas dentro y fuera de la Universidad.

A los colegas de la Facultad de Ingeniería, del Departamento de Ingeniería Electrónica, Director y docentes, que hicieron posible que el resultado de nuestra tesis sea consecuencia del trabajo interdisciplinar entre carreras.

En general a todas aquellas personas que hicieron posible que aportaron con su granito de arena al resultado de éste proyecto.

CONTENIDO

Tabla de Contenido

GLOSARIO	14
RESÚMEN	15
ABSTRACT	16
INTRODUCCIÓN	18
CAPITULO 1 MARCO TEÓRICO	31
1.1 SISTEMA DE POSICIONAMIENTO GLOBAL	31
1.2 DATA WAREHOUSE	32
1.3 BUSINESS INTELLIGENCE	34
1.4 MASHUP	36
1.4.1 Tipos de Mashups.....	38
1.5 EXTREME PROGRAMMING.....	40
CAPITULO 2 ANTECEDENTES.....	43
CAPITULO 3 HERRAMIENTAS DE DESARROLLO PARA LA CONSTRUCCIÓN DE LA PLATAFORMA WEB 2.0 DEL SISTEMA.....	45
3.1 LENGUAJE DE PROGRAMACIÓN	45
3.1.1 <i>Html</i>	45
3.1.2 <i>Javascript</i>	45
3.1.3 <i>Lenguaje de programación c sharp sobre .net</i>	46
3.2 ENTORNO DE DESARROLLO.....	47
3.2.1 <i>Microsoft visual studio 2010</i>	47
3.2.2 <i>Asp.net</i>	47
3.3 SGBD SISTEMA GESTOR DE BASES DE DATOS.....	48
3.3.1 <i>Sql server 2008 r2</i>	48
CAPITULO 4 METODOLOGÍA DE ANÁLISIS, DISEÑO Y MODELADO.....	50
4.1 FASE DE EXPLORACIÓN.....	50

4.2 ITERACIONES	50
4.3 FASE DE PRODUCCIÓN	51
4.4 TÉCNICAS DE RECOLECCIÓN Y ANÁLISIS DE LOS DATOS.....	52
4.4.1 <i>La forma</i>	52
4.4.2 <i>El contenido</i>	53
4.5 IDENTIFICACIÓN Y CARACTERIZACIÓN DE LA INNOVACIÓN PROPUESTA	54
4.6 LENGUAJE UNIFICADO DE MODELADO	54
4.6.1 <i>Tipos de conectores</i>	55
4.6.2 <i>Diagrama de casos de uso</i>	56
4.6.3 <i>Diagrama de clases</i>	56
 CAPITULO 5 DESARROLLO DE LA PLATAFORMA WEB 2.0.....	 58
5.1 FASE DE PLANIFICACIÓN	58
5.1.1 <i>Plan de entrega</i>	58
5.2 DISEÑO.....	60
5.2.1 <i>Requerimientos</i>	60
5.2.2 <i>Diagramas de clases de la aplicación web</i>	70
5.2.3 <i>Diagrama de clases – web service</i>	80
5.2.4 <i>Diagrama de clases de las tablas sobre la base de datos</i>	82
5.2.5 <i>diagrama de clases – servicio vms</i>	83
5.3 DATA WAREHOUSE DEL SISTEMA DE MONITOREO VEHICULAR EN TIEMPO REAL	83
 CAPITULO 6 DESARROLLO	 90
6.1 INTRODUCCIÓN.....	90
6.2 ARQUITECTURA DE LA PLATAFORMA	90
6.3 MÓDULO PRINCIPAL	90
6.4 MÓDULO VMS	91
6.4.1 <i>What's vms</i>	93
6.4.2 <i>Research objectives</i>	93
6.4.3 <i>System architecture</i>	94
6.5 APLICACIÓN VMS.....	94
6.5.1 <i>Pestaña home – botones de control</i>	94
6.5.2 <i>Pestaña bus owner</i>	95
6.5.3 <i>Pestaña Enterprise</i>	96
6.5.4 <i>Mapa</i>	96
6.6 MÓDULO MANAGEMENT	98
6.6.1 <i>Vehicles</i>	98

6.6.2 <i>Enterprises</i>	101
6.6.3 <i>Assignment</i>	104
CAPITULO 7 IMPLEMENTACIÓN.....	107
7.1 INTRODUCCIÓN.....	107
7.2 ARQUITECTURA PLATAFORMA VMS.....	108
8. PRUEBAS Y RESULTADOS.....	110
8.1 REPORTES.....	113
10. RECOMENDACIONES.....	119
11. BIBLIOGRAFIA.....	120
11.1 REFERENCIAS BIBLIOGRÁFICAS	120

LISTA DE TABLAS

Pág.

<i>Tabla 1 Descripción metodológica por objetivos</i>	51
<i>Tabla 2 Actividades</i>	53
<i>Tabla 3 Objetivo No. 1</i>	62
<i>Tabla 4 Objetivo No. 2</i>	62
<i>Tabla 5 Objetivo No. 3</i>	62
<i>Tabla 6 Funcionalidad No. 1</i>	63
<i>Tabla 7 Funcionalidad No. 2</i>	63
<i>Tabla 8 Objetivo No. 4</i>	64
<i>Tabla 9 Objetivo No. 5</i>	64
<i>Tabla 10 Objetivo No. 6</i>	64
<i>Tabla 11 Objetivo No. 7</i>	65
<i>Tabla 12 Funcionalidad No. 3</i>	65
<i>Tabla 13 Funcionalidad No. 4</i>	65
<i>Tabla 14 Funcionalidad No. 5</i>	66
<i>Tabla 15 Objetivo No. 8</i>	66
<i>Tabla 16 Objetivo No. 9</i>	66
<i>Tabla 17 Objetivo No. 10</i>	67
<i>Tabla 18 Funcionalidad No. 6</i>	67
<i>Tabla 19 Funcionalidad No. 7</i>	67
<i>Tabla 20 Requerimiento no funcional No. 1</i>	68
<i>Tabla 21 Requerimiento no funcional No. 2</i>	68
<i>Tabla 22 Requerimiento no funcional No. 3</i>	68
<i>Tabla 23 Actor – Usuario Anónimo</i>	69
<i>Tabla 24 Actor – Administrador</i>	69
<i>Tabla 25 Actor – Operador</i>	69
<i>Tabla 26 Actor – Usuario Registrado</i>	69
<i>Tabla 27 Velocidades promedio esperadas</i>	112

LISTA DE FIGURAS

	Pág.
<i>Ilustración 1</i> Arquitectura general del sistema	20
<i>Ilustración 2</i> Estructura Datawarehouse Fuente http://www.wikipedia.org/wiki/Base_de_Datos	32
<i>Ilustración 3</i> Business Intelligence Fuente http://www.wikipedia.org/wiki/Business_Intelligence	35
<i>Ilustración 4</i> Ciclo BI	36
<i>Ilustración 5</i> Métodos de desarrollo de software	41
<i>Ilustración 6</i> Microsoft Visual Studio 2010	47
<i>Ilustración 7</i> Microsoft SQL Server 2008 R2 Fuente http://www.wikipedia.org/wiki/SQL_server_2008	49
<i>Ilustración 8</i> Desarrollo	59
<i>Ilustración 9</i> Home Page Opción Universidad de Nariño	60
<i>Ilustración 10</i> Home Page Opción VMS.....	61
<i>Ilustración 11</i> Home Page Opción About VMS Team	61
<i>Ilustración 12</i> Diagrama de clase GetBrandsTable.....	70
<i>Ilustración 13</i> Diagrama de clase GetInsuranceCompanies	70
<i>Ilustración 14</i> Diagrama de clase GetUsersforVehicles	71
<i>Ilustración 15</i> Diagrama de clase ProfileCommon	71
<i>Ilustración 16</i> Diagrama de clase Utilities	71
<i>Ilustración 17</i> Diagrama de clase ProfileGroupAddress.....	72
<i>Ilustración 18</i> Diagrama de clase GetColorsTable.....	72
<i>Ilustración 19</i> Diagrama de clase GetLinesTable.....	73
<i>Ilustración 21</i> Diagrama de clase GetUsers.....	73
<i>Ilustración 20</i> Diagrama de clase VMS	74
<i>Ilustración 22</i> Diagrama de clase GetGPSByPlates	75
<i>Ilustración 23</i> Diagrama de clase ProfileGroupIdentifier	75
<i>Ilustración 24</i> Diagrama de clase ProfileGroupContact	75
<i>Ilustración 25</i> Diagrama de clase ProfileGroupMisc	76
<i>Ilustración 26</i> Diagrama de clase GetGPS	76
<i>Ilustración 27</i> Diagrama de clase GetRoles.....	76
<i>Ilustración 28</i> Diagrama de clase GetVehicleByOPerator.....	77
<i>Ilustración 29</i> Diagrama de clase GetTransportCompany	77
<i>Ilustración 30</i> Diagrama de clase GetGPSVehicles	78
<i>Ilustración 31</i> Diagrama de clase VMSOperations.....	78
<i>Ilustración 32</i> Diagrama de clase GetVehiclesTable.....	79
<i>Ilustración 33</i> Diagrama de clase GetVehicleByUser.....	79
<i>Ilustración 34</i> Diagrama de clase GetOperators	80

<i>Ilustración 35 Diagrama de clase MapWS</i>	80
<i>Ilustración 36 Diagrama de clases de las tablas de la base de datos VMS</i>	82
<i>Ilustración 37 Diagrama de clase Servicio VMS</i>	83
<i>Ilustración 38 Data Warehouse del sistema de monitoreo vehicular en tiempo real</i>	84
<i>Ilustración 39 Estructura del Data Warehouse</i>	85
<i>Ilustración 40 Estructura general proyecto inteligencia de negocios</i>	86
<i>Ilustración 41 Estructura obtenida</i>	86
<i>Ilustración 42 Origen de datos</i>	87
<i>Ilustración 43 Cubo multidimensional</i>	88
<i>Ilustración 44 Estructura de minería - Algoritmo Microsoft Time Series</i>	88
<i>Ilustración 45 Estructura de minería - Algoritmo Microsoft Association Rules</i>	89
<i>Ilustración 46 Módulo de Acceso para usuarios</i>	91
<i>Ilustración 47 Módulo VMS</i>	91
<i>Ilustración 48 Sección News</i>	92
<i>Ilustración 49 Opciones de la sección News</i>	92
<i>Ilustración 50 Sección What's VMS</i>	93
<i>Ilustración 51 Sección Research Objectives</i>	93
<i>Ilustración 52 Aplicación VMS</i>	94
<i>Ilustración 53 Pestaña Home – Botones de Control</i>	95
<i>Ilustración 54 Pestaña Bus Owner</i>	95
<i>Ilustración 55 Pestaña Enterprise</i>	96
<i>Ilustración 56 Mapa</i>	96
<i>Ilustración 57 Propiedades del mapa – Barra de opciones</i>	97
<i>Ilustración 58 Controles de información</i>	97
<i>Ilustración 59 Modulo Managment</i>	98
<i>Ilustración 60 Sección Vehicles</i>	99
<i>Ilustración 61 Formulario New Vehicle</i>	99
<i>Ilustración 62 Sección GPS</i>	100
<i>Ilustración 63 Formulario New GPS</i>	100
<i>Ilustración 64 Sección Assign GPS</i>	101
<i>Ilustración 65 Formulario New GPS Assignment</i>	101
<i>Ilustración 66 Sección Insurace Company</i>	102
<i>Ilustración 67 Formulario New Insurance Company</i>	102
<i>Ilustración 68 Formulario New Transport Company</i>	103
<i>Ilustración 69 Formulario New Users</i>	104
<i>Ilustración 70 Sección Assignment - By Operator</i>	105
<i>Ilustración 71 Formulario New Vehicles by Operator</i>	105
<i>Ilustración 72 Sección Vehicles by User</i>	106
<i>Ilustración 73 Formulario New Vehicle by Operator</i>	106
<i>Ilustración 74 Plataforma</i>	107
<i>Ilustración 75 Arquitectura Plataforma VMS</i>	108
<i>Ilustración 76 Resultados algoritmo de minería Microsoft Association Rules</i>	110
<i>Ilustración 77 Trazo de poligonos (Geocerca) sobre Google Maps - Downtown</i>	110

<i>Ilustración 78 Trazo de poligonos (Geocerca) sobre Google Maps – Panamericana</i>	111
<i>Ilustración 79 Resultados algoritmo de minería Microsoft Time Series.....</i>	112
<i>Ilustración 80 – Promedios de velocidad por día y bus.....</i>	113
<i>Ilustración 81 – Gráfico de barras de los promedios de velocidad por día de cada bus.....</i>	114
<i>Ilustración 82 – Reporte gráfico de la velocidad máxima, velocidad promedio y velocidad mínima del bus de placas SDP 587.....</i>	114

GLOSARIO

GPS: Sistema de Posicionamiento Global. Sistema que permite determinar en todo el mundo la posición de un objeto

NAVEGADOR SATELITAL: Sistema que obtiene información de una constelación de satélites que transmite rangos de señales utilizados para el posicionamiento y localización en cualquier parte del globo terrestre, ya sea en tierra, mar o aire.

MINERIA DE DATOS: Consiste en la extracción no trivial de información que reside de manera implícita en los datos. Dicha información era previamente desconocida y podrá resultar útil para algún proceso. En otras palabras, la minería de datos prepara, sondea y explora los datos para sacar la información oculta en ellos.

DATA WAREHOUSE: En el contexto de la informática, un almacén de datos (del inglés *data warehouse*) es una colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza.

WEB 2.0: El término Web 2.0 (2004–presente) está comúnmente asociado con un fenómeno social, basado en la interacción que se logra a partir de diferentes aplicaciones en la web, que facilitan el compartir información, la interoperabilidad, el diseño centrado en el usuario o D.C.U. y la colaboración en la World Wide Web.

BUSINESS INTELLIGENCE: Se denomina inteligencia empresarial, inteligencia de negocios o BI (del inglés *business intelligence*) al conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa.

MASHUP: Una aplicación web híbrida (mashup o remezcla) es un sitio web o aplicación web que usa contenido de otras aplicaciones Web para crear un nuevo contenido completo, consumiendo servicios directamente, siempre a través de protocolo http.

RESÚMEN

La plataforma web describe un sistema orientado a la localización de vehículos de transporte público en tiempo real de acuerdo a la intercomunicación entre un sistema de posicionamiento GPS, una interfaz electrónica de captura de datos, y una plataforma web que permite consumir los servicios de la base de datos, visualizar los diferentes componentes implementados como consecuencia del desarrollo y así ofrecer una herramienta web completa de ubicación.

La plataforma web permite acceder a los datos con el fin de presentar información importante para la toma de decisiones, controlar de tráfico y optimizar de rutas y tiempos del transporte urbano de la ciudad de Pasto.

La plataforma web obtiene datos específicos como control de pasajeros, consumo de combustible y todo lo referente a la línea de vida de una ruta común, así como todos los eventos que genere el vehículo.

La plataforma web está pensado en la optimización del tráfico y con ello en la reducción de de gases tóxicos y polución, adoptando así una meta del milenio propuesta por las Naciones Unidas.

La plataforma web facilita la recolección de datos de tráfico de forma exacta, evitando el error humano. De esta manera, las conclusiones sobre el tráfico en particular, son más precisas y certeras ya que se manejan sistemas de información para la toma de decisiones.

ABSTRACT

The web platform describes a system oriented to the localization of public transport vehicles in real time through intercommunication between a GPS system, an electronic data-catching interface, and a web application that allows consuming the Database services in order to bring to the users a full-location web tool.

The web platform allows the data access to estimate relevant information to help in the making decisions process, in the traffic control and in the optimization of routes and times in the public urban transport in the city of Pasto.

The web platform is designed to optimize traffic and thus the reduction of toxic gases and pollution, adopting a millennium development goal proposed by the United Nations.

The web platform makes easier the accurately recollection of traffic data, avoiding human error. This way, the findings on vehicular traffic, are more precise and accurate as they manage information systems for the making decisions process.

INTRODUCCIÓN

Los sistemas de posicionamiento global o mejor conocidos como GPS, permiten la ubicación de un objeto en cualquier parte del planeta donde los satélites posean cobertura. Esto brinda oportunidades y posibilidades enormes si de tratamiento de información refiere.

Las aplicaciones web van en aumento exponencial y representan una alternativa eficiente a la hora de establecer mecanismos y soluciones a diferentes problemas, como es el caso del tráfico vehicular.

La optimización del flujo de vehículos en las ciudades ha constituido un problema a resolver en corto plazo. La temática ambiental constituye un elemento importante si hablamos de las nuevas políticas del milenio.

El sistema de transporte requiere una parametrización y cuantificación de sus rutas, con el fin de poder realizar un diagnóstico en tiempos y demás que sean capaces de pronosticar y mejorar el servicio que actualmente las compañías de transporte urbano prestan a los usuarios.

De esta forma nace la idea de implementar un sistema que permita administrar diversos aspectos que son producto de los eventos propios de cada bus, específicamente de la ciudad de Pasto. Esto permitirá obtener datos importantes para la toma de decisiones que incidan en un mejor análisis a la hora de optimizar las rutas.

Hoy en día se puede observar que un buen análisis del tráfico mejora directamente las condiciones ambientales ya que se reducen las emanaciones de gases nocivos en el medio.

“El proceso de brindar soporte a la toma de decisiones a nivel estratégico ha sido reconocido por muchos investigadores, como un tema clave de investigación, y por muchas compañías empresariales como una importante área dentro de la inteligencia de negocios”¹.

¹ Fernández Carlos Molina, Imprecisión e Incertidumbre en el Modelo de Datos Multidimensional: Aplicación a la Minería de Datos, Editorial Universidad de Granada, Granada, España, 2005.

“La inteligencia de negocios (BI) representa las herramientas y sistemas que juegan un papel clave en el proceso estratégico de la planificación de una compañía. Estos sistemas permiten reunir, almacenar, y analizar los datos corporativos siendo una importante ayuda en la toma de decisiones”² .

Para distintas aplicaciones viales (diseño geométrico, programación, semaforización, estudio de, intersecciones, etc.) no solamente se necesita conocer o determinar los volúmenes de tránsito, si no las características del mismo, como velocidades, tiempos de detención o demora, etc.

Hasta no hace mucho tiempo, y aún en la actualidad, para determinar estos parámetros se utilizaba el llamado Método del Vehículo Piloto que básicamente consiste en disponer de un vehículo equipado con odómetro y dos o tres operadores con cronómetros los que toman los tiempos de detención y recorrido en los tramos parciales previamente determinados, consignando en una planilla los tiempos de cada progresiva. Detenciones, tiempos de marcha, etc. Luego, en Gabinete, se calculan los tiempos totales y parciales de detención, las velocidades medias de cada tramo, etc.

Además de la imprecisión propia de ese sistema y el error humano, este método difícilmente pueda determinar con exactitud ciertos parámetros como por ejemplo la velocidad instantánea en cada progresiva. Por otro lado, no solamente resulta de un elevado costo operativo en el terreno, sino que exige un posterior esfuerzo de cálculo en Gabinete.

Los Navegadores Satelitales o GPS (Sistema de Posicionamiento Global) son instrumentos portátiles que tienen la función básica de captar señales satelitales para determinar su ubicación geográfica en tiempo real, almacenando y procesando automáticamente la información relevada en el campo, pudiendo a su vez incorporar en la memoria un Sistema de Información Geográfica (GIS) que permite visualizar la posición en un mapa inteligente.

El Navegador Satelital resulta imprescindible para el relevamiento de los parámetros espacio-tiempo-velocidad-altura de cualquier recorrido, con una precisión imposible de obtener en forma manual. Cuando el GPS en modo navegador recorre un trayecto cualquiera en un móvil, va mostrando a tiempo real y acumulando automáticamente en la memoria, todos los parámetros de velocidad

² Zvenger Patricia Andrea, Introducción al Soporte de Decisiones: Incorporación de soluciones OLAP en entornos empresariales, Universidad Nacional del Sur, 2005.

máxima, velocidad media de movimiento y media total, tiempos de detención y de marcha, tiempo estimado de arribo, distancia hasta el punto de destino, altimetría y distancia recorrida, incluido el replanteo del mismo, como un track.

Este tipo de sistemas pueden ayudar a obtener información que permita mejorar las condiciones del tráfico, en este caso, del transporte urbano en una ciudad.

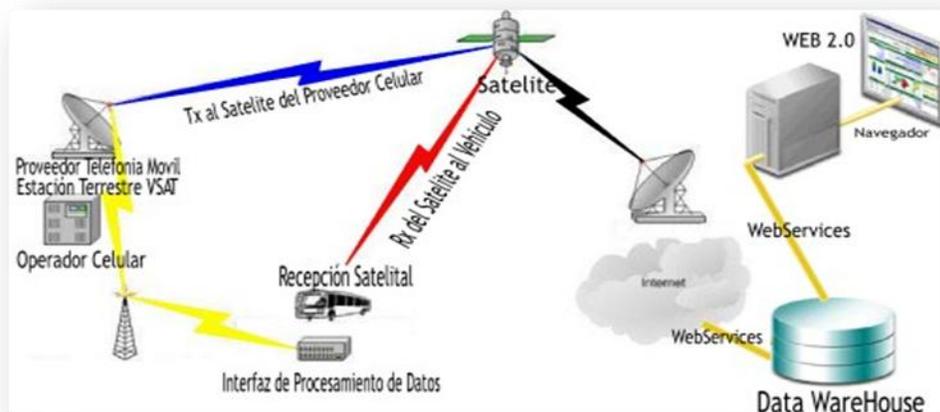
El poder capturar los datos a través de un GPS asegura que los datos sean más confiables a la hora de ser transferidos y procesados por la plataforma Web.

Teniendo en cuenta lo anterior, se propone este proyecto que tiene como objetivo principal la implementación de un aplicativo web que contribuya a la administración de las rutas del Sistema Integrado de Transporte Público de la ciudad de Pasto a través de análisis de rutas y toma de decisiones.

TEMA

El proyecto se enmarca como parte al Trabajo de Investigación Docente “**SISTEMA DE MONITOREO Y CONTROL DE MOVILIDAD VEHICULAR EN TIEMPO REAL**”, dirigido por el Ingeniero JESUS INSUASTY, registrado en el Sistema de Investigaciones de la Universidad de Nariño VIPRI y se encuentra bajo la línea de **software y manejo de información**.

Ilustración 1 Arquitectura general del sistema



La plataforma será una aplicación web 2.0 que implementará una interfaz dinámica y gráfica bajo los privilegios tanto de administradores como de usuarios para visualizar en tiempo real las trayectorias de determinada cantidad de buses de transporte público urbano en la ciudad de Pasto. Así mismo, podrán visualizarse reportes generales y específicos de velocidades, promedios, tiempos, cantidad de pasajeros en determinado momento y lugar apoyados en tecnologías de posicionamiento global como Google Maps™.

Durante el análisis, se hizo necesario implementar una interfaz web que integre el hardware con las diferentes aplicaciones de toma de datos. Las herramientas de visualización permiten controlar de manera eficiente los eventos en tiempo real, posibilitando una adecuada interpretación de la información.

La plataforma web integra diferentes funcionalidades:

La construcción de una plataforma web que administre:

- Usuarios
- Vehículos
- Rutas
- Estaciones

La construcción de servicios web que integre diferentes aplicaciones como:

- **Posición del Vehículo:** Identificar la ubicación precisa del vehículo sobre una ruta específica.
- **Análisis de Rutas:** Identificar la maya vial utilizada por cada uno de los buses de la ciudad de Pasto.
- **Control del Tráfico:** Establecer los tiempos de congestión más críticos en diferentes horas del día.

- **Control de Pasajeros:** Analizar en que momentos el flujo de pasajeros aumenta para optimizar el servicio

La construcción de una base de datos relacional que integre herramientas de análisis como:

- Data Mining (Minería de Datos)
- Data Warehouse (Repositorio de Datos)

La consulta utilizando una herramienta de búsqueda que permita ubicar información precisa y certera sobre los eventos registrados por el hardware instalado en cada vehículo.

El sistema sirve para cuatro propósitos generales:

- Un sistema de registro que permita organizar eficientemente los componentes tales como buses, rutas y usuarios.
- El sistema permite restringir el acceso a las personas u usuarios que no hayan sido registrados como administradores.
- Solamente los administradores pueden realizar cambios en el sistema y en la base de datos. Los usuarios comunes pueden ingresar a la página con el fin de aprovechar los servicios del sistema.
- Las diferentes consultas permiten el acceso a la información proveniente del análisis de los datos obtenidos por las GPS, haciendo uso de una herramienta de búsqueda.

El sistema se encuentra implementado sobre una plataforma web que integra los diferentes servicios utilizando para ello el potencial de un robusto sistema gestor de bases de datos. Esto permite que la información sea concisa, clara y certera para cualquier nivel de consulta.

DESCRIPCIÓN DEL PROBLEMA

El problema objeto de estudio del presente proyecto fue el fenómeno de movilidad vehicular. Este problema es generado principalmente por la carencia de información relevante sobre el comportamiento de dicho fenómeno, lo que implica que los procesos de toma de decisiones no hayan podido dar solución final a tal situación. La planeación eficiente y eficaz es requerida para enfrentar el problema en la actualidad y su proyección sostenible y escalable a futuro, pero dicha planeación requiere de información previa que otorgue un grado de conocimiento más profundo sobre la naturaleza del fenómeno.

En términos generales se puede hablar de múltiples problemas que se presentan alrededor del caos en la movilidad vehicular en las localidades urbanas. Tal fenómeno se presenta tanto en ciudades de menor tamaño como aquellas denominadas metrópolis; en este sentido, la ciudad de Pasto no es ajena a dicha problemática.

El caos vehicular no solo es exclusivo de zonas urbanas, las carreteras y autopistas sufren de esta problemática ya que se constituyen como conectores entre las zonas urbanas, si se pretende trazar a nivel global el flujo vehicular, se estaría tratando de representar un fenómeno con un nivel de complejidad equivalente a los destellos electroquímicos de las neuronas del cerebro humano.

Es preciso entender que la movilidad vehicular adopta comportamientos muy complejos, algunos afirman que dichos comportamientos son comparables a los de un organismo vivo. Teniendo en cuenta la magnitud del problema del caos vehicular en zonas urbanas, es evidente que de él se desprenden un gran número de situaciones que alteran la calidad de vida de los ciudadanos, como consecuencia se genera:

- Embotellamientos y congestión de vías
- Contaminación (Ruido + Polución)
- Guerra económica en los servicios de transporte público
- Inseguridad y accidentalidad

El problema objeto de estudio de esta investigación se centra en el monitoreo y control de la movilidad vehicular. No obstante, de él se desprende un conjunto de problemas derivados dependiendo del contexto donde el fenómeno se presente.

Las causas que originan el problema de movilidad vehicular alrededor del planeta son bastante comunes, el crecimiento desmedido del parque automotor, deficiencias en la infraestructura para soportarlo y la falta de información relevante que permita procesos de planeación eficaz y eficiente.

Es importante resaltar como principal consecuencia del problema del caos vehicular a la congestión, y derivado de ella, el fenómeno de emisión desmedida de gases contaminantes de los vehículos. Este fenómeno está comprobado científicamente que se relaciona en forma directa con el aceleramiento del calentamiento global, situación que tiene repercusiones gigantescas en todo el mundo.

Las Naciones Unidas, a través de su programa de Desarrollo del Milenio, han trazado los 8 objetivos globales de desarrollo para el 2015, y el penúltimo objetivo que dice textualmente “Asegurar la sostenibilidad del ambiente” indica la prioridad de combatir la emisión exagerada de gases de efecto invernadero. Además, la última meta que habla sobre fomentar una asociación mundial para el desarrollo, permite evidenciar que los sistemas de control e información globales son elementos claves para el logro de los objetivos planteados, los cuales deberán ser enriquecidos con información real y confiable.

La relación existente entre la movilidad vehicular y la emisión de gases es muy estrecha, los vehículos emiten mayor cantidad de gases contaminantes en situaciones de congestión vehicular, puesto que emplean marchas lentas y movimientos a corta distancia, además del desgaste en el funcionamiento interno. Por otro lado, los embotellamientos provocan un desorden general en el desarrollo de las actividades cotidianas de los ciudadanos, en realidad se trata de una problemática social como consecuencia de las irregularidades en el flujo normal de los vehículos; las personas llegan tarde a sus destinos, se incrementa la inseguridad en las vías y crece enormemente los niveles de estrés que son causa de múltiples enfermedades.

En conclusión, el caos en la movilidad vehicular afecta seriamente el desarrollo normal de la vida ciudadana, es evidente en todas las localidades urbanas del mundo y es un fenómeno de enorme complejidad.

FORMULACIÓN DEL PROBLEMA

¿El desarrollo de una aplicación web 2.0 para soporte a la toma de decisiones en base a los datos obtenidos por una GPS facilitará el acceso al conocimiento del comportamiento de los buses de transporte público y la consecuente toma de decisiones por parte de las instituciones de tránsito para mejorar las condiciones de tráfico del transporte urbano en la ciudad de Pasto?

SISTEMATIZACIÓN DEL PROBLEMA

- 1 ¿Existen herramientas de posicionamiento GPS/GPRS que ayuden a diagnosticar las condiciones del tráfico en ciudades como Pasto?
- 2 ¿Cómo implementar una aplicación Web 2.0 en un modelo no instanciado en la ciudad de Pasto que sistematizará algunos procesos para recopilación y depuración de información?
- 3 ¿Cómo implementar una aplicación Web 2.0 que permita la visualización y manipulación de información en tiempo real?
- 4 ¿Cómo realizar la comunicación entre la aplicación Web y la GPS?
- 5 ¿Cómo implementar una adecuada conexión entre el Data Warehouse y la aplicación Web con el fin de que los servicios que se prestan sean óptimos?
- 6 ¿Qué conjunto de datos es el más adecuado para realizar las respectivas pruebas del software?
- 7 ¿Cómo se debe realizar el diseño de la interfaz gráfica con el fin de que sea amigable para el usuario final?
- 8 ¿De qué forma se puede implementar la aplicación, de tal manera que sus servicios puedan ser consumidos por otros dispositivos?
- 9 ¿Cómo realizar las pruebas de la aplicación Web 2.0?

OBJETIVOS

Objetivo General

Diseñar e Implementar el Web Site con tecnología Web 2.0 como aplicación integradora de última generación mediante la técnica de Mashup de servicios internacionales (posicionamiento global en tiempo real), que permita obtener datos de una GPS instalada en varios buses de transporte urbano de servicio público con el fin de generar reportes que permitan la toma de decisiones en instituciones de tránsito en la ciudad de Pasto.

Objetivos Específicos

- Analizar, diseñar y desarrollar un módulo que permita el modelamiento multidimensional del modelo relacional con el fin de manipular de manera adecuada los datos.
- Analizar, diseñar y desarrollar un módulo que permita consumir los servicios de posicionamiento global tales como Google Maps™ y puedan ser usados por la aplicación.
- Analizar, diseñar y desarrollar un módulo que permita visualizar y manipular de manera gráfica los datos, con el propósito de mostrarle al usuario una información concisa, fácil de asimilar y de manejar.
- Analizar, diseñar y desarrollar un módulo que permita generar reportes de la información contenida en un Data Warehouse, con el objeto de proporcionar portabilidad a la información.
- Integrar todos los módulos desarrollados en una sola aplicación de tipo Web 2.0 con el ánimo de presentar una herramienta completa, funcional y que satisfaga las necesidades del usuario.
- Implementar la comunicación del GPS con SQL Server 2008 con el propósito de extraer los datos con los que trabajará la aplicación Web 2.0.
- Diseñar e Implementar el sistema de Data Warehouse (bodega de datos) para mantener y controlar la información censada en tiempo real así como

la integración y análisis inteligente de la información a través de Business Intelligence.

- Descubrir reglas de comportamiento preliminar del fenómeno de movilidad vehicular del transporte público de buses de la ciudad.
- Obtener conjuntos de datos reales para la realización de las pruebas con la aplicación Web 2.0.
- Realizar las pruebas y analizar los resultados para garantizar la confiabilidad de la información generada por la aplicación Web 2.0.
- Dar a conocer los resultados del proyecto a través de la publicación de un artículo en una revista o conferencia nacional y/o internacional, con el fin de compartir con la comunidad el proyecto realizado.
- Poner en funcionamiento el sistema en ambientes reales y divulgar sus resultados.

JUSTIFICACIÓN

A nivel nacional, el DEPARTAMENTO NACIONAL DE PLANEACION como organismo estatal colombiano, ha trazado las directrices que apuntan a los planes de desarrollo del país. En este orden de ideas, en el plan denominado “VISION COLOMBIA – II CENTENARIO – 2019 CIUDADES AMABLES”, el departamento de gerencia de la Infraestructura manifiesta las estrategias a seguir para el plan de movilidad de las ciudades y frente a esto, se plantea LA POLITICA DE TRANSPORTE PARA LAS CIUDADES, donde su segundo punto dice textualmente “Adoptar soluciones operativas de bajo costo y alto impacto”.

Puntualmente el producto esperado es un sistema integrado con las siguientes características:

- La totalidad de la información disponible se encontrará en el Web Site del proyecto (dentro de la infraestructura de los servidores de la Universidad de Nariño) que integra los datos censados con un tiempo de respuesta de máximo 2 segundos sobre el estado real; este Web Site utiliza tecnología Web 2.0 que

a través de un Mashup con Google Earth™ permitirá mostrar el comportamiento de la movilidad vehicular sobre fotografías satelitales de la ciudad de Pasto.

- El sistema de reportes sobre proyecciones basados en inteligencia de negocios servirá de apoyo a la toma de decisiones frente a la problemática de la movilidad vehicular de la ciudad. Se pretende que la información recolectada en tiempo real alimenten una bodega de datos que, a través del uso de algoritmos predictivos de inteligencia de negocios, permita tener un conocimiento más profundo del fenómeno de movilidad vehicular y de esta forma, sentar las bases sobre reglas de comportamiento del transporte en la ciudad con el fin de brindar información relevante a las autoridades competentes.
- El sistema trabajará inicialmente para una pequeña muestra de buses urbanos, el objetivo es demostrar la factibilidad del uso masivo del sistema de monitoreo de bajo costo. El sistema planteado será ofrecido en primera instancia a las empresas de transporte público (buses) otorgando control y seguridad de sus móviles. Se evidencia el interés que tienen los dueños de dichas empresas para tener un control integrado de su parque automotor puesto que su intención es ser eficientes en la prestación del servicio contrastando con la reducción de gastos de operación.
- La demostración inicial del sistema será socializada ante la Secretaría de Transporte Municipal, así como a la Alcaldía de Pasto y la Gobernación de Nariño, con el fin de buscar apoyo legal y económico para la implementación masiva de los dispositivos móviles cuyo valor de inversión es relativamente bajo y su mantenimiento mensual es mínimo en contraste al servicio que presta. Se trata de masificar la parte del sistema que es móvil puesto que el resto del sistema de recepción de información, análisis y tratamiento de datos ya estaría en funcionamiento en dicho momento.
- La adaptabilidad del sistema completo es bastante alta, esto es, de forma inicial, la ciudad elegida para el montaje del sistema piloto es Pasto, sin embargo, es factible su implementación posterior en cualquier ciudad de Colombia atendiendo las políticas gubernamentales sobre transporte con soluciones de bajo costo y alto impacto y de igual forma en cualquier ciudad del mundo.

- La sistematización de procesos rutinarios y la optimización de los mismos hace posible que la calidad de vida mejore. Esta implementación permitirá comprender y mejorar las condiciones de tráfico actuales.

CAPITULO 1 MARCO TEÓRICO

1.1 SISTEMA DE POSICIONAMIENTO GLOBAL

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, un vehículo o una nave, con una precisión hasta de centímetros (si se utiliza GPS diferencial), aunque lo habitual son unos pocos metros de precisión. Aunque su invención se atribuye al gobierno francés y belga, el sistema fue desarrollado, instalado y actualmente operado por el Departamento de Defensa de los Estados Unidos.

El GPS funciona mediante una red de 32 satélites (28 operativos y 4 de respaldo) en órbita sobre el globo, a 20 200 km, 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 las 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.

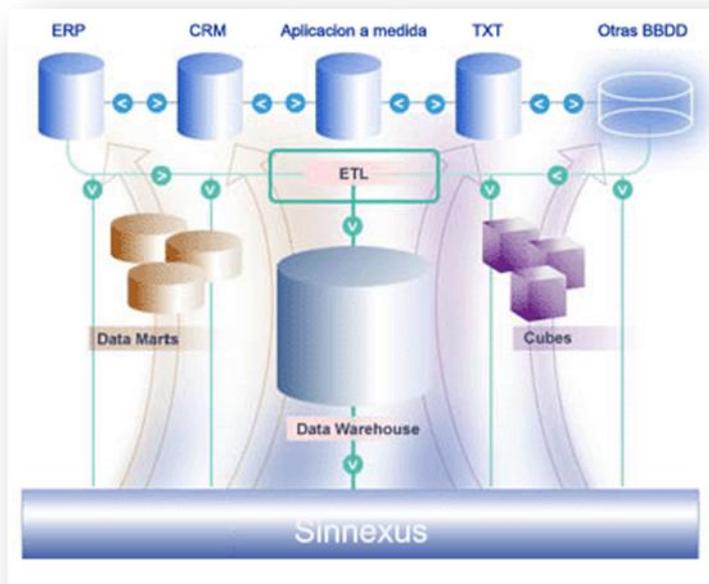
Actualmente la Unión Europea está desarrollando su propio sistema de posicionamiento por satélite, denominado Galileo.

1.2 DATA WAREHOUSE

Un Datawarehouse es una base de datos corporativa que se caracteriza por integrar y depurar información de una o más fuentes distintas, para luego procesarla permitiendo su análisis desde infinidad de perspectivas y con grandes velocidades de respuesta. La creación de un datawarehouse representa en la mayoría de las ocasiones el primer paso, desde el punto de vista técnico, para implantar una solución completa y fiable de Business Intelligence.

La ventaja principal de este tipo de bases de datos radica en las estructuras en las que se almacena la información (modelos de tablas en estrella, en copo de nieve, cubos relacionales... etc). Este tipo de persistencia de la información es homogénea y fiable, y permite la consulta y el tratamiento jerarquizado de la misma (siempre en un entorno diferente a los sistemas operacionales).

Ilustración 2 Estructura Datawarehouse Fuente http://www.wikipedia.org/wiki/Base_de_Datos



El término Datawarehouse fue acuñado por primera vez por Bill Inmon, y se traduce literalmente como *almacén de datos*. No obstante, y como cabe suponer, es mucho más que eso. Según definió el propio Bill Inmon, un datawarehouse se caracteriza por ser:

- **Integrado:** los datos almacenados en el datawarehouse deben integrarse en una estructura consistente, por lo que las inconsistencias existentes entre los diversos sistemas operacionales deben ser eliminadas. La información suele estructurarse también en distintos niveles de detalle para adecuarse a las distintas necesidades de los usuarios.
- **Temático:** sólo los datos necesarios para el proceso de generación del conocimiento del negocio se integran desde el entorno operacional. Los datos se organizan por temas para facilitar su acceso y entendimiento por parte de los usuarios finales. Por ejemplo, todos los datos sobre clientes pueden ser consolidados en una única tabla del datawarehouse. De esta forma, las peticiones de información sobre clientes serán más fáciles de responder dado que toda la información reside en el mismo lugar.
- **Histórico:** el tiempo es parte implícita de la información contenida en un datawarehouse. En los sistemas operacionales, los datos siempre reflejan el estado de la actividad del negocio en el momento presente. Por el contrario, la información almacenada en el datawarehouse sirve, entre otras cosas, para realizar análisis de tendencias. Por lo tanto, el datawarehouse se carga con los distintos valores que toma una variable en el tiempo para permitir comparaciones.
- **No volátil:** el almacén de información de un datawarehouse existe para ser leído, pero no modificado. La información es por tanto permanente, significando la actualización del datawarehouse la incorporación de los últimos valores que tomaron las distintas variables contenidas en él sin ningún tipo de acción sobre lo que ya existía.

Otra característica del datawarehouse es que contiene metadatos, es decir, datos sobre los datos. Los metadatos permiten saber la procedencia de la información, su periodicidad de refresco, su fiabilidad, forma de cálculo... etc.

Los metadatos serán los que permiten simplificar y automatizar la obtención de la información desde los sistemas operacionales a los sistemas informacionales.

Los Datawarehouse contienen datos que son necesarios o útiles para una organización, es decir, que se utiliza como un repositorio de datos para posteriormente transformarlos en información útil para el usuario. Un almacén de datos debe entregar la información correcta a la gente indicada en el momento óptimo y en el formato adecuado. El almacén de datos da respuesta a las necesidades de usuarios expertos, utilizando Sistemas de Soporte a Decisiones (DSS), Sistemas de información ejecutiva (EIS) o herramientas para hacer

consultas o informes. Los usuarios finales pueden hacer fácilmente consultas sobre sus almacenes de datos sin tocar o afectar la operación del sistema.

En el funcionamiento de un almacén de los datos son muy importantes las siguientes ideas:

- Integración de los datos provenientes de bases de datos distribuidas por las diferentes unidades de la organización y que con frecuencia tendrán diferentes estructuras (fuentes heterogéneas). Se debe facilitar una descripción global y un análisis comprensivo de toda la organización en el almacén de datos.
- Separación de los datos usados en operaciones diarias de los datos usados en el almacén de datos para los propósitos de divulgación, de ayuda en la toma de decisiones, para el análisis y para operaciones de control. Ambos tipos de datos no deben coincidir en la misma base de datos, ya que obedecen a objetivos muy distintos y podrían entorpecerse entre sí.

Periódicamente, se importan datos al almacén de datos de los distintos sistemas de planeamiento de recursos de la entidad (ERP) y de otros sistemas de software relacionados con el negocio para la transformación posterior. Es práctica común normalizar los datos antes de combinarlos en el almacén de datos mediante herramientas de extracción, transformación y carga (ETL). Estas herramientas leen los datos primarios (a menudo bases de datos OLTP de un negocio), realizan el proceso de transformación al almacén de datos (filtración, adaptación, cambios de formato, etc.) y escriben en el almacén.

1.3 BUSINESS INTELLIGENCE

Business Intelligence es la habilidad para transformar los datos en información, y la información en conocimiento, de forma que se pueda optimizar el proceso de toma de decisiones en los negocios.

Ilustración 3 Business Intelligence Fuente
http://www.wikipedia.org/wiki/Business_Intelligence



Desde un punto de vista más pragmático, y asociándolo directamente con las tecnologías de la información, se puede definir Business Intelligence como el conjunto de metodologías, aplicaciones y tecnologías que permiten reunir, depurar y transformar datos de los sistemas transaccionales e información desestructurada (interna y externa a la compañía) en información estructurada, para su explotación directa (reporting, análisis OLTP / OLAP, alertas...) o para su análisis y conversión en conocimiento, dando así soporte a la toma de decisiones sobre el negocio.

La inteligencia de negocio actúa como un factor estratégico para una empresa u organización, generando una potencial ventaja competitiva, que no es otra que proporcionar información privilegiada para responder a los problemas de negocio: entrada a nuevos mercados, promociones u ofertas de productos, eliminación de islas de información, control financiero, optimización de costes, planificación de la producción, análisis de perfiles de clientes, rentabilidad de un producto concreto, etc.

Los principales productos de Business Intelligence que existen hoy en día son:

- Cuadros de Mando Integrales (CMI)
- Sistemas de Soporte a la Decisión (DSS)
- Sistemas de Información Ejecutiva (EIS)

Por otro lado, los principales componentes de orígenes de datos en el Business Intelligence que existen en la actualidad son:

- Datamart
- Datawarehouse

Los sistemas y componentes del BI se diferencian de los sistemas operacionales en que están optimizados para preguntar y divulgar sobre datos. Esto significa típicamente que, en un datawarehouse, los datos están desnormalizados para apoyar consultas de alto rendimiento, mientras que en los sistemas operacionales suelen encontrarse normalizados para apoyar operaciones continuas de inserción, modificación y borrado de datos. En este sentido, los procesos ETL (extracción, transformación y carga), que nutren los sistemas BI, tienen que traducir de uno o varios sistemas operacionales normalizados e independientes a un único sistema desnormalizado, cuyos datos estén completamente integrados.

En definitiva, una solución BI completa permite:

- Observar ¿Qué está ocurriendo?
- Comprender ¿Por qué ocurre?
- Predecir ¿Qué ocurrirá?
- Colaborar ¿Qué debería hacer el equipo?
- Decidir ¿Qué camino se debe seguir?

Ilustración 4 Ciclo BI



1.4 MASHUP

Una aplicación web híbrida (mashup o remezcla) es un sitio web o aplicación web que usa contenido de otras aplicaciones Web para crear un nuevo contenido completo, consumiendo servicios directamente, siempre a través de protocolo http.

El contenido de un mashup normalmente proviene de sitios web de terceros a través de una interfaz pública o usando un API. Otros métodos que constituyen el

origen de sus datos incluyen sindicadores web (RSS o Atom), Screen scraping, etc.

Los mashups están revolucionando el desarrollo web del mismo modo que los weblogs han revolucionado la publicación en línea.

Los mashups permiten que cualquiera combine, de forma innovadora, datos que existen en diferentes páginas web. Requieren pocos conocimientos técnicos, las APIs existentes son sencillas y potentes y los mashups son relativamente fáciles de diseñar.

Los creadores de mashups son generalmente gente innovadora que desea combinar de formas nuevas y creativas datos disponibles públicamente.

Así como hay mashups muy útiles, existen otros que no pasan de ser sólo novedosos o publicitarios, con mínima utilidad práctica.

Los defensores e impulsores de las aplicaciones Web 2.0 afirman que los mashups son un ejemplo de este nuevo movimiento en el que los usuarios crean, participan e interactúan activamente.

El contenido usado en mashups es típicamente obtenido de otra fuente vía una interface pública o API (web services), aunque existe gente en la comunidad que considera que los casos en que las interfaces son privadas no deberían contar como mashups. Otros métodos de obtener contenido para mashups incluyen Web Feeds (i.e RSS o Atom) y screen scraping.

Mucha gente experimenta con mashups usando las APIs de Amazon, eBay, Flickr, Google, Microsoft, Yahoo o YouTube lo que ha llevado a la creación de un editor mashup.

La arquitectura de los mashups está siempre compuesta de tres partes:

- El proveedor de contenidos: fuente de los datos. Los datos están disponibles vía una API y diferentes protocolos web como RSS, REST y Web Service.
- El sitio mashup: es la nueva aplicación web que provee un nuevo servicio utilizando diferente información y de la que no es dueña.

- El web browser cliente: es la interface de usuario del mashup. En una aplicación web, el contenido puede ser mezclado por los web browser clientes usando lenguaje web del lado del cliente. Por ejemplo javascript.

Los mashups deben ser diferenciados de simples embebidos de datos de otro sitio para formar un documento compuesto. Un sitio que permite al usuario embeber vídeos de youtube, por ejemplo, no es un sitio mashup. Como ya se dijo , el sitio mismo debe acceder información externa a él usando una API y procesar esos datos de modo de incrementar su valor para el usuario.

1.4.1 Tipos de Mashups. Los mashups se presentan actualmente en tres formas: mashups de consumidores, mashups de datos y mashups empresariales.

El tipo más conocido es el de mashup de consumidores, que está muy bien ejemplificado por muchas aplicaciones que utilizan Google Maps. Los mashups de este tipo combinan datos de fuentes varias, escondiendo ello tras una interface gráfica simple.

Un mashup de datos mezcla datos de tipo similar proveniente de diferentes fuentes (ver Yahoo Pipes, <http://pipes.yahoo.com/pipes/>). Por ejemplo, combinando los datos de múltiples feeds RSS en un solo feed con nuevo un front-end gráfico.

Un mashup empresarial (ejemplo, JackBe, <http://www.jackbe.com>) integra usualmente datos de fuentes externas e internas. Por ejemplo, podría crear un reporte sobre la cuota de mercado de un negocio combinando la lista externa de todas las casas vendidas la semana anterior con datos internos de las casas vendidas por una sola agencia.

Un mashup de negocio es una combinación de todo lo anterior, enfocando en agregación de datos y presentación y agregando adicionalmente una funcionalidad colaborativa, haciendo que el resultado final sea una aplicación de negocio apropiada.

Mashups dentro de mashups son conocidos como “mashups monstruos”.

Es importante reconocer que los mashups ayudan o facilitan la integración de aplicaciones orientadas a arquitecturas SOA.

1.4.2 Ejemplos. Los mashups pueden ser divididos en cuatros grandes categorías: "mapas", "vídeo y fotos", "búsqueda y compras" y "noticias". Algunos ejemplos de cada categoría:

Mapas

Chicago Crime - El departamento de policía de Chicago tiene un mashup (http://gis.chicagopolice.org/CLEARMap_crime_sums/startPage.htm) que integra la base de datos del departamento de crímenes reportados con Google Maps de modo de ayudar a detener crímenes en ciertas áreas y avisar a los ciudadanos de áreas potencialmente más peligrosas.

WikiCrimes - WikiCrimes (<http://www.wikicrimes.org>) es un sitio web tipo wiki donde los usuarios de Internet pueden reportar crímenes pinchando banderas en un mapa basado en Google Maps. El sitio distingue categorías dependiendo del color.

minnus - minnus(<http://www.minnus.com.ar>) minnus es una comunidad virtual donde sus usuarios tienen acceso a información de todo tipo, ya sea cultural, social, histórica, comercial, ambiental, turística, de transito, etc. Dicha información se encuentra geoposicionada en un mapa creado con imágenes de satélite para así formar grupos con mismos intereses.

Vídeo y fotos

Flickr - Flickr es un sitio de almacenamiento de imágenes que permite a los usuarios organizar sus colecciones de imágenes y compartirlas. Utilizando su API el contenido puede ser usado en otros sitios creando mashups. Flickrvision (<http://flickrvision.com>) es un ejemplo.

Búsqueda y compras

Travature - Travature (<http://www.travature.com>) es un portal de viajes que ha integrado motores de meta búsquedas con guías de viajes tipo wiki y reseñas de hoteles. También permite compartir experiencias entre viajeros.

Noticias

Digg - Digg (<http://digg.com/>) es un mashup de varios sitios de noticias controlado casi enteramente por los usuarios del sitio.

BFreeNews.com - BFreeNews (<http://bfree news.com/>) es un mashup de fuentes de noticias de calidad cruzadas con recomendaciones de noticias de Twitter y búsquedas de Google. Muestra las noticias más comentadas en twitter y más indexadas por Google en las últimas 24 horas.

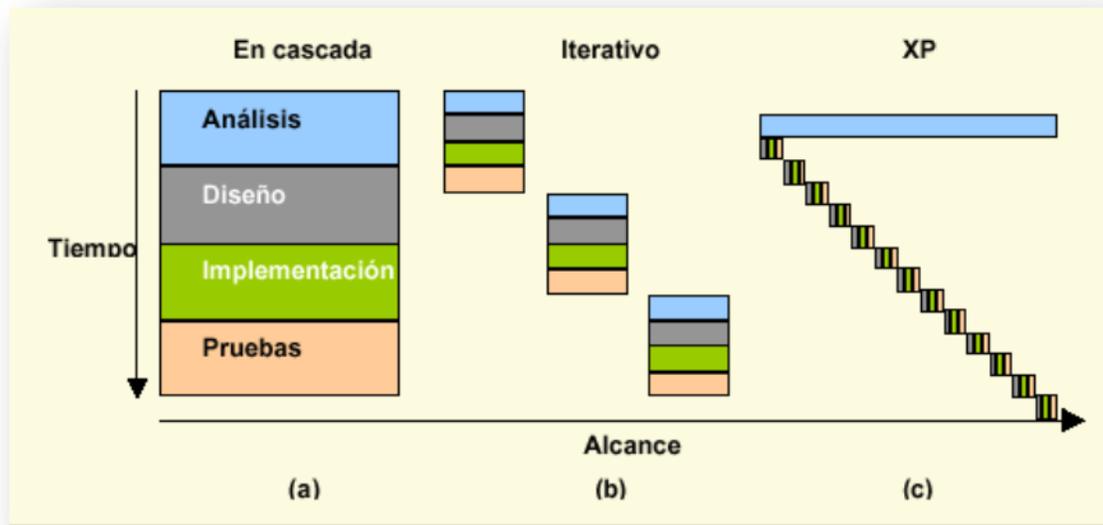
Enciclopedias

Histourist - Histourist (<http://www.histourist.com/>) es un Mashup Semántico que ofrece una Enciclopedia multimedia geolocalizada de lugares históricos. Los artículos se preparan mediante una combinación de editores y robots de software que explotan los recursos on-line en fuentes definidas como "confiables" (BBC, National Geographic, DBpedia, The History Channel, etc.)y en particular los servicios de la web semántica para enriquecer los artículos con videos, fotos, bibliografía, y clasificarlo en las taxonomías del servicio.

1.5 EXTREME PROGRAMMING

XP (*eXtreme Programming*) es una nueva disciplina para el desarrollo de software, que ha irrumpido recientemente con gran revuelo en el maremágnum de métodos, técnicas y metodologías existentes. Concretando más, se trata de una metodología «ligera», en contraposición a las metodologías «pesadas» como Métrica. La Programación Extrema es una metodología de desarrollo de software que se basa en la simplicidad, la comunicación y la retroalimentación o reutilización del código desarrollado (reciclado de código). Cuenta con tan solo 6 años de vida, pero con un gran respaldo por parte de grandes empresas como la **Ford, DaimlerChrysler, First Union National Bank -USA-**, etc., que lo que buscan en definitiva es la reducción de costes.

Ilustración 5 Métodos de desarrollo de software



CAPITULO 2 ANTECEDENTES

Al ser el problema de orden global, existen una gran cantidad de estudios, investigaciones y adaptaciones tecnológicas que han contribuido de alguna manera a contrarrestar los efectos del caos en la movilidad vehicular.

Para comenzar, el problema del transporte ha sido permanente, muchas ideas han sido propuestas y varias medidas han sido implementadas para resolver el problema “eterno” de movilidad en las ciudades. La congestión y ruido se conocen desde la antigua roma, y las restricciones del tráfico tales como las calles de una sola vía y zonas de aparcamiento fueron introducidas en la época de Julio César.

A nivel internacional existen institutos especializados sobre el problema de movilidad vehicular, transporte y tráfico. En Estados Unidos existe el Institute of Transportation Engineers que fue fundado en 1930, es en este país donde la problemática se agudizó exageradamente por la alta producción de vehículos de consumo general. Se crea entonces la carrera profesional sobre la Ingeniería de Transporte para brindar soluciones científicas al problema de tráfico y movilidad vehicular.

El matemático Dietrich Braess, autor célebre de la “Paradoja de Braess” afirma que agregando la capacidad extra a una red (cualquiera que sea), cuando las entidades móviles escogen en forma egoísta y aleatoria su ruta, puede reducir a veces el desempeño general. Esto es porque el equilibrio de tal sistema no es necesariamente óptimo. Por consiguiente, NO ES LA SOLUCION, LA SIMPLE AMPLIACION DE LA RED VIAL para el mejoramiento de la movilidad vehicular. La demostración matemática existe a través del documento “Proceedings of IEEE 2001 Conference on Intelligent Transportation Systems 837-842” tomado de “From Characterizing Braess’s Paradox for Traffic Networks, J. N. Hagstrom and R. A. Abrams”. En esta última investigación se demuestra científicamente que la ampliación de la red vial tiende a EMPEORAR el problema de movilidad vehicular, investigación realizada en la Universidad de Chicago en el 2001.

A nivel nacional existen antecedentes sobre proyectos de enorme inversión que están encaminados a resolver problemas de movilidad urbana, las ciudades de Medellín, Bogotá y Cali, entre otras, son ejemplos de incorporación de sistemas de transporte masivo que si bien han mejorado la movilidad de las personas, no han sido completamente satisfactorios en cuanto a la solución del problema del caos de vehículos.

Según los administradores del sistema de Transmilenio en Bogotá, para solucionar el problema de congestión de los buses sería necesario incrementar la proporción de buses por usuario, lo que implicaría un aumento en el costo de los viajes; se cree que los bogotanos que preferirían pagar más para viajar más cómodos son una minoría. Por otro lado, la densidad de pasajeros por bus es comparable a la de sistemas de transporte masivo en otras ciudades del mundo (e.g. metros de Madrid, Barcelona, París, Roma y Londres).

Probablemente la crítica a los atestados que viajan los buses se debe al contraste con los buses y busetas de servicio público tradicionales. En ellos se presenta una situación de sobreoferta, que trae como consecuencia que viajen vacíos la mayoría del tiempo (aunque en horas punta se pueden llenar bastante).

En términos generales, el problema del caos vehicular es común en casi todas las ciudades del mundo. La disciplina asociada al estudio del fenómeno que es la Ingeniería del Transporte a través de su producción de conocimiento sobre dicho problema ha permitido concluir que el fenómeno del caos vehicular es de naturaleza compleja y que vale la pena tomar datos para entender sus manifestaciones particulares de cada contexto. La experiencia ha enseñado que no es la única solución el montaje de un sistema de transporte masivo o la ampliación de la infraestructura vial. Esta es la esencia de los estudios sobre movilidad sostenible.

CAPITULO 3 HERRAMIENTAS DE DESARROLLO PARA LA CONSTRUCCIÓN DE LA PLATAFORMA WEB 2.0 DEL SISTEMA

3.1 LENGUAJE DE PROGRAMACIÓN

3.1.1 Html. El HTML, Hyper Text Markup Language (Lenguaje de marcación de Hipertexto) es el lenguaje de marcas de texto utilizado normalmente en la www (World Wide Web). Fue creado en 1986 por el físico nuclear Tim Berners-Lee; el cual tomo dos herramientas preexistentes: El concepto de Hipertexto (Conocido también como link o ancla) el cual permite conectar dos elementos entre si y el SGML (Lenguaje Estándar de Marcación General) el cual sirve para colocar etiquetas o marcas en un texto que indique como debe verse. HTML no es propiamente un lenguaje de programación como C++, Visual Basic, etc., sino un sistema de etiquetas. HTML no presenta ningún compilador, por lo tanto algún error de sintaxis que se presente éste no lo detectará y se visualizara en la forma como éste lo entienda.

El entorno para trabajar HTML es simplemente un procesador de texto, como el que ofrecen los sistemas operativos Windows (Bloc de notas), UNIX (el editor vi o ed) o el que ofrece MS Office (Word). El conjunto de etiquetas que se creen, se deben guardar con la extensión .htm o .html.

Estos documentos pueden ser mostrados por los visores o "browsers" de páginas Web en Internet, como Netscape Navigator, Mosaic, Opera y Microsoft Internet Explorer.

También existe el HTML Dinámico (DHTML), que es una mejora de Microsoft de la versión 4.0 de HTML que le permite crear efectos especiales como, por ejemplo, texto que vuela desde la página palabra por palabra o efectos de transición al estilo de anuncio publicitario giratorio entre página y página.

3.1.2 Javascript. JavaScript es un lenguaje de scripting basado en objetos sin tipo y liviano, utilizado para acceder a objetos en aplicaciones. Principalmente, se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas. JavaScript es un dialecto de

ECMAScript y se caracteriza por ser un lenguaje basado en prototipos, con entrada dinámica y con funciones de primera clase. JavaScript ha tenido influencia de múltiples lenguajes y se diseñó con una sintaxis similar al lenguaje de programación Java, aunque más fácil de utilizar para personas que no programan. Todos los navegadores modernos interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

3.1.3 Lenguaje de programación c sharp sobre .net. C# o C Sharp es un lenguaje de programación que está incluido en la Plataforma .NET y corre en el Lenguaje Común en Tiempo de Ejecución (CLR, Common Language Runtime). El primer lenguaje en importancia para el CLR es C#, mucho de lo que soporta la Plataforma .NET está escrito en C#.

C# deriva de C y C++, es moderno, simple y enteramente orientado a objetos, simplifica y moderniza a C++ en las áreas de clases, namespaces, sobrecarga de métodos y manejo de excepciones. Se eliminó la complejidad de C++ para hacerlo más fácil de utilizar y menos propenso a errores.

Algunas características de C# son:

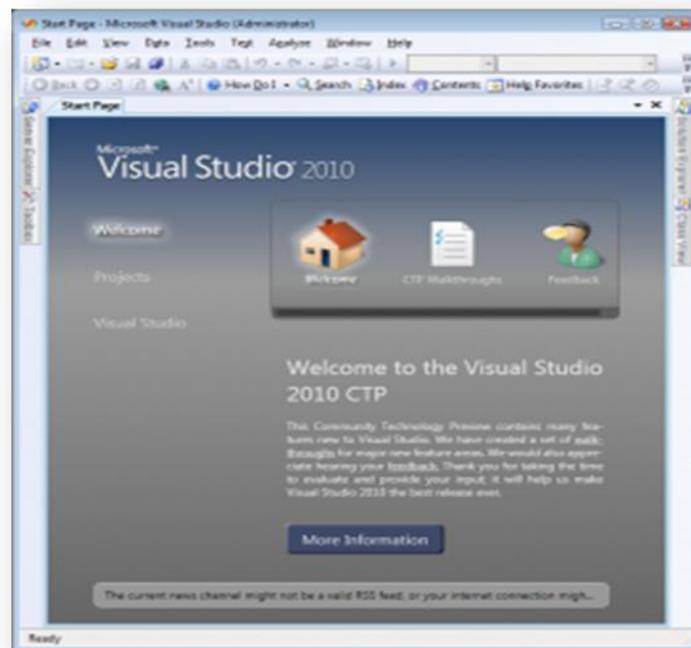
- C# provee el beneficio de un ambiente elegante y unificado.
- No soporta herencia múltiple, solamente el *runtime* .NET permite la herencia múltiple en la forma de interfaces, las cuales no pueden contener implementación.
- No maneja apuntadores, para emular la función de los apuntadores se utiliza delegates el cual provee las bases para el .NET event model.
- Por default trabaja con código administrado.
- La Plataforma .NET provee un colector de basura que es responsable de administrar la memoria en los programas C#.
- El manejo de errores está basado en excepciones.
- Soporta los conceptos como encapsulación, herencia y polimorfismo de la programación orientada a objetos.
- El Modelo completo de clases está construido en la cima del .NET Virtual Object System (VOS). El modelo de objetos es parte de la infraestructura y ya no forma parte del lenguaje de programación.

3.2 ENTORNO DE DESARROLLO

3.2.1 Microsoft visual studio 2010. “Microsoft Visual Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.”

Esta herramienta se incorpora muy bien a las exigencias del proyecto de investigación por su versatilidad, robustez y facilidad a la hora de implementar servicios web.

Ilustración 6 Microsoft Visual Studio 2010



3.2.2 Asp.net. “ASP.NET es un framework para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el CommonLanguageRuntime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

Cualquier persona que este familiarizada con el desarrollo de aplicaciones web sabrá que el desarrollo web no es una tarea simple. Ya que mientras que un modelo de programación para aplicaciones de uso común está muy bien establecido y soportado por un gran número de lenguajes, herramientas de desarrollo, la programación web es una mezcla de varios lenguajes de etiquetas, un gran uso de lenguajes de script y plataformas de servidor. Por desgracia para el programador de nivel intermedio, el conocimiento y habilidades que se necesitan para desarrollar aplicaciones web tienen muy poco en común con las que son necesarias en el desarrollo tradicional de aplicaciones.

3.3 SGBD SISTEMA GESTOR DE BASES DE DATOS

3.3.1 Sql server 2008 r2. Microsoft SQL Server es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. **Microsoft SQL Server** constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son **Oracle** o **MySQL**.”

Características

- Soporte de transacciones.
- Escalabilidad, estabilidad y seguridad.
- Soporta procedimientos almacenados.
- Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

Este sistema incluye una versión reducida, llamada MSDE con el mismo motor de base de datos pero orientado a proyectos más pequeños, que en sus versiones 2005 y 2008 pasa a ser el SQL Express Edition, que se distribuye en forma *gratuita*.

Ilustración 7 Microsoft SQL Server 2008 R2 Fuente http://www.wikipedia.org/wiki/SQL_server_2008



CAPITULO 4 METODOLOGÍA DE ANÁLISIS, DISEÑO Y MODELADO

La metodología de desarrollo que más se adecua a las necesidades del proyecto, es la Programación Extrema (eXtreme Programming – XP), en primer lugar porque esta técnica tiene como principio mejorar las relaciones interpersonales como clave para el desarrollo de la aplicación, promoviendo el trabajo en equipo y el aprendizaje de los desarrolladores, así como también hacer entregas periódicas de la aplicación funcional.

Teniendo en cuenta las fases del proceso XP, la investigación se llevó a cabo de la siguiente forma:

4.1 FASE DE EXPLORACIÓN

Dentro de esta fase se realizó el análisis de varios de los servicios de mapas en la Internet, con el fin de usar el mejor de ellos para el módulo de posicionamiento global de la aplicación. También se determinó el formato con el cual la GPS empaqueta la información recolectada y es enviada a la aplicación.

Al final de esta etapa se determinó el estado actual de las aplicaciones, las falencias y las historias de usuario que indiquen las correcciones que se deben aplicar.

Igualmente en esta fase tendrá lugar el análisis de los nuevos requerimientos que soportará la nueva herramienta y mejorar la interfaz gráfica de usuario.

4.2 ITERACIONES

En esta etapa se realizaron iteraciones entre el análisis, el diseño y su implementación. Dentro del análisis y diseño de la nueva herramienta se planificó mediante diagramas UML la forma en que se desarrolla el software, determinado por los requisitos, producto de esta fase.

4.3 FASE DE PRODUCCIÓN

Esta fase tiene que ver con la implementación de la herramienta a través de trabajo de programación. Dentro de esta fase se hace necesaria la documentación de nuevas características que se prevean para su inclusión e implementación posterior, de ser así hay que regresar a la etapa anterior de análisis y diseño.

Paralelamente a partir de la primera fase del proyecto, o sea la de Exploración se realizó la respectiva documentación mediante UML de todas las actividades realizadas durante el ciclo de vida del proyecto.

El proyecto responde a una investigación inmersa dentro del paradigma empírico-analítico, de tipo cuantitativo y con enfoque experimental. De acuerdo a lo anterior, el presente proyecto integra diseño y desarrollo de componentes hardware y software, por consiguiente es necesario discriminar las metodologías usadas para cada objetivo, así:

Tabla 1 Descripción metodológica por objetivos.

OBJ.ESP.	METODOLOGIA	RECURSOS
1	<ul style="list-style-type: none">• Revisión documental• Análisis de riesgos• Adaptación del estándar internacional	<ul style="list-style-type: none">• GSM/GPRS International standard
2	<ul style="list-style-type: none">• Simulación digital• Montaje de circuitos• Puesta a punto• Recolección de datos a través de sensores, adecuación de señales por medio de hardware y envío de la información vía GSM.	<ul style="list-style-type: none">• MathLAB™ + MULTISIM -> Proteus• Tarjeta programadora de Micro controladores• GPRS Modem + Interfaces Electrónicas + Sensores
3	<ul style="list-style-type: none">• Análisis de datos adquiridos• Aplicación de algoritmos de cifrado	<ul style="list-style-type: none">• Estándares internacionales para empaquetamiento digital de datos

4	<ul style="list-style-type: none"> • Modelación UML (Unified Modeling Language) para el data warehouse • Modelación UML (Unified Modeling Language) para el módulo de Business Intelligence 	<ul style="list-style-type: none"> • UML International standard • UML International standard
5	<ul style="list-style-type: none"> • Ejecución de Algoritmos preestablecidos de Business Intelligence 	<ul style="list-style-type: none"> • Reportes sobre descubrimiento de conocimiento acerca del fenómeno a partir de la información recolectada • Construcción de lineamientos generales para el mejoramiento de la movilidad vehicular.
6	<ul style="list-style-type: none"> • Service-Oriented Architecture • Web Engineering Process 	<ul style="list-style-type: none"> • SOA Specification • Software Engineering Methods (Roger Pressman)
7	<ul style="list-style-type: none"> • Muestreo automatizado • Diseño de Bitácoras de Producto 	<ul style="list-style-type: none"> • Registros de memoria de los datos sensados en tiempo real • Bitácoras de Producto

4.4 TÉCNICAS DE RECOLECCIÓN Y ANÁLISIS DE LOS DATOS

Como lo indica la tabla anterior, por tratarse de un sistema de monitoreo en tiempo real de un fenómeno, se hace necesario la toma de muestro automatizado de ciertas variables importantes a los intereses del proyecto. En este sentido, el instrumento de recolección de datos es el recurso de que se valen los investigadores para acercarse al fenómenos y extraer información de él.

El instrumento trata dos aspectos:

4.4.1 La forma. Se refiere a las técnicas que se utilizan para la tarea de aproximación a la realidad (en este caso se trata de muestro automatizado).

4.4.2 El contenido. Queda expresado en la especificación de los datos que se necesitan adquirir. se concreta en una serie de ítems que no son otra cosa que los indicadores que permiten medir a las variables.

A diferencia de otras investigaciones, los datos son recolectados en registros de memoria de las interfaces electrónicas. En el desarrollo de la investigación se otorga el formato a manera de tablas.

La información sobre tasa de flujo vehicular y volumen del flujo vehicular, así como otras variables complementarias (velocidades, espaciamiento promedio y densidad, intervalos simples, pasajeros, consumo de combustible, emisión de gases, entre otros), fueron censados en tiempo real.

El análisis de los datos recolectados se lo realiza a través de modelos internacionales de aproximación al fenómeno del tráfico vehicular. Teniendo en cuenta que existen varios modelos, el elegido por los investigadores es el modelo patentado por el Doctor Bruce Douglas Greenshields denominado MODELO LINEAL de análisis de flujo vehicular a través del ajuste por mínimos cuadrados de la relación velocidad/densidad vehicular.

El uso de técnicas de análisis de información en el presente proyecto se realiza con algoritmos de Minería de Datos en la parte de Business Intelligence en servidor. Para tal menester se hace necesario la recuperación de grandes volúmenes de información lo cual implica realizar una captura de datos cada 3 segundos en movimiento y 1 minuto cuando el vehículo esta inactivo durante las 24 horas del día.

Tabla 2 Actividades

ACTIVIDADES LÓGICAS ANTERIORES	ACTIVIDADES PLANIFICADAS			ACTIVIDADES LÓGICAS POSTERIORES
	ORDEN	DETALLE	DURACIÓN EN SEMANA	
--	A	Análisis del Estado actual	4	B,C
A	B	Análisis y Diseño de la nueva herramienta	6	D
A	C	Historias de Usuario	6	D
B,C	D	Plan de versiones	4	E,F
D	E	Desarrollo del Aplicativo	24	G

D	F	Pruebas del Aplicativo	24	G
E,F	G	Montaje y Puesta en Marcha Herramienta	6	H
G	H	Contraste de Herramientas	4	I
H	I	Documentación y Artículo	48	--

4.5 IDENTIFICACIÓN Y CARACTERIZACIÓN DE LA INNOVACIÓN PROPUESTA

El proyecto es innovador puesto que:

- No existen antecedentes a nivel nacional sobre movilidad vehicular a través de sistemas satelitales e intercomunicados mediante el uso de la red de telefonía celular.
- Se trata de una solución de trabajo multidisciplinar, de bajo costo y alto impacto donde se requiere moldeamiento matemático, diseño de interfaces electrónicas y diseño de software altamente escalable y con comportamiento inteligente.
- Apunta significativamente a las 2 últimas metas de desarrollo del milenio propuestas por las Naciones Unidas, en cuanto a estrategias de sostenibilidad del ambiente y fomento de desarrollo en comunidad y ambientes colaborativos se refiere.
- Apunta significativamente a la política gubernamental del Departamento Nacional de Planeación frente a los problemas de movilidad urbana, donde se especifica la aprobación y promoción de soluciones operativas de bajo costo y alto impacto.

4.6 LENGUAJE UNIFICADO DE MODELADO

“El Lenguaje de Modelado Unificado (UML:Unified Modeling Language) es la sucesión de una serie de métodos de análisis y diseño orientadas a objetos que aparecen a fines de los 80's y principios de los 90s.UML es llamado un lenguaje de modelado, no un método. Los métodos consisten de ambos de un lenguaje de modelado y de un proceso. El UML , fusiona los conceptos de la orientación a

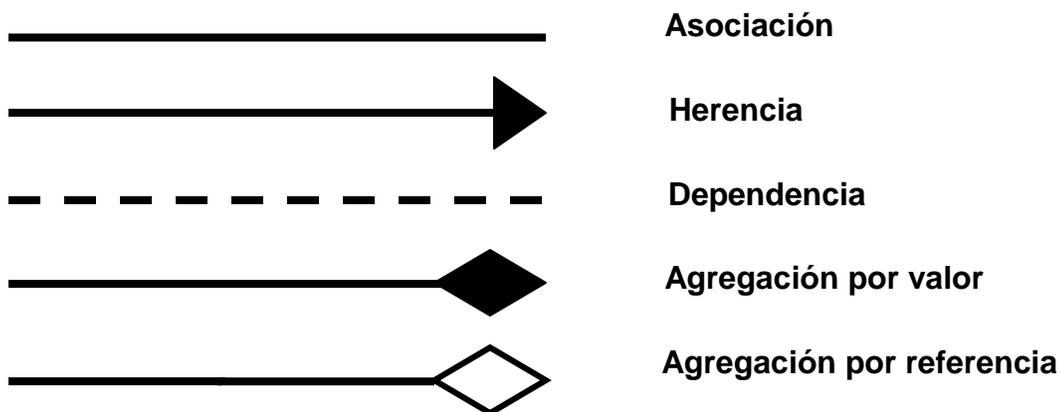
objetos aportados por Booch, OMT y OOSE (Booch, G. et al., 1999). UML incrementa la capacidad de lo que se puede hacer con otros métodos de análisis y diseño orientados a objetos. Los autores de UML apuntaron también al modelado de sistemas distribuidos y concurrentes para asegurar que el lenguaje maneje adecuadamente estos dominios.

El lenguaje de modelado es la notación (principalmente gráfica) que usan los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño.

La estandarización de un lenguaje de modelado es invaluable, ya que es la parte principal del proceso de comunicación que requieren todos los agentes involucrados en un proyecto informático. Si se quiere discutir un diseño con alguien más, ambos deben conocer el lenguaje de modelado y no así el proceso que se siguió para obtenerlo.”

El Lenguaje Unificado de Modelado permite administrar de una forma visual los procesos y procedimientos de los que consta nuestra aplicación. Con este lenguaje se puede definir el flujo de datos e información, así como las actividades y eventos que las transforman. Para ello se utilizan diferentes tipos de conectores, que permiten identificar las relaciones existentes entre las entidades. En últimas se lograría una síntesis completa del funcionamiento real del aplicativo.

4.6.1 Tipos de conectores



Los diferentes tipos de relaciones pueden definirse de la siguiente manera:

- **Asociación.** Esta relación determina que los objetos de una clase se encuentran unidos a los objetos de otra clase.
- **Herencia (Generalización).** Esta relación indica que una clase hija obtiene sus métodos y atributos de una súper clase con la ventaja de que también puede tener métodos y atributos propios.
- **Dependencia.** Es una relación de uso donde una clase depende de otra, es decir, que un cambio en la clase independiente tendrá efecto en la clase dependiente. Una relación de dependencia se presenta cuando una clase utiliza objetos de otra como argumentos de sus operaciones.
- **Agregación.** “Es una especialización de la asociación entre dos clases que indica que los objetos de una clase forman parte de los objetos de la otra, es decir que estructuralmente un objeto contiene a otro.
- **Por valor:** Es un tipo de relación estática, en donde el tiempo de vida del objeto incluido está condicionado por el tiempo de vida del que lo incluye. Este tipo de relación es comúnmente llamada **Composición** (el Objeto base se construye a partir del objeto incluido, es decir, es "parte/todo").
- **Por referencia:** Es un tipo de relación dinámica, en donde el tiempo de vida del objeto incluido es independiente del que lo incluye. Este tipo de relación es comúnmente llamada
- **Agregación** (el objeto base utiliza al incluido para su funcionamiento)”.

4.6.2 Diagrama de casos de uso. Los casos de uso se encargan de definir el comportamiento del sistema o una parte de él según una tarea en específico. Estos revelan las tareas que generan un resultado importante para el agente.

4.6.3 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.”

Las visualizaciones que se pueden hacer con este tipo de diagramas son: requerimientos en entidades y actuaciones, la arquitectura conceptual de un dominio, soluciones de diseño en una arquitectura, componentes de software orientados a objetos

CAPITULO 5 DESARROLLO DE LA PLATAFORMA WEB 2.0

El proyecto define un diseño de una plataforma que permita comunicar la base de datos con el hardware. Para esto se utilizó herramientas licenciadas como Visual Studio 2010 y Sql Server 2008 que se encuentran bajo licencia de campus agreement y que permite implementar la plataforma a nivel educativo.

Estas herramientas son muy potentes a la hora de realizar desarrollos bajo el enfoque XP (eXtreme Programming) ya que se pueden hacer prototipos y pruebas simuladas de forma fácil y óptima.

La organización de todos los procedimientos que se realicen sobre la plataforma se verán reflejados en la jerarquización que utilizan las herramientas Microsoft ®. Con esto podemos realizar modificaciones de todo tipo sin afectar la estructura de la plataforma.

Con ello aseguramos que el código sea fácil de entender y por lo tanto escalable a diferentes niveles. De la misma forma la comunicación entre los procesos tiene una mejor visión y por lo tanto se puede seguir la línea de vida de una actividad.

5.1 FASE DE PLANIFICACIÓN

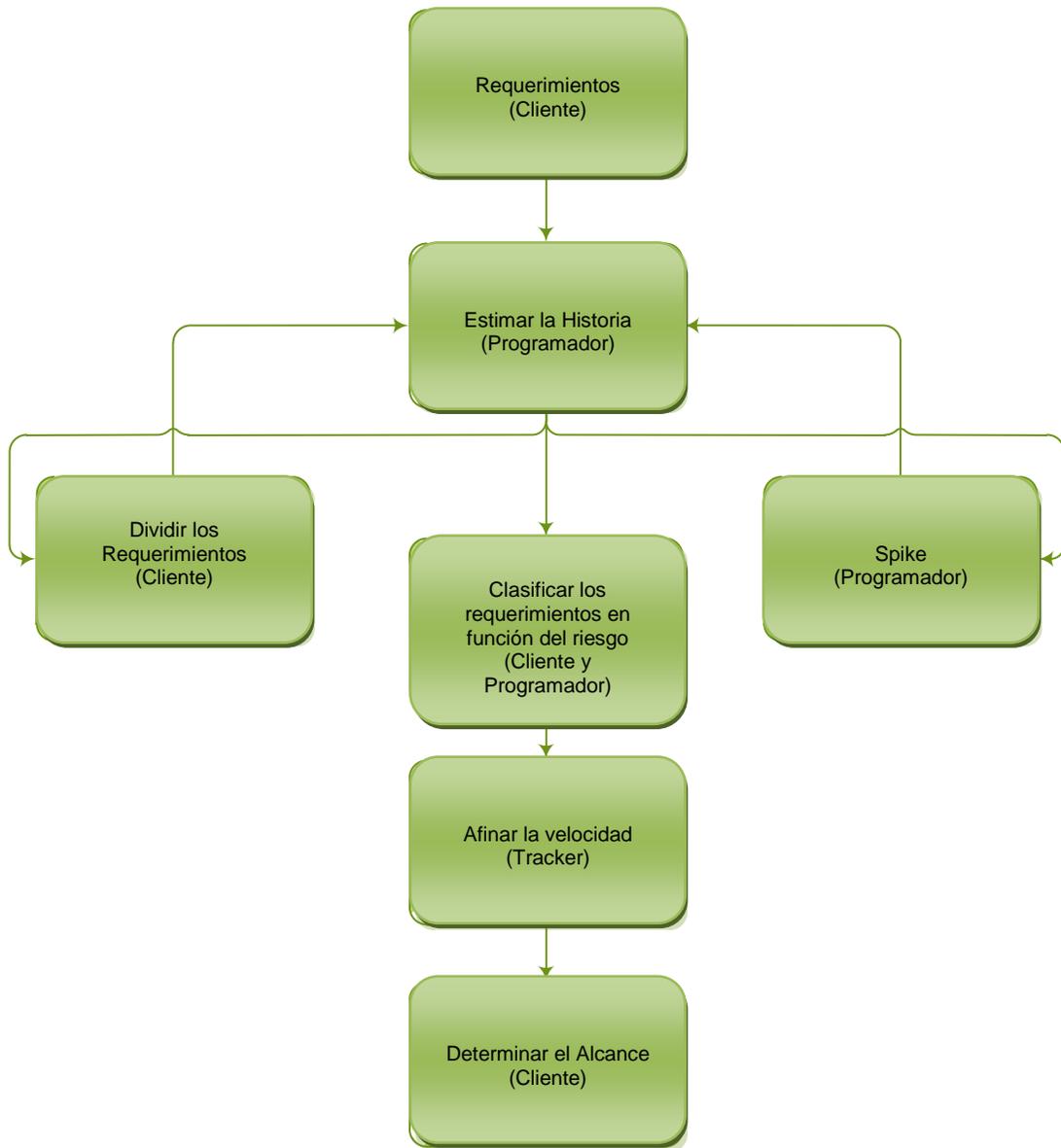
- **Requerimientos:** El principal foco de información a recolectar para el desarrollo fue el de la observación por parte del programador. La experiencia a la hora de implementaciones de éste tipo estableció el diseño de una maqueta en borrador de la plataforma inicial con herramientas robustas como Visual Studio 2010 ® y SQL Server 2008 ®.
- La entrevista estableció la factibilidad de la plataforma en un ambiente web y el tipo de conexión de datos que debía manejarse.

Estos elementos determinan el alcance del proyecto, sujeto a la toma de datos del transporte público urbano y a la toma de decisiones con el fin de la optimización del tráfico.

5.1.1 Plan de entrega. Por estar sujeto a un prototipo, el plan de entregas solo se mide en una versión básica aplicada a los diferentes servicios propuestos en los objetivos iniciales del proyecto. Esta versión beta implementa varias iteraciones

hasta su entrega, como resultado de las pruebas realizadas sobre los vehículos piloto donde fue implementado el sistema.

Ilustración 8 Desarrollo



5.2 DISEÑO

Debido a la filosofía de XP, el diseño aborda una implementación sencilla que permite realizar modificaciones sin alterar en gran medida la estructura completa del software. Para eso se aplicaron casos de usos, con el fin de determinar los componentes de la estructura total.

5.2.1 Requerimientos. El diseño incorpora una estructura de la siguiente forma.

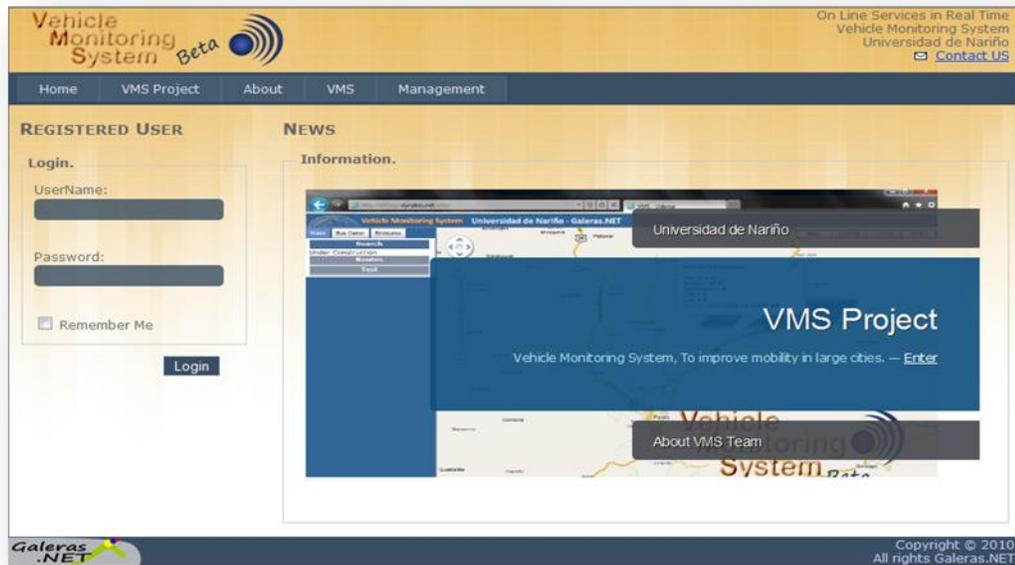
HOME PAGE OPCIÓN UNIVERSIDAD DE NARIÑO

Ilustración 9 Home Page Opción Universidad de Nariño



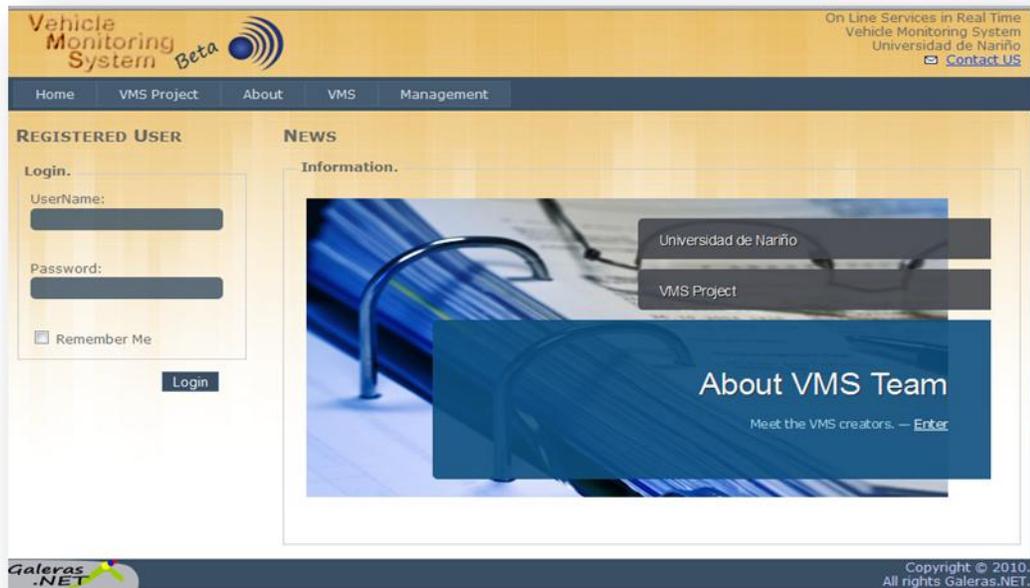
HOME PAGE OPCIÓN VMS

Ilustración 10 Home Page Opción VMS



HOME PAGE OPCIÓN ABOUT VMS TEAM

Ilustración 11 Home Page Opción About VMS Team



Para este módulo general se realizan los siguientes casos de uso:

-M
MÓDULO ADMINISTRADOR: Tiene como el acceso privilegiado al sistema. administra la página y todos los eventos que se generen a través de ella.

OBJETIVOS

Tabla 3 Objetivo No. 1

CÓDIGO	OBJ – 001
DESCRIPCIÓN	Establecer una plataforma que permita el ingreso de diferentes tipos de usuarios, como Administradores y Clientes.
ESTADO	CRITICO
COMENTARIO	

Tabla 4 Objetivo No. 2

CÓDIGO	OBJ - 002
DESCRIPCIÓN	Controlar el acceso de los usuarios al sistema así como implementar políticas de seguridad y restricción de externos.
ESTADO	CRITICO
COMENTARIO	Se definen políticas que permitan garantizar que sean los usuarios registrados quienes accedan al sistema, ya sea para consumo de sus servicios o para correcciones de la plataforma sea el caso.

Tabla 5 Objetivo No. 3

CÓDIGO	OBJ - 003
DESCRIPCIÓN	Diseñar una interfaz amigable con el usuario
ESTADO	IMPORTANTE
COMENTARIO	

Posibilita un manejo adecuado de la interfaz inicial por parte del usuario

FUNCIONALIDADES

Tabla 6 Funcionalidad No. 1

CÓDIGO	RF-FUN01	OBJETIVO ASOCIADO	OBJ-001 OBJ-002
DESCRIPCIÓN	A través de una tabla de usuarios se valida si la clave y el usuario si corresponden a entidades registradas dentro de la plataforma		
ESTADO	APROBADO		
COMENTARIO	Cada respuesta vendrá acompañada de un mensaje dependiendo de si es afirmativa o negativa,		

Tabla 7 Funcionalidad No. 2

CÓDIGO	RF-FUN02	OBJETIVO ASOCIADO	OBJ-003
DESCRIPCIÓN	Los accesos hacia los diferentes servicios están ubicados de tal forma que interactúen amigablemente con el usuario, desplegando pantallas dinámicas y controles interactivos.		
ESTADO	ACCESO		
COMENTARIO	La dinámica de los controles visualmente es atractiva al público cliente.		

- Módulo VMS:** Este módulo permite la visualización y el control de los vehículos que están registrados en el sistema. A través de una interfaz y utilizando Google Maps® se pueden visualizar las rutas y los buses respectivos. Esto entrega información en tiempo real, también accesible al usuario desde la interfaz.

OBJETIVOS

Tabla 8 Objetivo No. 4

CÓDIGO	OBJ – 004
DESCRIPCIÓN	Desarrollar una interfaz que visualice los componentes y los diferentes elementos asociados al reconocimiento en tiempo real de los vehículos registrados.
ESTADO	CRITICO
COMENTARIO	Se establece una serie de controles basados en AJAX que ofrecen recursos y animaciones para los controles estáticos.

Tabla 9 Objetivo No. 5

CÓDIGO	OBJ – 005
DESCRIPCIÓN	Administrar los datos para evitar errores de escritura y duplicidad con el fin de que las bases de datos sean coherentes.
ESTADO	IMPORTANTE
COMENTARIO	Se establece una serie de controles basados en AJAX que ofrecen recursos y animaciones para los controles estáticos.

Tabla 10 Objetivo No. 6

CÓDIGO	OBJ – 006
DESCRIPCIÓN	Interconectar la base de datos con la interfaz con el fin de que vincule los elementos almacenados al momento de la solicitud por parte del usuario.
ESTADO	CRITICO
COMENTARIO	A través de las herramientas de Visual Studio 2010 se establecen las conexiones con la base de datos, utilizando controles como Table

Tabla 11 Objetivo No. 7

CÓDIGO	OBJ – 007
DESCRIPCIÓN	Implementar controles que permitan la interacción con la interfaz de tal forma que identifique las rutas específicas, los buses activos e inactivos, las características de cada bus, etc.
ESTADO	IMPORTANTE
COMENTARIO	La visualización de la información se implementa a través de controles de Visual Studio 2010.

FUNCIONALIDADES

Tabla 12 Funcionalidad No. 3

CÓDIGO	RF- FUN03	OBJETIVO ASOCIADO	OBJ- 004 Y OBJ-005
DESCRIPCIÓN	Los accesos hacia los diferentes servicios están ubicados de tal forma que interactúen amigablemente con el usuario, desplegando pantallas dinámicas y controles interactivos.		
ESTADO	ACCESO		
COMENTARIO	La dinámica de los controles visualmente es atractiva al público cliente.		

Tabla 13 Funcionalidad No. 4

CÓDIGO	RF- FUN04	OBJETIVO ASOCIADO	OBJ- 006
DESCRIPCIÓN	La conexión con la base de datos se hace a través de controles optimizados propios de la arquitectura de Visual Studio.		
ESTADO	ACCESO		
COMENTARIO			

Tabla 14 Funcionalidad No. 5

CÓDIGO	RF-FUN05	OBJETIVO ASOCIADO	OBJ-007
DESCRIPCIÓN	El sistema visualiza las rutas específicas de cada bus, establece las características propias de cada vehículo y reporta su estado en todo momento.		
ESTADO	PROCESADO		
COMENTARIO	<p>Todos los usuarios, sean clientes u administradores, pueden obtener información acerca de este servicio.</p> <ul style="list-style-type: none"> • Módulo Management: Este módulo permite recopilar los datos almacenados en la base de datos y la transforma en información importante para el administrador. Implementa diferentes opciones según sea la necesidad. 		

OBJETIVOS

Tabla 15 Objetivo No. 8

CÓDIGO	OBJ – 008
DESCRIPCIÓN	Establecer un diseño que permita relacionar las necesidades del administrador con la información específica de los vehículos
ESTADO	IMPORTANTE
COMENTARIO	

Tabla 16 Objetivo No. 9

CÓDIGO	OBJ – 009
DESCRIPCIÓN	Establecer niveles de información que permita sintetizar los reportes a generar para la toma de decisiones.
ESTADO	IMPORTANTE
COMENTARIO	Se establecen detalles que discriminan para qué tipo de cliente se necesita la información, sea vehículo, empresa o administrador en particular.

Tabla 17 Objetivo No. 10

CÓDIGO	OBJ – 010
DESCRIPCIÓN	Administrar los datos de tal forma que no haya duplicidad a la hora de insertar vehículos u empresas.
ESTADO	CRÍTICO
COMENTARIO	

FUNCIONALIDADES

Tabla 18 Funcionalidad No. 6

CÓDIGO	RF-FUN06	OBJETIVO ASOCIADO	OBJ- 008 Y OBJ-009
DESCRIPCIÓN	El nivel de detalle de la información ingresada visualizada en tablas permite que se establezca un control sobre los datos almacenados. Con ello se diseñan reportes con el fin de lograr optimizaciones de los procesos de tráfico.		
ESTADO	PROCESADO		
COMENTARIO	Son acciones de tipo administrador		

Tabla 19 Funcionalidad No. 7

CÓDIGO	RF-FUN07	OBJETIVO ASOCIADO	OBJ- 010
DESCRIPCIÓN	A través de acciones programadas se controla la duplicidad y los errores en los datos. Con ello se garantiza que la información generada este depurada en su totalidad.		
ESTADO	PROCESADO		
COMENTARIO	Son acciones de tipo administrador		

.....R

EQUERIMIENTOS NO FUNCIONALES

Tabla 20 Requerimiento no funcional No. 1

CÓDIGO	RNF-001	OBJETIVO ASOCIADO	OBJ-001 HASTA OBJ-010
DESCRIPCIÓN	El sistema debe maneja un modelo cliente servidor, orientado a la web y a consumir servicios de internet como Google Maps ®.		

Tabla 21 Requerimiento no funcional No. 2

CÓDIGO	RNF-002	OBJETIVO ASOCIADO	OBJ-001 HASTA OBJ-010
DESCRIPCIÓN	La plataforma web debe contar con un diseño óptimo, atractivo y eficiente que permita que los usuarios de todo nivel intuyan su funcionamiento.		

Tabla 22 Requerimiento no funcional No. 3

CÓDIGO	RNF-003	OBJETIVO ASOCIADO	OBJ- 002,005,006, 010
DESCRIPCIÓN	El sistema debe utilizar una base de datos como SQL Server 2008 para la administración de datos.		

ESTADO DE ACTORES (ROLES)

Tabla 23 Actor – Usuario Anónimo

NOMBRE:	Usuario Anónimo
DESCRIPCIÓN	Cualquier usuario que a través de la página puede consumir los servicios de visualización en VMS.
OBJETIVOS	Ingresar al sistema y ejecuta funciones comunes

Tabla 24 Actor – Administrador

NOMBRE:	Administrador
DESCRIPCIÓN	Controla en su totalidad el sitio, las funciones, los reportes y las bases de datos
OBJETIVOS	Administrar todas las funciones y servicios del Sistema VMS.

Tabla 25 Actor – Operador

NOMBRE:	Operador
DESCRIPCIÓN	Visualiza los componentes de información de cada uno de los vehículos, así como sus rutas específicas y tiempos de partida.
OBJETIVOS	Controlar el tráfico de los buses en el día.

Tabla 26 Actor – Usuario Registrado

NOMBRE:	Usuario Registrado
DESCRIPCIÓN	El usuario puede visualizar su grupo de buses con registro administrando únicamente aquellos que pertenezcan a su empresa

OBJETIVOS

Ingresar al sistema y ejecutar funciones de administración restringidas.

5.2.2 Diagramas de clases de la aplicación web.

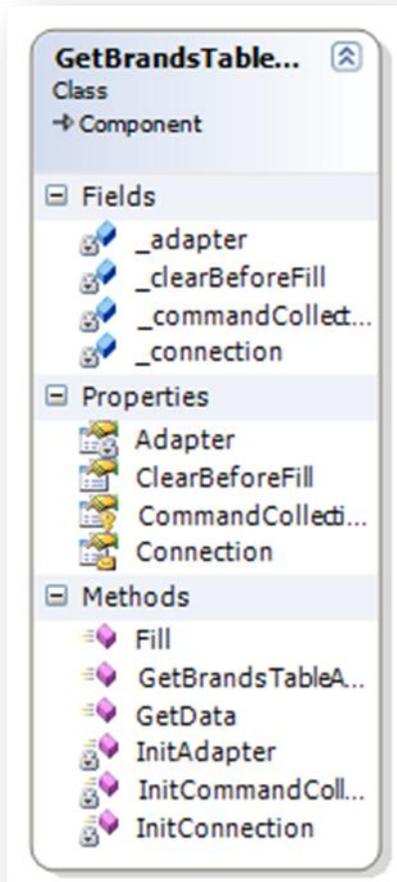


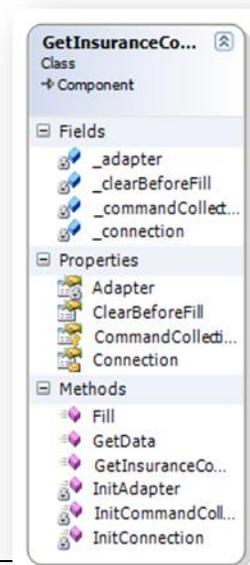
Ilustración 12 Diagrama de clase GetBrandsTable

GetBrandsTable: Administra las marcas de cada uno de los vehículos que se encuentren registrados en el sistema VMS.

Estas identifican el tipo de vehículo, permitiendo darle ciertas características que lo diferencien de otros también registrados.

Ilustración 13 Diagrama de clase GetInsuranceCompanies

GetInsuranceCompanies: Esta clase administra las diferentes compañías que tengan a su cargo vehículos registrados en el sistema. Esto permite agrupar los vehículos y así organizar las entidades del sistema.



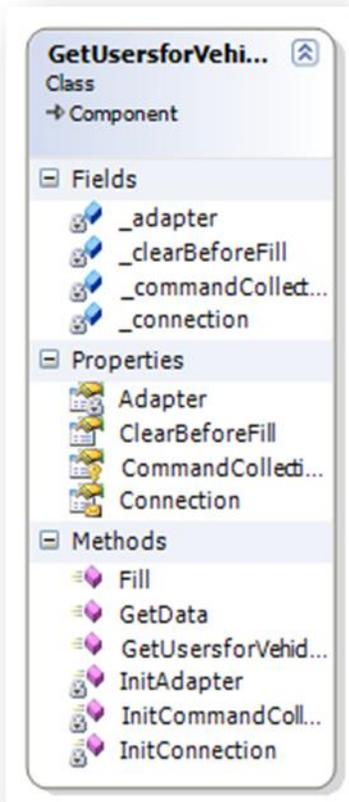


Ilustración 14 Diagrama de clase
GetUsersforVehicles

GetUsersforVehicles: Esta clase permite administrar los vehículos y que usuarios se encuentran relacionados con ellos.

Ilustración 15 Diagrama de clase
ProfileCommon

ProfileCommon: Permite consolidar una referencia de los usuarios del sitio web, obteniendo información personal y básica.

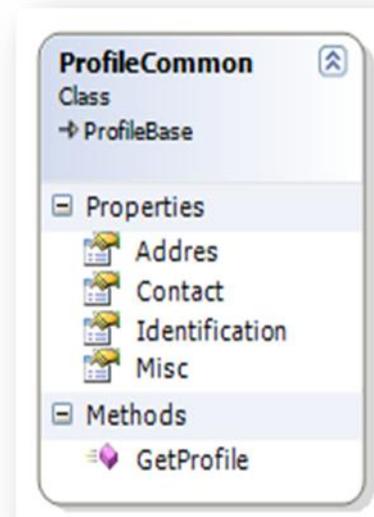
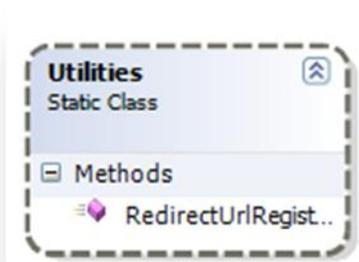
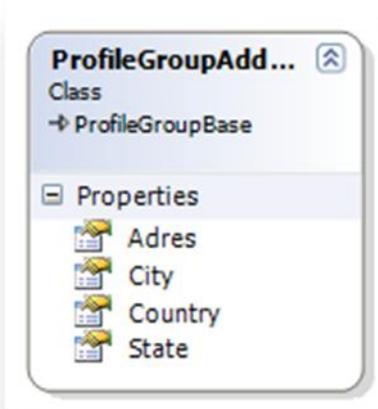


Ilustración 16 Diagrama de clase
Utilities



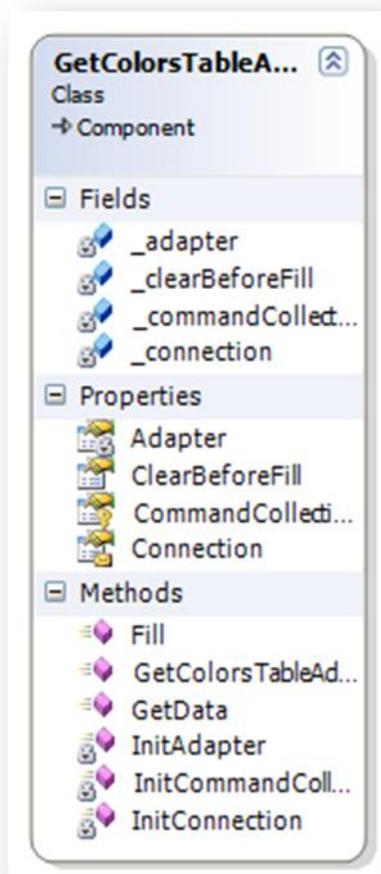
Utilities: Sirve como puente URL de direccionamiento.

Ilustración 17 Diagrama de clase ProfileGroupAddress



ProfileGroupAddress: Se encarga de definir un perfil para las direcciones de grupo.

Ilustración 18 Diagrama de clase GetColorsTable



GetColorsTable: Sirve como propiedad de los vehículos con el fin de definirles características que identifiquen mejor el objeto registrado.

Ilustración 19 Diagrama de clase GetLinesTable

GetLinesTable: Caracteriza al vehículo identificando a qué tipo de línea pertenece, entendiéndose como la categoría del automóvil

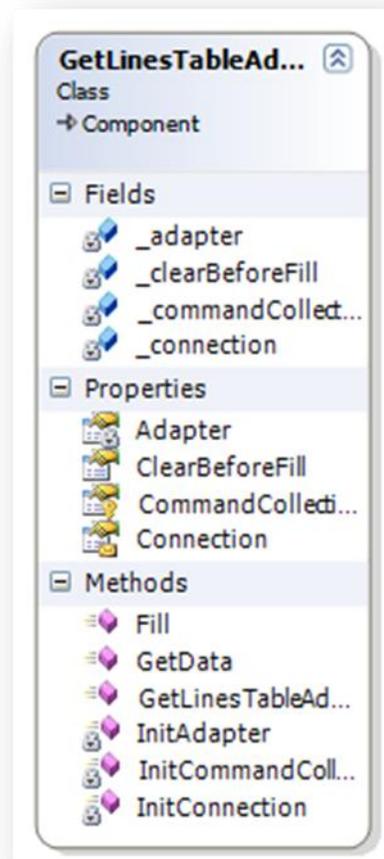
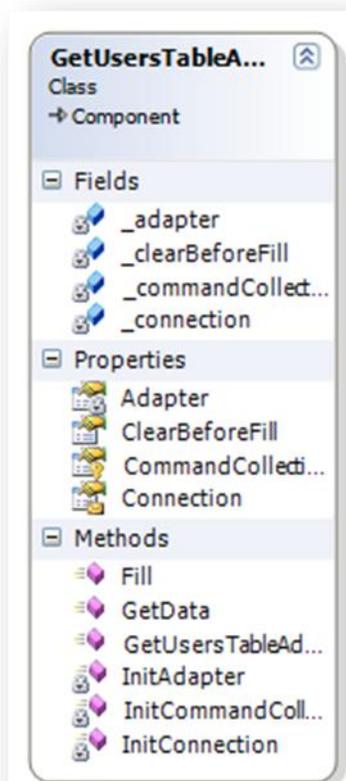


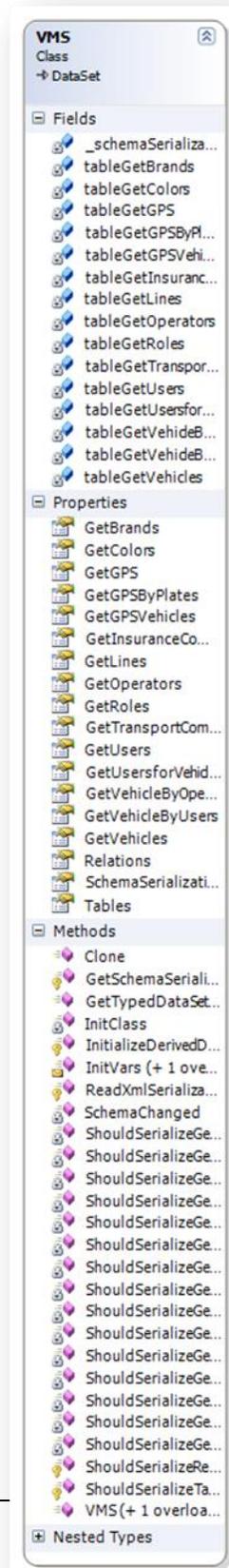
Ilustración 20 Diagrama de clase GetUsers



GetUsers: Se encarga de administrar a los usuarios que podrán acceder al sistema.

Ilustración 21 Diagrama de clase VMS

VMS: Esta súper clase en realidad administra el sitio web, sirviendo de interfaz de captura de datos, así como de la asignación de las relaciones entre los vehículos, los eventos, empresas y usuarios. Permite, a través de interfaces amigables con el usuario, obtener información necesaria a la hora de realizar un registro del vehículo, características de éste y su posición en tiempo real en el mapa.



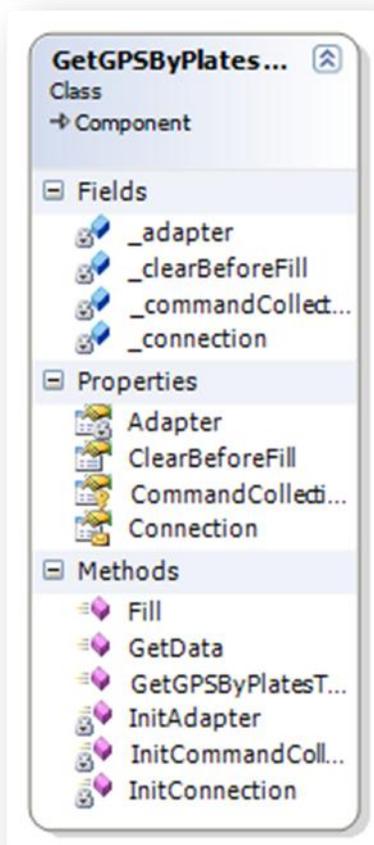
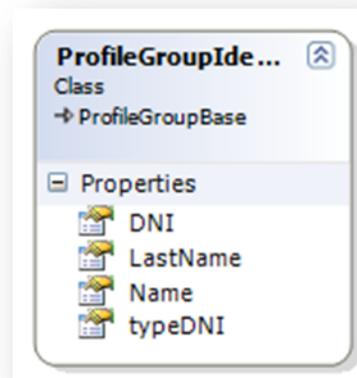


Ilustración 22 Diagrama de clase GetGPSByPlates

GetGPSByPlates: La función de esta clase es de servir como apoyo a las solicitudes de los usuarios a la hora de ubicar el identificador de una GPS asociada a un vehículo por placa.

Ilustración 23 Diagrama de clase ProfileGroupIdentifier



ProfileGroupIdentifier: Administra el identificador de grupo de perfil. Esto se realiza con el fin de caracterizar ciertos grupos de empresas dentro del sistema y así lograr organización entre las entidades.

Ilustración 24 Diagrama de clase ProfileGroupContact

ProfileGroupContact: Permite caracterizar a los contactos en grupos para un mejor manejo de información.

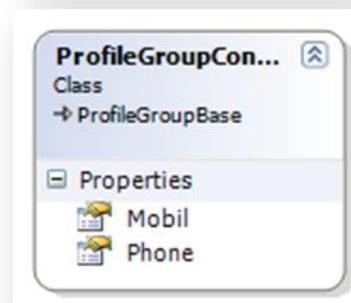
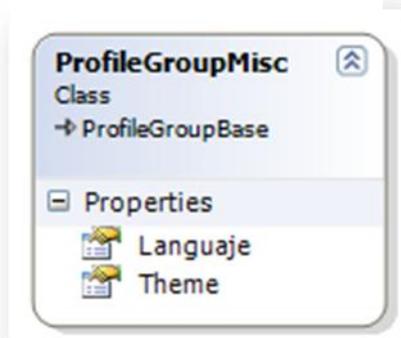


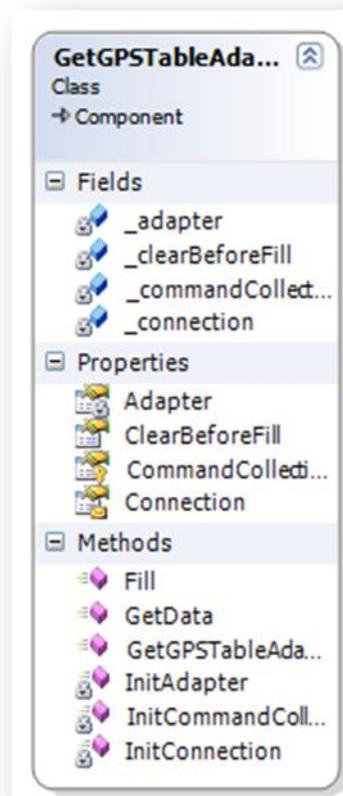
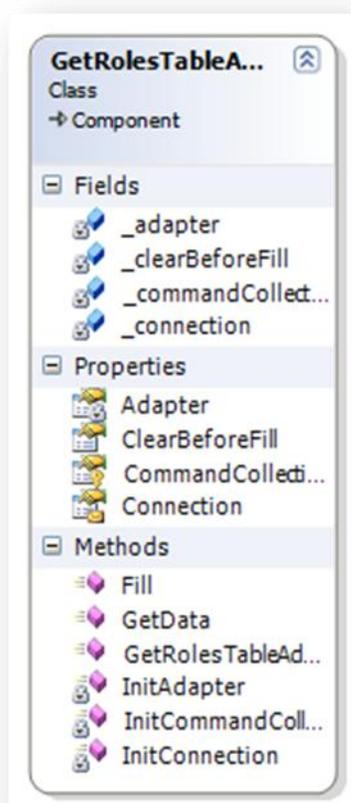
Ilustración 25 Diagrama de clase ProfileGroupMisc



ProfileGroupMisc: Este perfil es generado automáticamente por Visual Studio con el fin de llevar una relación de los datos.

GetGPS: Permite identificar una GPS registrada de acuerdo al patrón de búsqueda.

Ilustración 26 Diagrama de clase GetGPS



GetRoles: La clase permite identificar los roles de los usuarios. A saber, dentro del aplicativo se reconocen 4 tipos de usuarios: el administrador, el operario, el usuario registrado y el usuario común.

Ilustración 27 Diagrama de clase GetRoles

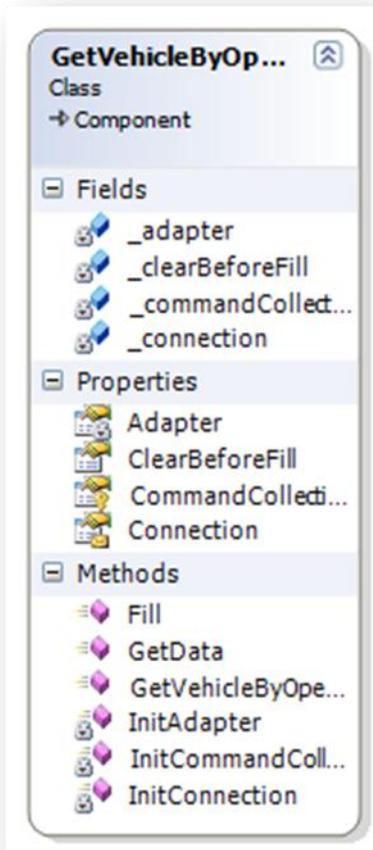
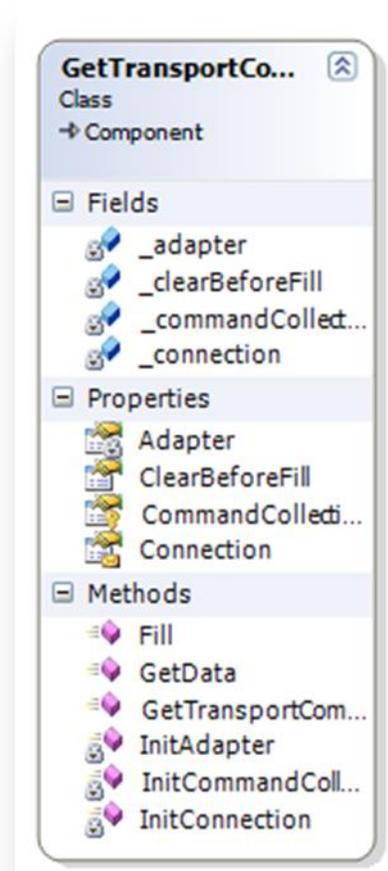


Ilustración 28 Diagrama de clase GetVehicleByOperator

GetVehicleByOperator: Esta clase se encarga de realizar una búsqueda de los vehículos registrados por operador como parámetros.

Ilustración 29 Diagrama de clase GetTransportCompany

GetTransportCompany: Administra a las compañías de transporte, en este caso las de transporte público.



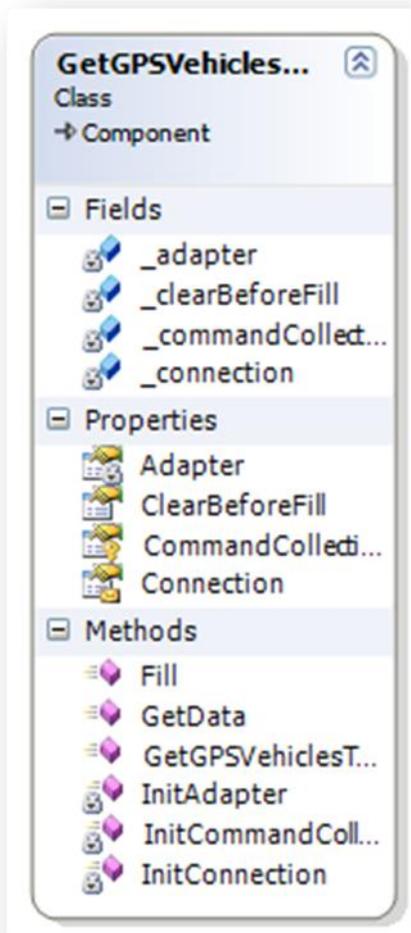
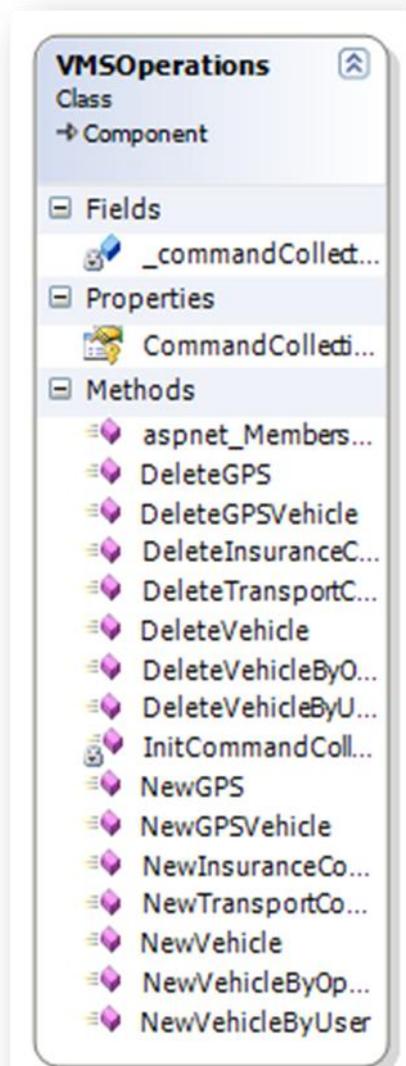


Ilustración 30 Diagrama de clase GetGPSVehicles

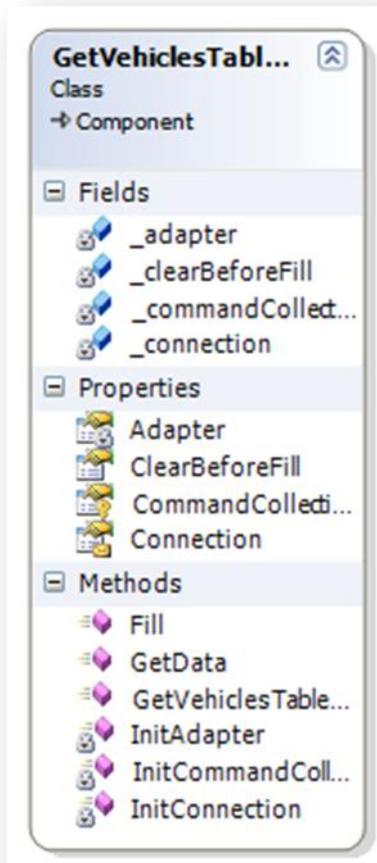
GetGPSVehicles: Pêrmite relacionar los GPS registrados con los vehiculos registrados.

Ilustración 31 Diagrama de clase VMSOperations



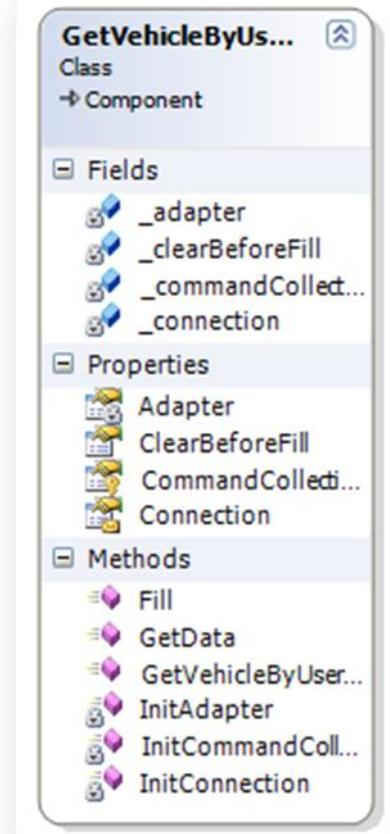
VMSOperations: Esta clase implementa los métodos que se utilizan en la aplicación web. Permite realizar las operaciones de registro sobre el sistema.

Ilustración 32 Diagrama de clase
GetVehiclesTable



GetVehiclesTable: Administra los vehículos. Esta clase es implementada por el sistema de desarrollo.

Ilustración 33 Diagrama de clase
GetVehicleByUser



GetVehicleByUser: Permite administrar vehículos por usuario.

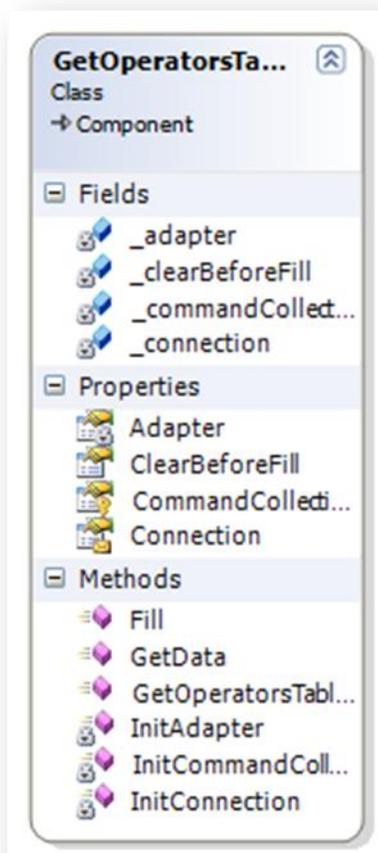
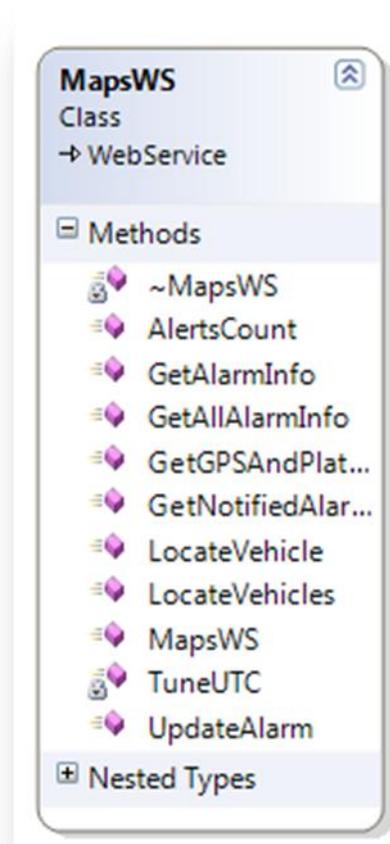


Ilustración 34 Diagrama de clase GetOperators

GetOperators: Permite administrar los operadores registrados por los administradores.

Ilustración 35 Diagrama de clase MapWS

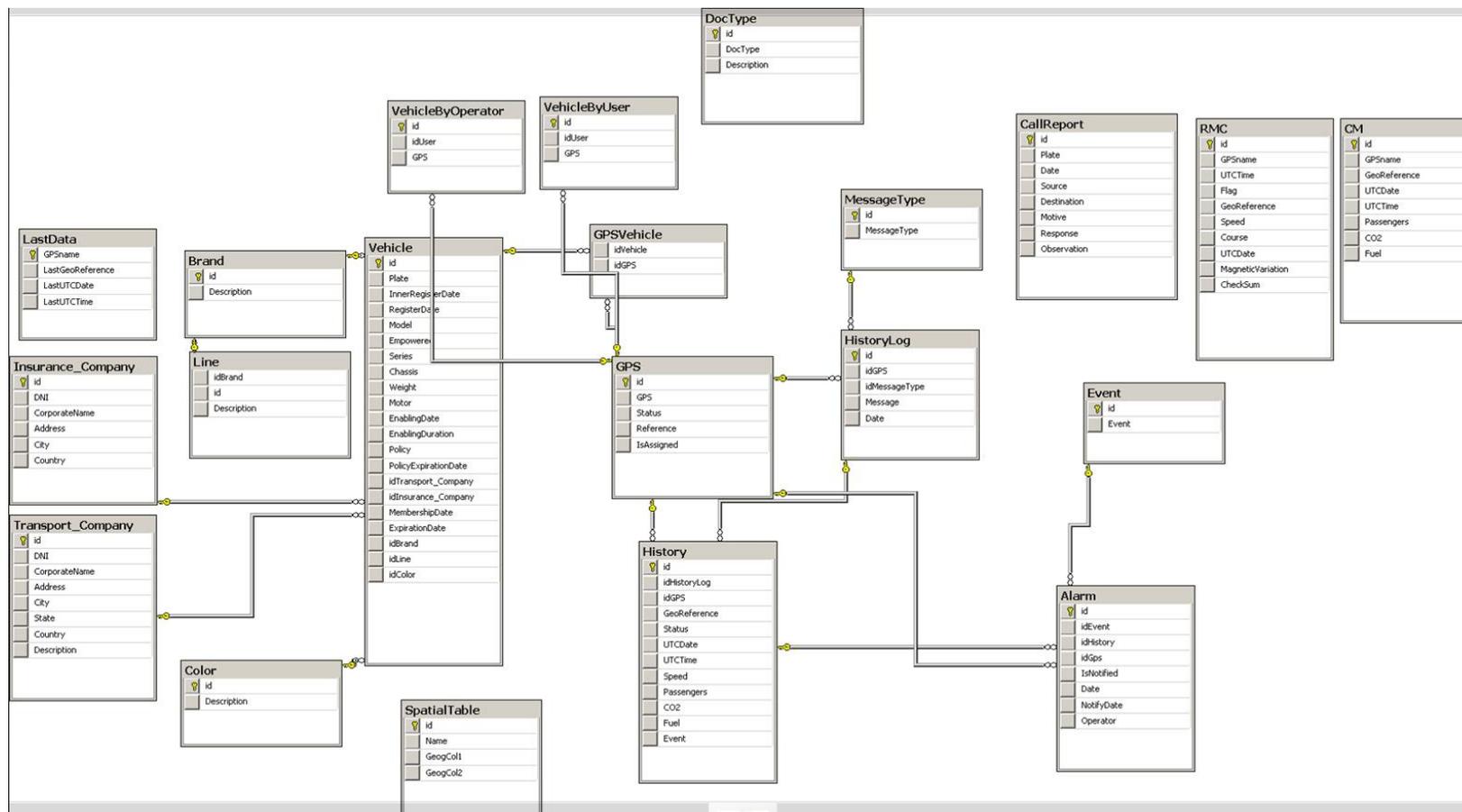


5.2.3 Diagrama de clases – web service.

MapsWS: Esta clase permite la administración visual y el consumo del servicio por parte de la aplicación sobre Google Maps®. Sobre esta clase se realizan todas las operaciones de posición, notificaciones, alarmas, eventos, etc.

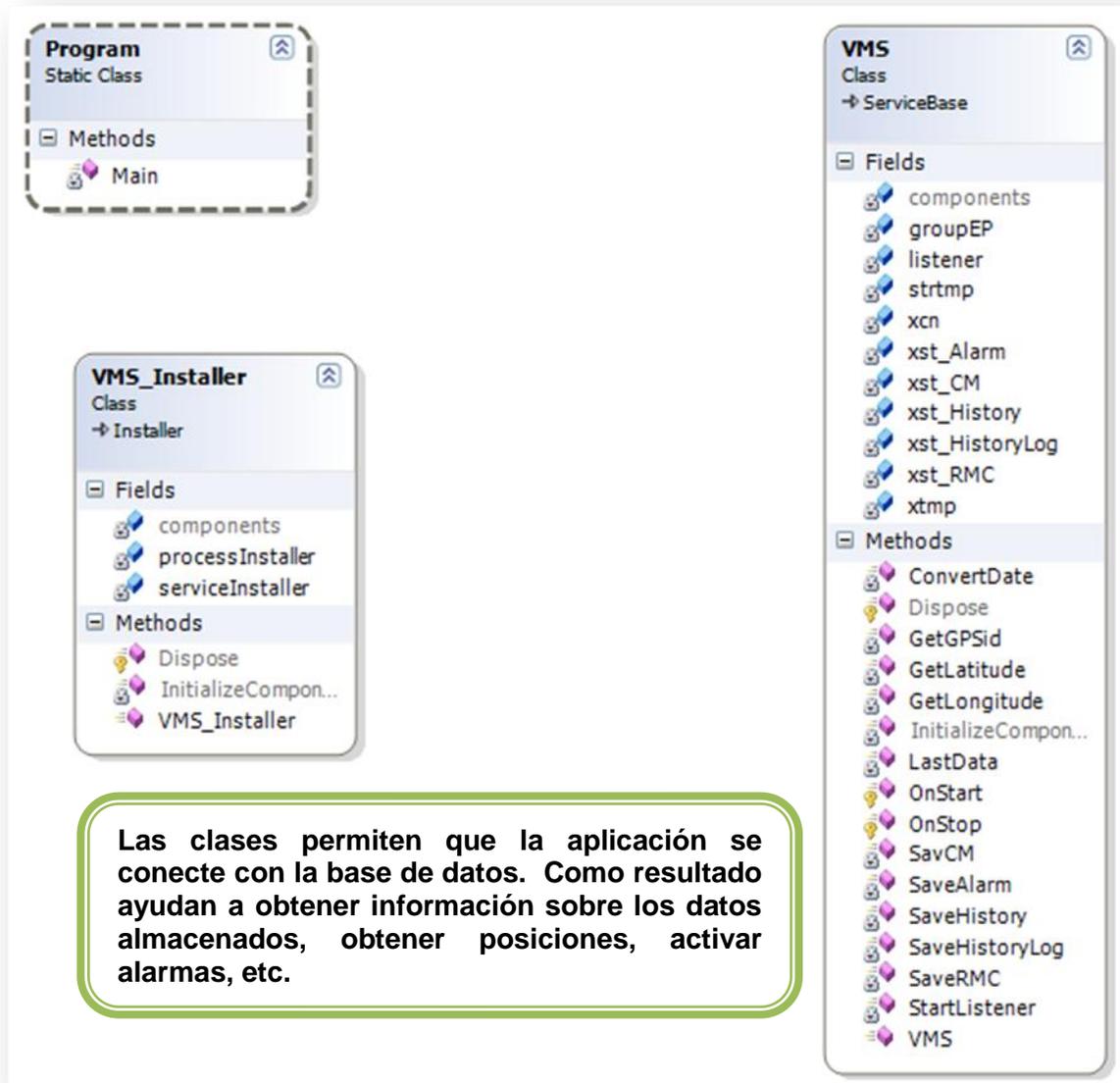
5.2.4 Diagrama de clases de las tablas sobre la base de datos. Este diagrama establece las relaciones existentes entre las tablas en la base de datos. Algunas tablas se encuentran sin relación y sirven como soporte a las funciones de otras.

Ilustración 36 Diagrama de clases de las tablas de la base de datos VMS



5.2.5 diagrama de clases – servicio vms. Estas clases permiten que puedan consumirse los diferentes servicios de la aplicación.

Ilustración 37 Diagrama de clase Servicio VMS



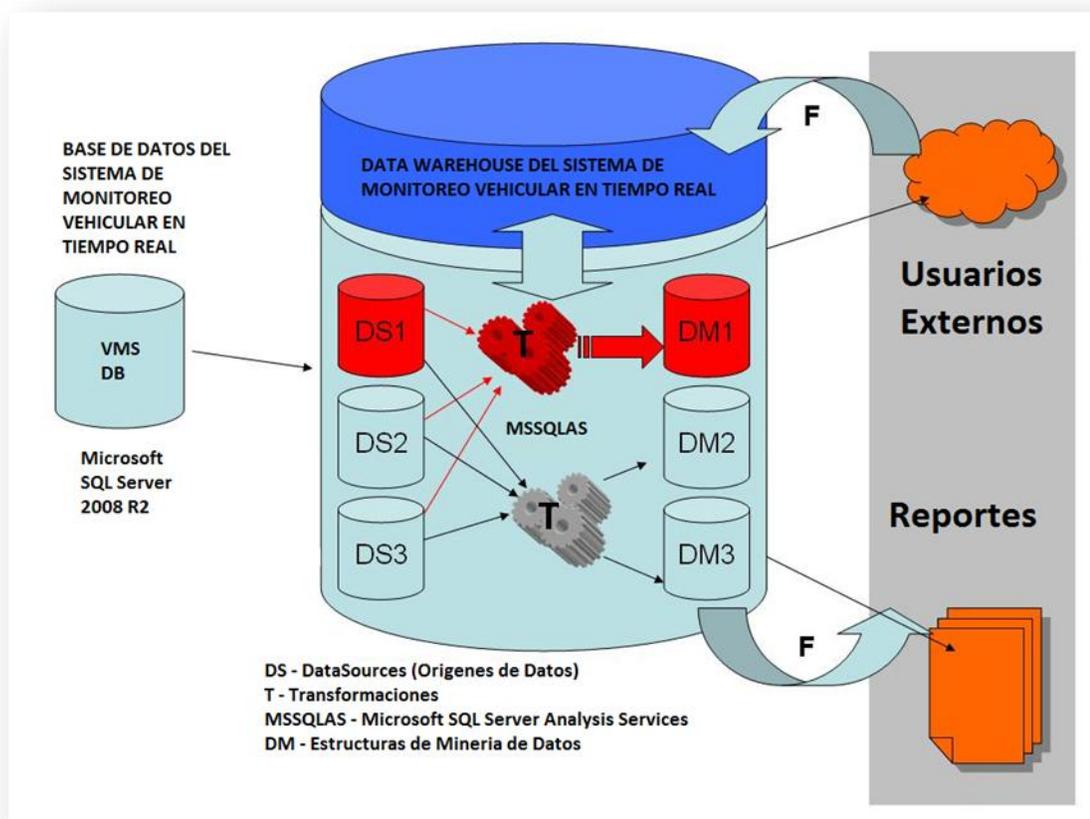
5.3 DATA WAREHOUSE DEL SISTEMA DE MONITOREO VEHICULAR EN TIEMPO REAL

La Data Warehouse (Bodega de Datos) del sistema de monitoreo vehicular en tiempo real es un conjunto de datos integrados u orientados a un propósito, que varían con el tiempo y que no son transitorios, los cuales soportan el proceso de toma de decisiones de la administración y está orientada al manejo de grandes volúmenes de datos provenientes del monitoreo vehicular en tiempo real en la ciudad de Pasto.

Estos datos cubren períodos de tiempo de recolección de información de los vehículos, lo que trae consigo que se tengan diferentes esquemas de los datos fuentes, La concentración de esta información está orientada a su análisis para apoyar la toma de decisiones oportunas y fundamentadas, Previo a su utilización se debe aplicar procesos de análisis, selección y transferencia de datos seleccionados desde las fuentes.

La capa de inteligencia de negocios sigue el esquema a continuación:

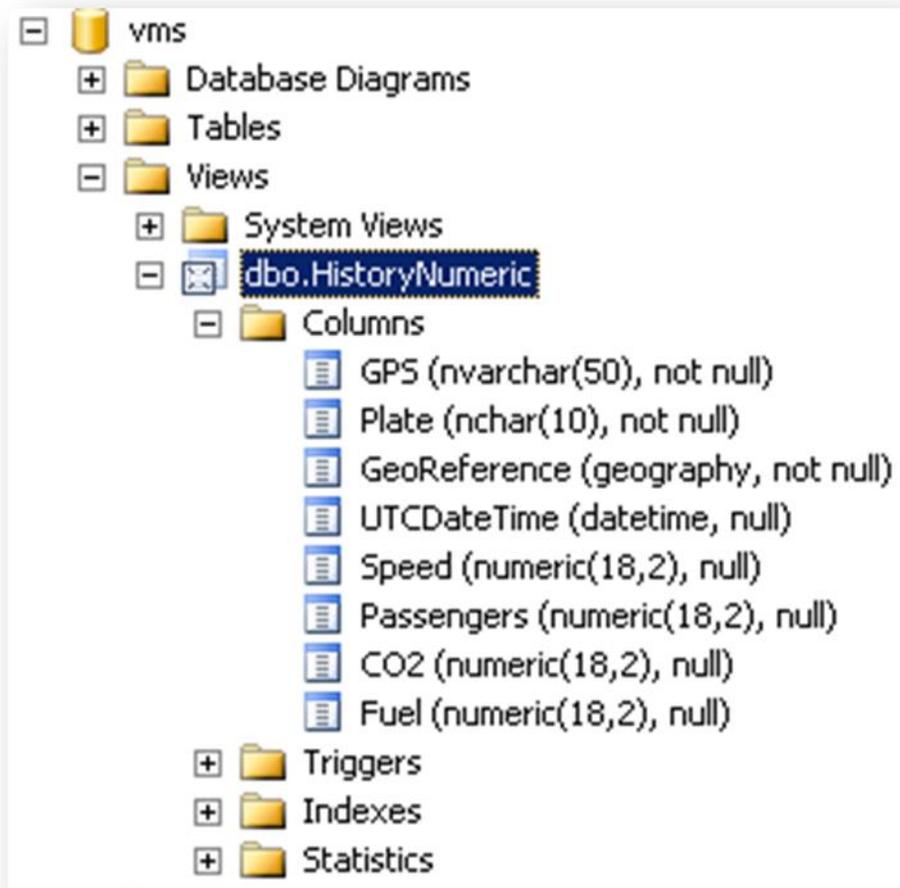
Ilustración 38 Data Warehouse del sistema de monitoreo vehicular en tiempo real



El Data Warehouse toma los datos de la base de datos del sistema vehicular en tiempo real y realiza los siguientes procesos:

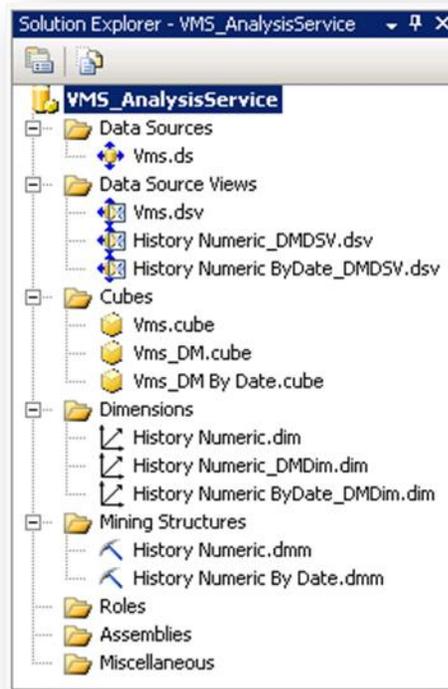
- Limpieza de datos del motor. Esto se lo realiza a través de una vista de la base de datos que contiene la siguiente estructura:

Ilustración 39 Estructura del Data Warehouse



- Una vez limpiados los datos del motor se crea la estructura general del proyecto de inteligencia de negocios:

Ilustración 40 Estructura general proyecto inteligencia de negocios



Al final obtendremos una estructura como la siguiente:

Ilustración 41 Estructura obtenida

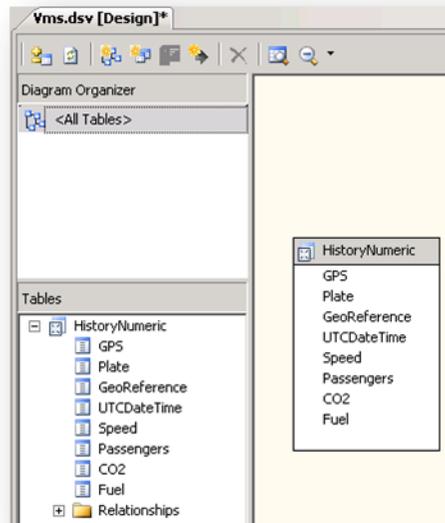
Column	Alias	Table	Output	Sort Type	Sort Order	Filter	Or...	Or...
GPS		GPS	<input checked="" type="checkbox"/>					
Plate		Vehicle	<input checked="" type="checkbox"/>					
GeoReference		History	<input checked="" type="checkbox"/>					
dbo.GetDateTi...	UTCda...		<input checked="" type="checkbox"/>					
CONVERT (nu...	Speed		<input checked="" type="checkbox"/>					

```

SELECT  dbo.GPS.GPS, dbo.Vehicle.Plate, dbo.History.GeoReference, dbo.GetDateTime(dbo.History.UTCDate, dbo.History.UTCTime) AS UTCDateTime,
        CONVERT(numeric(18, 2), dbo.History.Speed) AS Speed, CONVERT(numeric(18, 2), dbo.History.Passengers) AS Passengers,
        CONVERT(numeric(18, 2), dbo.History.CO2) AS CO2, CONVERT(numeric(18, 2), dbo.History.Fuel) AS Fuel
FROM    dbo.History INNER JOIN
        dbo.GPS ON dbo.History.idGPS = dbo.GPS.id INNER JOIN
        dbo.GPSVehicle ON dbo.GPS.id = dbo.GPSVehicle.idGPS INNER JOIN
        dbo.Vehicle ON dbo.GPSVehicle.idVehicle = dbo.Vehicle.id
    
```

- Los orígenes de datos son configurados de la siguiente forma:

Ilustración 42 Origen de datos



Ver anexo 2 Código Fuente Data Warehouse

- Luego se procede a crear el cubo multidimensional que responde a la estructura a continuación:

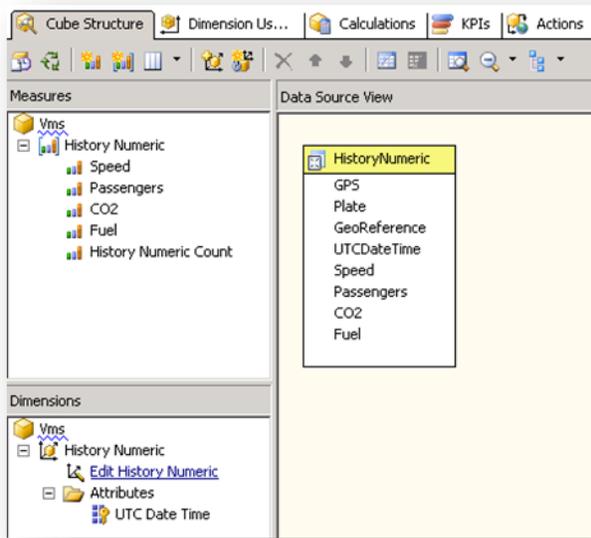
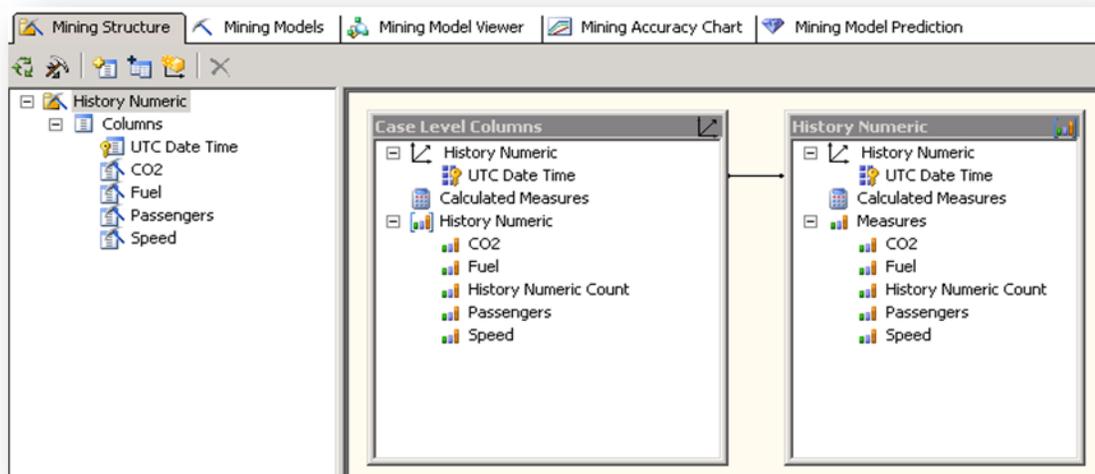


Ilustración 43 Cubo multidimensional

- Establecer la estructura de minería con el Algoritmo MICROSOFT TIME SERIES

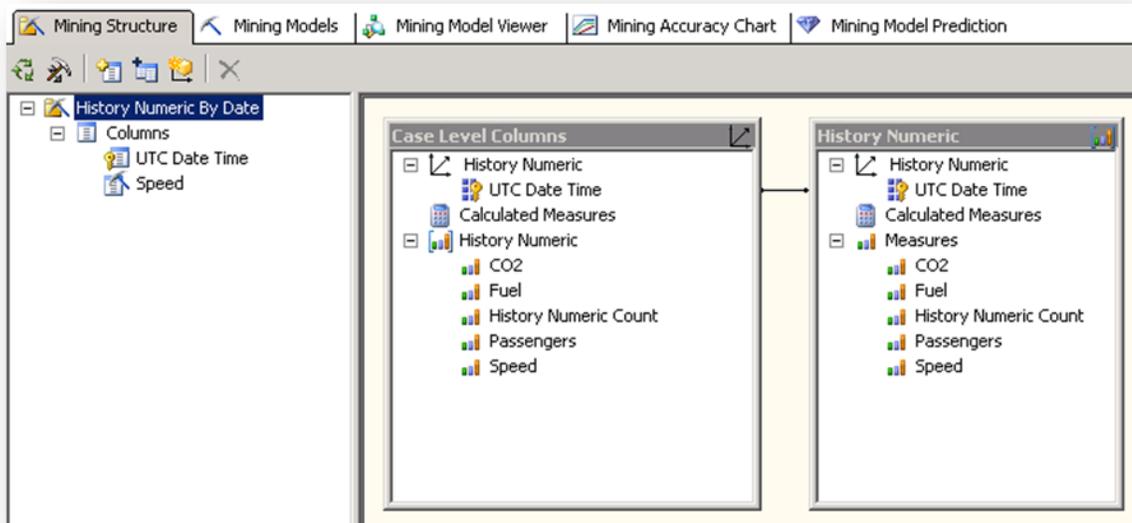
Ilustración 44 Estructura de minería - Algoritmo Microsoft Time Series



Ver anexo 2 Código Fuente Data Warehouse

- Establecer la estructura de minería con el Algoritmo MICROSOFT ASOCIATION RULES

Ilustración 45 Estructura de minería - Algoritmo Microsoft Association Rules



CAPITULO 6 DESARROLLO

6.1 INTRODUCCIÓN

La plataforma fue desarrollada utilizando las herramientas de Visual Studio 2010® en lenguaje C# debido a su compatibilidad con otros sistemas, entre ellas el sistema gestor de base de datos SQL Server 2008 R2®.

6.2 ARQUITECTURA DE LA PLATAFORMA

Se reconocen tres módulos generales. El módulo principal que restringe el ingreso de usuarios al sistema, a saber, según los niveles de privilegios otorgados inicialmente por los administradores. El segundo módulo VMS que permite visualizar en tiempo real la posición y las rutas específicas de cada vehículo de transporte público de la ciudad de Pasto. El tercer módulo Management que permite registrar los vehículos y de la misma forma realizar reportes para la toma de decisiones.

6.3 MÓDULO PRINCIPAL

Este módulo se encarga de mostrar las opciones de los diferentes servicios que maneja la plataforma, así como el de ser portal para el ingreso de los varios tipos de usuario al portal.

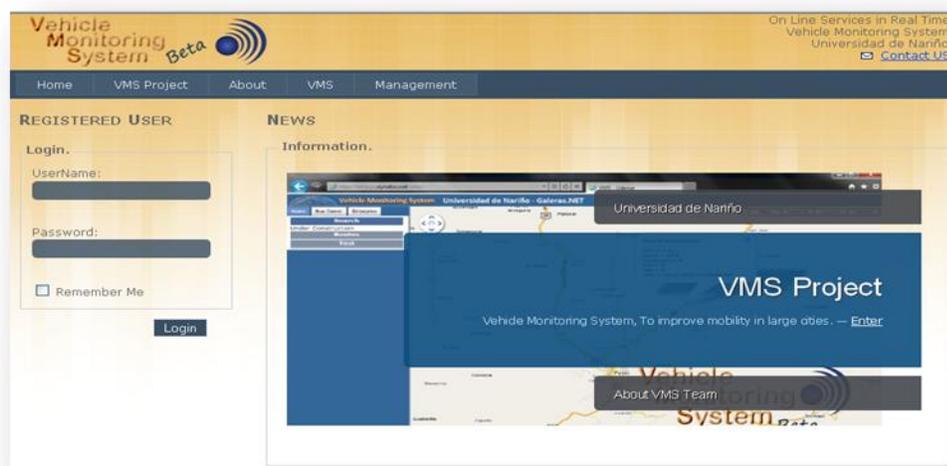
Ilustración 46 Módulo de Acceso para usuarios



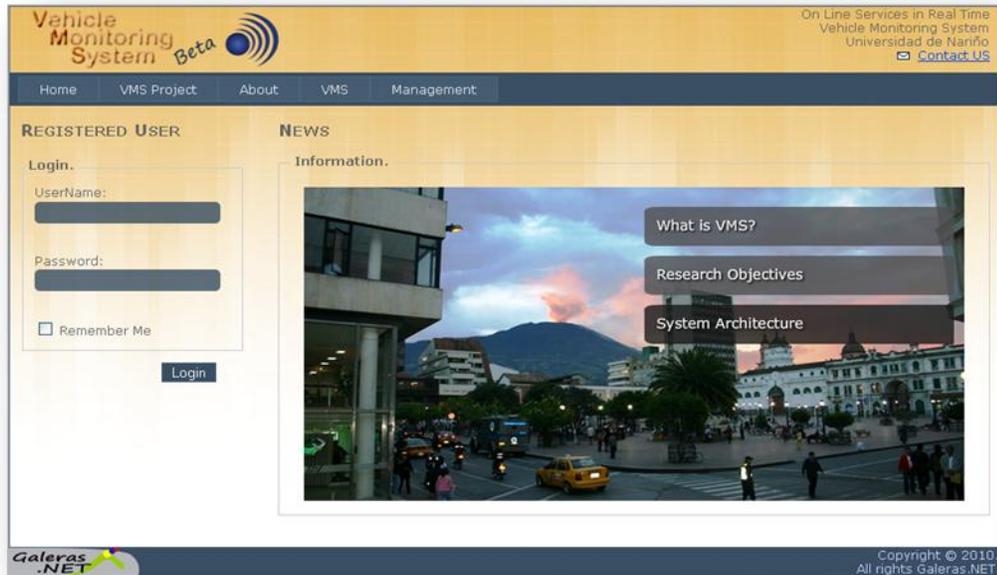
6.4 MÓDULO VMS

Este módulo permite controlar visualmente el estado real de los vehículos y sus eventos generados durante su ciclo de vida en la ruta en el día. La interfaz inicial incluye información sobre el equipo de proyecto, la plataforma para ingresar al sistema.

Ilustración 47 Módulo VMS

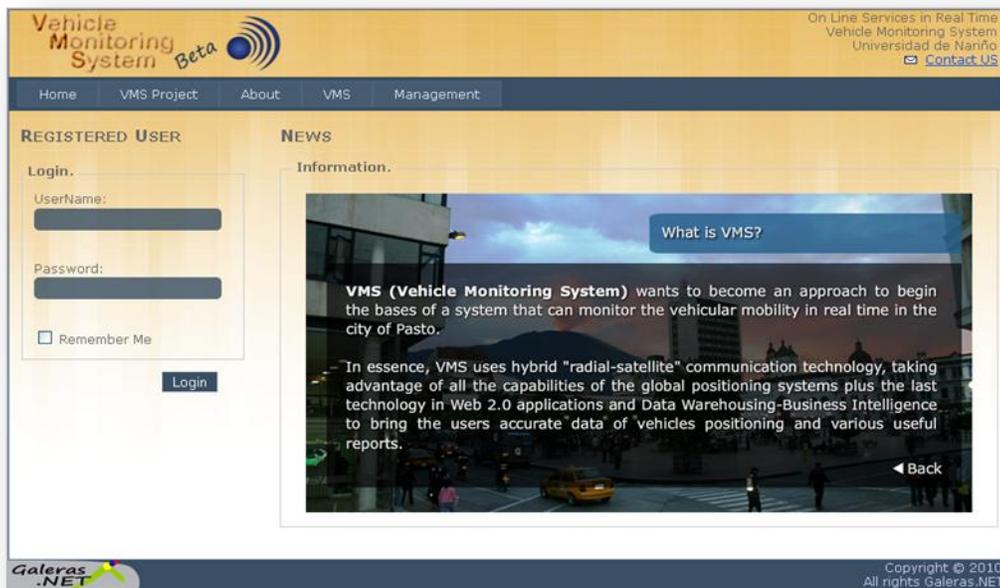


Este vínculo direcciona automáticamente a la siguiente página:
Ilustración 48 Sección News



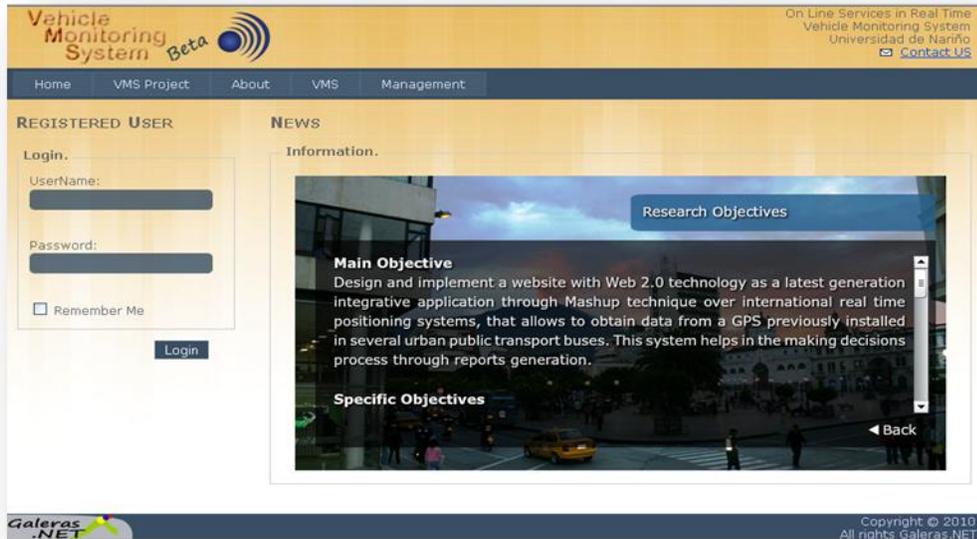
Sobre esta página se puede visualizar las siguientes opciones:

Ilustración 49 Opciones de la sección News



6.4.1 What's vms. Este vínculo brinda información de la función de VMS y del servicio que provee.

Ilustración 50 Sección What's VMS



6.4.2 Research objectives. Aquí se mencionan los objetivos de la investigación, generales y específicos que brindan información acerca del alcance del proyecto.

Ilustración 51 Sección Research Objectives



6.4.3 System architecture. Identifica la estructura completa del Sistema de Monitoreo en Tiempo Real VMS.

6.5 APLICACIÓN VMS.

Ilustración 52 Aplicación VMS



VMS contiene los siguientes controles.

6.5.1 Pestaña home – botones de control.

Ilustración 53 Pestaña Home – Botones de Control



- **Search:** Permite la búsqueda de vehículos previamente registrados en la base de datos.
- **Routes:** Permite la búsqueda de rutas activas.
- **Test:** Realiza un análisis básico de la ruta activa.

6.5.2 Pestaña bus owner.

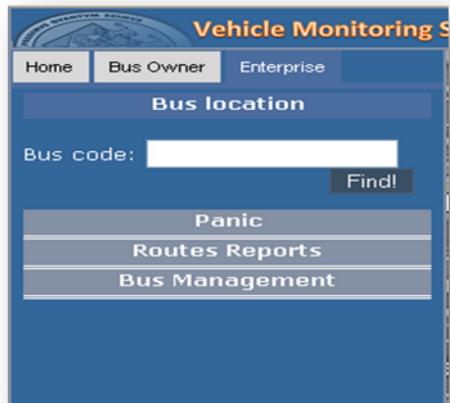
Ilustración 54 Pestaña Bus Owner



- **CHOOSE A VEHICLE:** Al desplegar el botón se puede escoger un vehículo para ejecutar alguna acción.
- **MONEY REPORT GENERATOR:** Permite visualizar un reporte de ingresos sobre una ruta específica.
- **LOGS:** Reporta los eventos sobre una ruta específica.

6.5.3 Pestaña Enterprise.

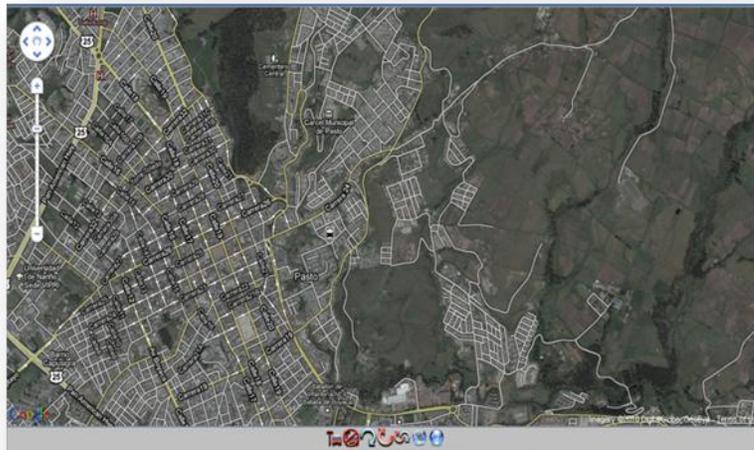
Ilustración 55 Pestaña Enterprise



- **BUS CODE:** Manualmente se puede ingresar el número de la ruta del vehículo del que se desee generar información.

6.5.4 Mapa.

Ilustración 56 Mapa



El mapa de referencia permite ubicar visualmente los vehículos que se encuentren activos e inactivos. También muestra las rutas en funcionamiento. El mapa es proporcionado a través del servicio de Google Maps®.

- **PROPIEDADES DEL MAPA**

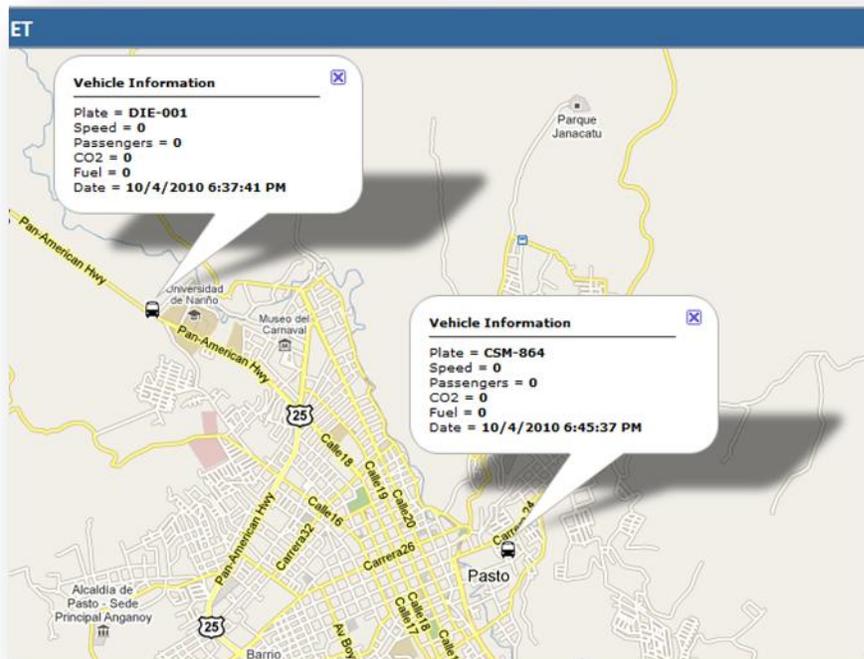
Ilustración 57 Propiedades del mapa – Barra de opciones



Estos controles cambian el tipo de visualización del mapa, dejando solamente las rutas, el mapa en relieve con vista satelital o el modo híbrido, que pone las rutas y las vías sobre la vista en relieve del mapa.

- **CONTROLES DE INFORMACIÓN**

Ilustración 58 Controles de información



Estos controles permiten dar cierta información específica como identificador de la GPS, número de pasajeros, velocidad, gasolina, fecha y hora del evento registrado.

6.6 MÓDULO MANAGEMENT

Management se encarga de la administración de la base de datos de la plataforma. También reporta información desde la base de datos y sobre esta realiza estadísticas para la toma de decisiones.

Ilustración 59 Modulo Management

Vehicle Monitoring System Beta

Home VMS Management About

VEHICLES: 2

New Edit Delete

ID	Plate	Serial Register	Register	Model	Employment	Series	Chassis	Weight	Motor	Fueling	Duration	Policy	Policy Exp.	Transport Co.	Insurance Co.	Membership	Expiration	Brand	Line	Color
222	4479115-702-4234-4448-11134614243	139-884	10/30/2010	2000	2000	1234	1234	1200	1500	10/15/2010	12	exp	10/18/2010	Transport Company 1	Insurance Company 1	10/18/2010	10/21/2010	CHRYSLER	1500	GRN
223	4715444-1350-4170-wad-40366742237	108-401	10/03/2010	2010	2010	exp	exp	113	exp	10/20/2010	12	exp	10/18/2010	Transport Company 1	Insurance Company 1	10/22/2010	10/18/2010	NISSAN	45-128	AZUL BLURPA

Galerías .NET

Copyright © 2010. All rights Galerías .NET

6.6.1 Vehicles. Contiene tres secciones que controlan el registro de los vehículos, las GPS y en qué bus se encuentran asignados.

- **Vehicles:** Permite el registro de los buses y el control de información acerca de estos.

Ilustración 60 Sección Vehicles

#	Plata	Inner Register	Register	Model	Empowered	Series	Chassis	Weight	Motor	Enabling	Duration	Policy	Policy Exp.	Transport Co.	Insurance Co.	Membership	Expiration	Brand	Line	Color
498A-1815de81c3b	COM-884	10/03/2010	10/20/2010	2000	2000	1234	1234	1200	1.500	10/15/2010	12	asf	10/15/2010	Transport Company 1	Insurance Company 1	10/16/2010	10/27/2010	CHEVROLET	1.500	GRIS
4515d6af-9104-422d-8483-9d9c19423f	DRE-001	10/03/2010	10/20/2010	2010	2010	asf	asf	123	asf	10/20/2010	12	asf	10/18/2010	Transport Company 1	Insurance Company 1	10/22/2010	10/19/2010	AUTECO	KB-125	AZUL BRUNO
2c3d9142-7043-4679-85af-563598a1a498	qwe	10/04/2010	10/10/2010	2010	2010	qwe	qwe	1234	1234	10/10/2010	120	asf	10/12/2010	Transport Company 1	Insurance Company 1	10/20/2010	10/14/2010	AUTECO BUAI	FUS CLASSIC SL	AZUL BLANCO

- **Botón New:** Ingresa un nuevo vehículo a la base de datos. Se pueden ingresar n vehículos. Se especifica la siguiente información.

Ilustración 61 Formulario New Vehicle

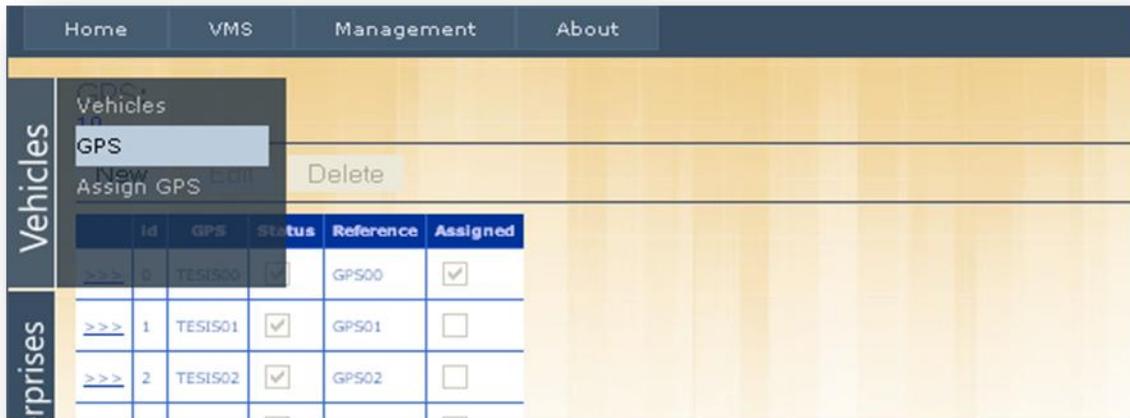
New Vehicle:

Plate: Register Date: Model: 2010
 Empowered: 2010 Series: Chassis:
 Weight (Kg): Motor:
 Enabling Date: Duration:
 Policy: Exp. Date:
 Transport Co.: Transport Insurance Co.: Insurance
 Membership Date: Exp. Date:
 Brand/Line: D.A.C. Color: "BLANCO ROJO"

- **Botón Edit:** Seleccionando un registro edita la información del vehículo.
- **Botón Delete:** Permite borrar un registro de vehículo de la base de datos.

- **GPS:** Controla y administra los GPS.

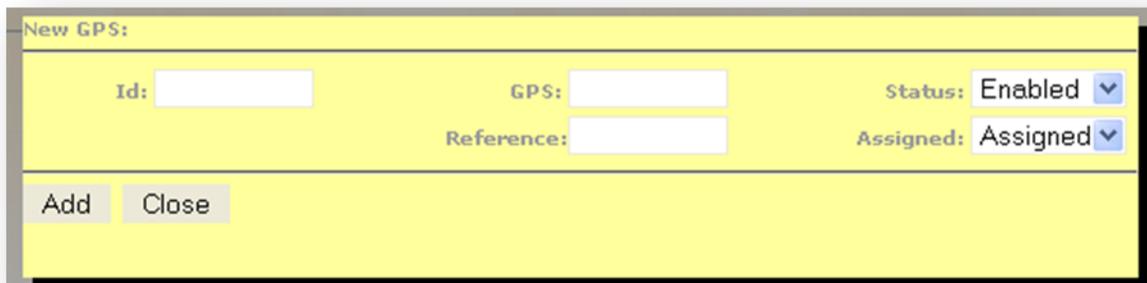
Ilustración 62 Sección GPS



Id	GPS	Status	Reference	Assigned
>>>	TESIS00	<input checked="" type="checkbox"/>	GPS00	<input checked="" type="checkbox"/>
>>>	1 TESIS01	<input checked="" type="checkbox"/>	GPS01	<input type="checkbox"/>
>>>	2 TESIS02	<input checked="" type="checkbox"/>	GPS02	<input type="checkbox"/>

- **Botón New:** Ingresa un GPS con sus características y referencias a la base de datos. Esta función se realiza en la siguiente ventana.

Ilustración 63 Formulario New GPS



New GPS:

Id: GPS: Status: Enabled ▾

Reference: Assigned: Assigned ▾

Add Close

- **Botón Edit:** Permite corregir la información de una gps seleccionada.
- **Botón Delete:** Borra cualquier registro de GPS registrada en la base de datos.
- **ASSIGN GPS:** Permite controlar la asignación de GPS a un vehículo en particular.

Ilustración 64 Sección Assign GPS

Id Vehicle	Vehicle	Id GPS	Vehicle Plate	GPS Name
0	CSM-864	0	CSM-864	TESIS00
5	DIE-001	5	DIE-001	TESIS05
9	qwer	9	qwer	TESIS09

- **Botón New:** Ingresa la relación entre una GPS y el vehículo registrado. Esta función se hace a través de la siguiente ventana.

Ilustración 65 Formulario New GPS Assignment

New GPS Assignment:

Vehicle: CSM-864 GPS: TESIS00

Add Close

- **Botón Edit:** Cambia o modifica la relación que puede tener una GPS con un vehículo registrado.
- **Botón Delete:** Borra la relación que tiene un vehículo con una GPS en específico.

6.6.2 Enterprises. Éste módulo permite realizar consultas y controles a nivel general de los buses, compañías de seguros, empresas y operarios del sistema.

- **INSURANCE COMPANY:** Permite administrar las compañías de seguros que provean sus servicios a las empresas de los buses.

Ilustración 66 Sección Insurance Company



- **Botón New:** Permite hacer el registro de una compañía de seguros.

Ilustración 67 Formulario New Insurance Company

- **Botón Edit:** Modifica o corrige la información de una compañía de seguros registrada anteriormente.
- **Botón Delete:** Elimina un registro de alguna compañía de seguros.
- **TRANSPORT COMPANY:** Administra las empresas de transporte a las que el sistema presta su servicio. Esta sección contiene las siguientes opciones.

- **Botón New:** Permite el registro de una empresa de Transportes. Este procedimiento se lleva a cabo en la siguiente ventana.

Ilustración 68 Formulario New Transport Company

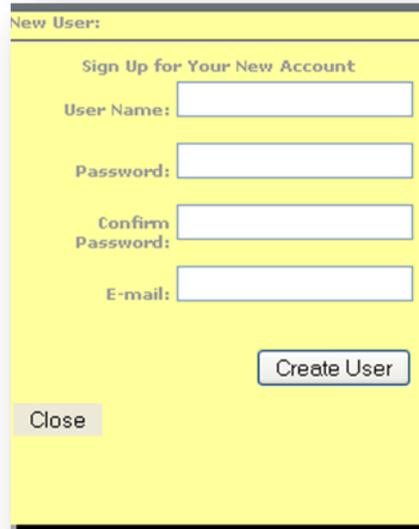
The image shows a web form titled "New Transport Company" with a yellow background. The form contains the following fields and labels:

- DNI: [input field]
- Corporate Name: [input field]
- Address: [input field]
- City: [input field]
- State: [input field]
- Country: [input field]
- Description: [input field]

At the bottom of the form, there are two buttons: "Add" and "Close".

- **Botón Edit:** Modifica la información de alguna empresa de transportes ingresada anteriormente por el administrador.
- **Botón Delete:** Elimina un registro de empresa de transportes seleccionada de la tabla de visualizaciones general.
- **USERS:** Administra los diferentes tipos de usuario y sus roles dentro del sistema. Aquí se definen si el usuario es administrador, operario y usuario registrado. Contiene las siguientes opciones.
 - **Botón New:** Registra un nuevo usuario y permite definir su rol dentro del sistema. Este procedimiento se hace en la siguiente ventana.

Ilustración 69 Formulario New Users



New User:

Sign Up for Your New Account

User Name:

Password:

Confirm Password:

E-mail:

Create User

Close

- **Botón Edit:** Modifica o corrige la información de un usuario registrado anteriormente.
- **Botón Delete:** Permite eliminar un registro de usuario registrado y seleccionado de la ventana de visualizaciones.

6.6.3 Assignment. Esta opción permite relacionar los vehículos con los usuarios y operadores registrados. Aquí se establece las relaciones que tienen las empresas de transportes con los vehículos a su cargo. Este módulo presenta las siguientes opciones.

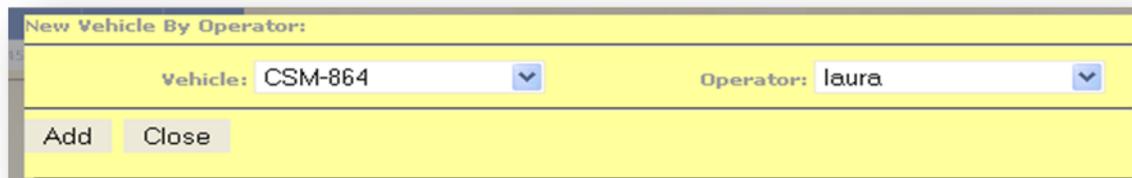
- **By Operator:** Permite administrar los registros por operador. Contiene las siguientes opciones.

Ilustración 70 Sección Assigment - By Operator



- **Botón New:** Realiza un registro a la base de datos de la relación del operador con el vehículo en específico. Este procedimiento se realiza en la siguiente ventana.

Ilustración 71 Formulario New Vehicles by Operator



- **Botón Edit:** Modifica la relación que existe entre el operador y el vehículo.
- **Botón Delete:** Elimina una relación entre el operador y un vehículo.
- **By User:** Sobre este vínculo se puede relacionar los vehículos por usuario. Contiene las siguientes opciones.

Ilustración 72 Sección Vehicles by User

Id	User	GPS	Plate
c167db7-9f53-418f-8a0c-c749faed51bd	15edba14-e196-42b6-8c77-7c50360159dc	TESIG00	CSM-864
00d429d2-a971-4680-8d09-a3c3af94c11	15edba14-e196-42b6-8c77-7c50360159dc	TESIG09	qeer

- **Botón New:** Permite crear una relación entre un usuario registrado y un vehículo. Este procedimiento se puede realizar en la siguiente ventana.

Ilustración 73 Formulario New Vehicle by Operator

Vehicle: CSM-864 Operator: user

Add Close

- **Botón Edit:** Con ésta opción se puede modificar la relación entre usuario registrado y vehículo.
- **Botón Delete:** Suprime la relación entre usuario registrado y vehículo.

CAPITULO 7 IMPLEMENTACIÓN

7.1 INTRODUCCIÓN

Para implementar la plataforma web, se utilizó la Visual Studio 2010 bajo el lenguaje de programación C# y JavaScript. Como navegadores se utilizaron Mozilla Firefox, Internet Explorer 8, Internet Explorer 9 y Google Chrome para realizar las pruebas de compatibilidad.

Ilustración 74 Plataforma



La plataforma no depende del sistema operativo debido a su modo de servicio sobre la web. Sobre la base de datos se realiza la implementación de los servicios de minería de datos y se construye el Data Warehouse.

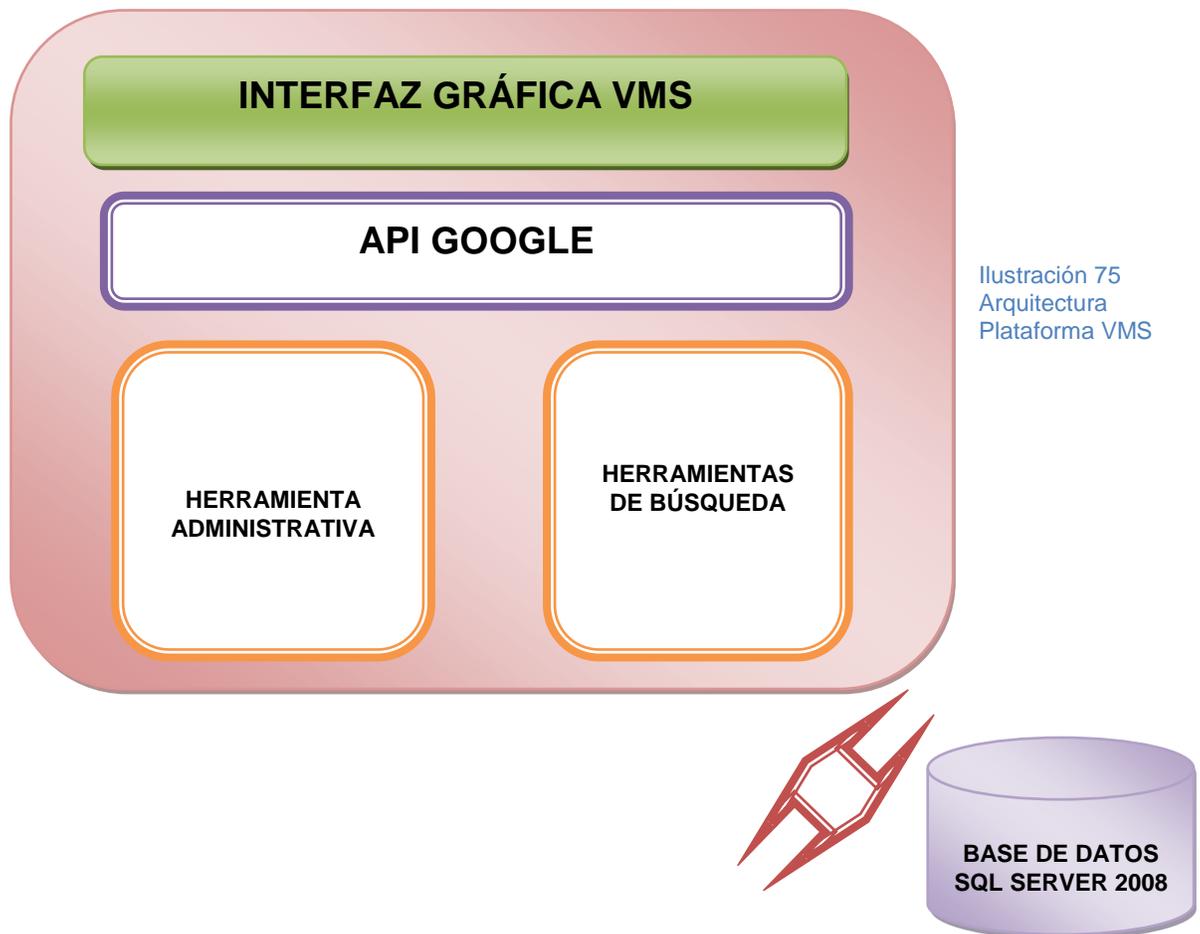
Los métodos almacenados realizados en la programación de la plataforma optimizan la respuesta a los eventos que el usuario exija. De la misma forma estos métodos contribuyen a la inserción de registros en las tablas de la base de datos y por ende minimizan el consumo de recursos en el servidor.

Todos los servicios web se encuentran en un servidor ubicado en la Universidad de Nariño. Este servidor cuenta con las siguientes características:

- **Sistema Operativo:** Windows Server 2008 R2 Enterprise Version 6.1.7600 Build 7600.
- **Modelo:** Hp Compaq dc7900 Small Form Factor System Type x64-based
- **Procesador:** Core 2 Quad CPU Q8400 2.66GHz
- **Memoria Ram:** 4 Gb DDR3
- **Disco Duro:** 500 Gb Hitachi Deskstar P7K500
- **Red:** 1 red Giga bit Ethernet

Los datos llegan desde las GPS y son reconocidos como una trama. Pueden ser de dos tipos.

7.2 ARQUITECTURA PLATAFORMA VMS.



Una vez instalado el dispositivo de captura de datos, la GPS empieza a entramar los eventos cada 3 segundos en movimiento. Cuando se detecta que el vehículo permanece quieto, el censado de datos se lleva cada minuto. De esta forma se minimiza los eventos y por lo tanto el costo en bits de datos.

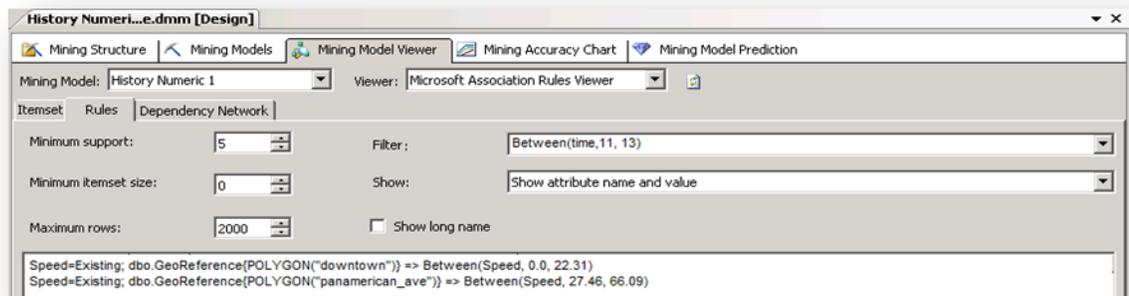
El socket programado recibe los datos y los almacena en las tablas para ser consumidos por el aplicativo web.

Una vez los datos se encuentren en las tablas de la base de datos, el aplicativo puede generar información de todo tipo, según sea requerida por todos los diferentes tipos de usuario.

8. PRUEBAS Y RESULTADOS

Al aplicar los algoritmos de minería de datos MICROSOFT TIME SERIES y MICROSOFT ASSOCIATION RULES se obtuvieron los siguientes resultados:

Ilustración 76 Resultados algoritmo de minería Microsoft Association Rules



Al definir las geocercas de la siguiente forma:

Geocerca downtown = POLYGON ("downtown") con los parámetros de Google Maps

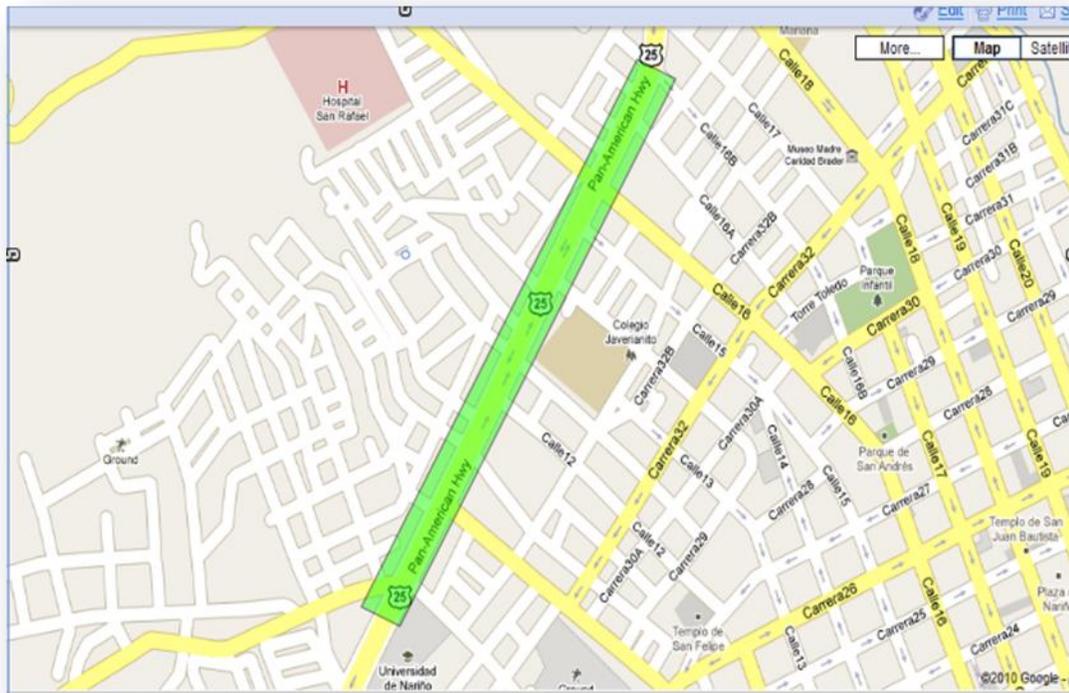
Ilustración 77 Trazo de poligonos (Geocerca) sobre Google Maps - Downtown



Se encontró que el promedio de velocidad de los vehículos monitoreados oscilan entre 0.0 y 22.31 km/h entre las 11am y la 1pm.

En cambio la geocerca panamerican_ave = POLYGON (“panamerican_ave”) con los parámetros de Google Maps.

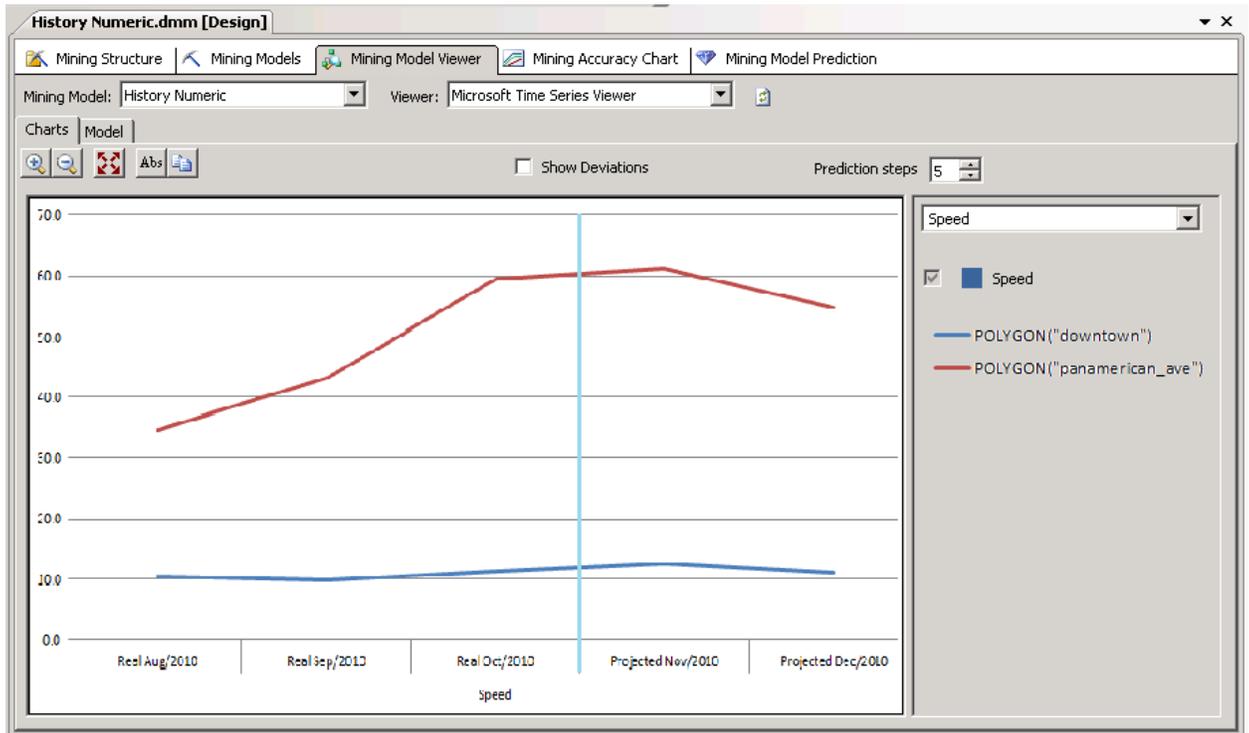
Ilustración 78 Trazo de poligonos (Geocerca) sobre Google Maps – Panamericana



Se encontró que el promedio de velocidad de los vehículos monitoreados oscilan entre 27.46 y 66.09 km/h entre las 11am y la 1pm.

Por otra parte el algoritmo MICROSOFT TIME SERIES obtuvo los siguientes resultados:

Ilustración 79 Resultados algoritmo de minería Microsoft Time Series



Mostrando las predicciones para noviembre y diciembre del presente año. Haciendo la lectura de la gráfica se obtienen los valores de las dos geocercas downtown y panamerican_ave. Los datos monitoreados de agosto, septiembre y octubre ayudan a soportar los datos predichos para noviembre y diciembre en cuanto a promedios de velocidad de los vehículos monitoreados.

Los resultados de las velocidades promedio esperadas para los vehículos monitoreados son:

Tabla 27 Velocidades promedio esperadas

	Projected Nov/2010	Projected Dec/2010
POLYGON("downtown")	12.4 km/h	11.1 km/h
POLYGON("panamerican_ave")	61.2 km/h	54.8 km/h

8.1 REPORTE

Todos los datos almacenados por la base de datos permiten consolidar informes que ayuden a la toma de decisiones. Es con este módulo que los operadores del sistema pueden consolidar resultados con el fin de optimizar los tiempos, rutas y horarios con los que el sistema de transporte urbano funciona. En la figura 80 se puede ver un reporte generado con los datos de velocidades de cada bus. Son muchos los datos de velocidad por día que se toman de cada bus (aproximadamente 26282 por cada uno), por lo que para ahorrar espacio y mostrar información más útil se promedia las velocidades por día de cada bus.

Velocidades promedio por día

Siguiente

1 of 1 Find | Next

Placas

Fechas	SDP367	SDP461	SDP587	SDP801	SDP837	SJP236	SJP301
2010/10/25	22,61	24,16	20,61	21,91	22,06	17,60	20,15
2010/10/26	20,84	21,51	21,21	21,29	21,49	20,76	20,80
2010/10/27	21,18	22,11	21,21	21,34	21,91	21,53	21,01
2010/10/28	21,56	21,07	20,39	21,67	21,50	20,73	21,07
2010/10/29	21,82	22,04	21,77	21,30	22,09	22,61	21,79
2010/10/30		22,59	21,17	21,14	21,99	22,67	22,36
2010/10/31		20,55	20,98	20,40	19,41	20,02	19,96
2010/11/02	17,42	17,98	17,10	17,89	19,79	18,29	18,17
2010/11/03	18,58	19,40	19,11	17,96	18,18	18,25	18,49
2010/11/04	17,86	18,79	17,97	19,28	18,24	18,56	18,49
2010/11/05	18,11	18,06	17,53	18,03	18,19	18,67	18,54
2010/11/06	18,02	18,58	17,53	17,59	18,20	18,52	18,50
2010/11/07			18,36	18,13	18,26	16,57	19,71
2010/11/08	17,96	17,65	17,63	17,72	18,29	18,22	19,33
2010/11/09	17,80	17,84	18,39	17,59	17,89	18,43	18,48
2010/11/10	17,09	18,08	18,54	17,38	17,92	18,41	18,48
Promedio total	19,30	20,03	19,34	19,41	19,71	19,37	19,71

Ilustración 80 – Promedios de velocidad por día y bus

Además se puede generar el anterior reporte como un gráfico de barras que facilita la comprensión y lectura de los anteriores datos (véase figura 81).

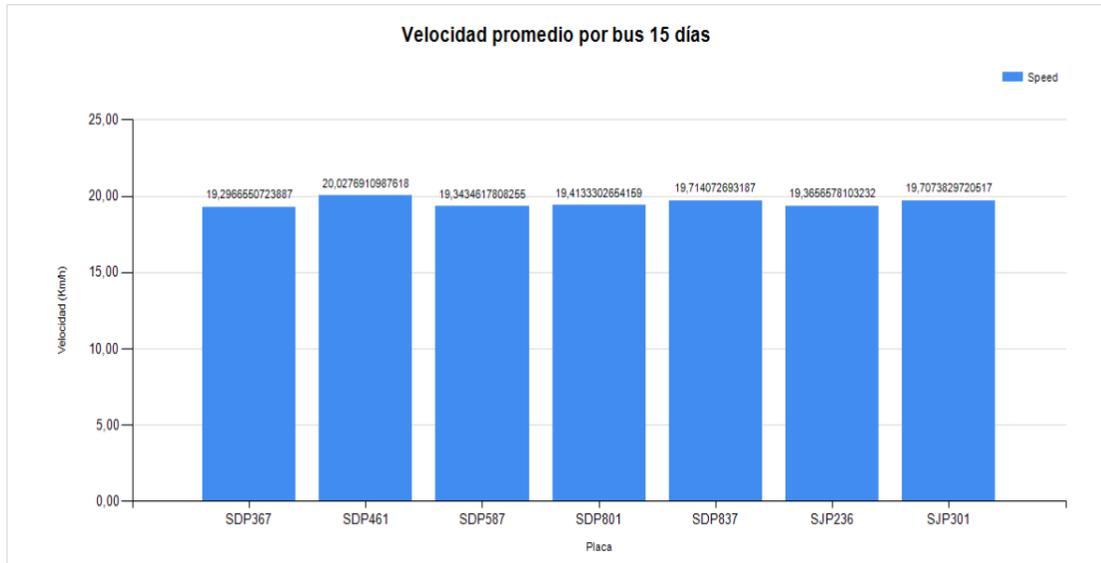


Ilustración 81 – Gráfico de barras de los promedios de velocidad por día de cada bus

Un segundo reporte hace uso de los datos de velocidad y mediante algoritmos busca la mayor velocidad registrada en el vehículo, la velocidad promedio y la mínima velocidad registrada; en este caso para el bus de placas SDP-587.



Bus de placas SDP587

Ilustración 82 – Reporte gráfico de la velocidad máxima, velocidad promedio y velocidad mínima del bus de placas SDP 587

9. CONCLUSIONES

La plataforma web utiliza los servicios internacionales Mashup's como Google Maps® a través de una infraestructura con un GPS instalado en vehículos de transporte público o cualquier tipo de vehículo.

Los eventos reportados por el modem GPS/ GPRS instalada ayudan a determinar resultados debido a que los datos almacenados contienen información como distancia, combustible consumido, velocidades promedio, puntos de inactividad del vehículo, tiempos parciales y totales por cada uno de los buses, etc.

La manipulación grafica de los datos ayuda a que los usuarios obtengan información concisa y fácil de asimilar. Esto es muy importante si tratamos con diferentes tipos de usuario ya que se manejan diferentes roles dentro de las sesiones en el servidor.

Utilizar reportes en este tipo de aplicaciones, donde los datos se obtienen en grandes cantidades, ayuda al analista a interpretar y a filtrar la información necesaria con el fin de tomar decisiones. Estos reportes ajustados al tiempo de captura de datos enmarcan ejemplos que pueden solucionar problemas como el tráfico a determinadas horas del día, reducción de gases, optimización de tiempos y demás variables aplicables al tráfico en una ciudad.

La herramienta web se integra completamente con el hardware de posicionamiento GPS/ GPRS, estableciendo así un producto que satisface las necesidades de los usuarios finales. La implementación en niveles de usuario permite que los servicios del aplicativo sean consumidos de forma segura.

El dispositivo GPS/ GPRS y la base de datos implementada en SQL Server 2008® se acoplan funcionalmente a través del dispositivo socket que permite la recepción de las tramas de la GPS/ GPRS y de comunicación con las tablas en la base de datos. Esto permite que el producto sea escalable a diferentes niveles y a diferentes servicios.

La bodega de Datos (Data Warehouse) permite que la información censada se contenga de forma íntegra y que sobre ella se pueda realizar un análisis inteligente a través de funciones programadas dentro del aplicativo web.

Las aplicaciones desarrolladas bajo una filosofía web, permiten que puedan ser consumidas desde cualquier parte del mundo. Con esto se puede lograr una cobertura mucho más amplia sin necesidad de realizar una mayor inversión en equipos. Cualquier dispositivo que soporte un navegador y mínimas tecnologías puede consumir los recursos que la plataforma ofrece.

Todos los datos obtenidos han sido censados en tiempo real, lo que permite que la información producida sobre ellos sea utilizable en el momento de realizar pronósticos y gráficas.

El sistema inicialmente fue probado en vehículos de uso personal, como automóviles, lo que conlleva a que el ambiente de impacto sea real, y que los datos obtenidos puedan ser utilizados para consolidar información importante.

10. RECOMENDACIONES

- Utilizar el sistema VMS como apoyo a la toma de decisiones en entidades de control de Tránsito.
- Implementar la herramienta a otro nivel con el fin de obtener y consolidar información importante a gran escala para realizar proyecciones multinivel.
- Realizar versiones posteriores con el fin de adaptar el aplicativo a varios entornos públicos y privados.

11. BIBLIOGRAFIA

- [1] Fernández Carlos Molina, Imprecisión e Incertidumbre en el Modelo de Datos Multidimensional: Aplicación a la Minería de Datos, Editorial Universidad de Granada, Granada, España, 2005.
-

- [2] Zvenger Patricia Andrea, Introducción al Soporte de Decisiones: Incorporación de soluciones OLAP en entornos empresariales, Universidad Nacional del Sur, 2005.
-

11.1 REFERENCIAS BIBLIOGRÁFICAS

- CASCETTA, E. Transportation Systems Analysis: Models and Applications, 2/e (Springer Optimization and Its applications). ISBN: 978-0262632898. MIT Press. 2008.
- es.wikipedia.org/wiki/Mashup_%28aplicaci%C3%B3n_web_h%C3%ADbrida%29
- es.wikipedia.org/wiki/JavaScript
- es.wikipedia.org/wiki/Microsoft_Visual_Studio
- es.wikipedia.org/wiki/Microsoft_SQL_Server
- es.wikipedia.org/wiki/Programaci%C3%B3n_extrema
- http://es.wikipedia.org/wiki/Diagrama_de_clases

- HABERMAN, R. Mathematical Models: Mechanical Vibrations, Population Dynamics, and Traffic Flow (Classic in Applied Mathematics). ISBN: 0-89871-408-7. New Jersey. 1998.

- MAYINGER, F. Mobility and Traffic in the 21st Century. Germany. ISBN: 3-540-41612-9. 2001.
- TOLEDO, B. A. et al. Tráfico vehicular como sistema complejo. Bogotá (Colombia) – Santiago (Chile). Departamento de Física de las universidades de Chile y la Universidad de los Andes, 2005.
- www.sinnexus.com/business_intelligence/datawarehouse.aspx
- www.sinnexus.com/business_intelligence/
- www.sparxsystems.com.ar/download/ayuda/index.html?packagediagram.htm
- www.ia.uned.es/ia/asignaturas/adms/GuiaDidADMS/node61.html

ANEXOS

Anexo 1. Marcas Registradas Utilizadas

- SQL Server 2008, Windows Vista Ultimate y Visual Studio 2010 son marcas registradas de Microsoft Corporation.
- Google Maps ™ es una marca registrada de Google.

Anexo 2. Script Base de Datos.

```
USE [master]
GO
/***** Object: Database [vms]  Script Date: 10/05/2010 18:44:07 *****/
CREATE DATABASE [vms] ON PRIMARY
( NAME = N'vms', FILENAME = N'D:\Program Files\Microsoft SQL
Server\MSSQL10_50.SQL_SYSTEMDEPT\MSSQL\DATA\vms.mdf' , SIZE =
7168KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )
LOG ON
( NAME = N'vms_log', FILENAME = N'D:\Program Files\Microsoft SQL
Server\MSSQL10_50.SQL_SYSTEMDEPT\MSSQL\DATA\vms_log.ldf' , SIZE =
4672KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)
GO
ALTER DATABASE [vms] SET COMPATIBILITY_LEVEL = 100
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [vms].[dbo].[sp_fulltext_database] @action = 'enable'
end
GO
ALTER DATABASE [vms] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [vms] SET ANSI_NULLS OFF
GO
```

```
ALTER DATABASE [vms] SET ANSI_PADDING OFF
GO
ALTER DATABASE [vms] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [vms] SET ARITHABORT OFF
GO
ALTER DATABASE [vms] SET AUTO_CLOSE OFF
GO
ALTER DATABASE [vms] SET AUTO_CREATE_STATISTICS ON
GO
ALTER DATABASE [vms] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [vms] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [vms] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [vms] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [vms] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [vms] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [vms] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [vms] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [vms] SET DISABLE_BROKER
GO
ALTER DATABASE [vms] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [vms] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [vms] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [vms] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [vms] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [vms] SET READ_COMMITTED_SNAPSHOT OFF
```

```

GO
ALTER DATABASE [vms] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [vms] SET READ_WRITE
GO
ALTER DATABASE [vms] SET RECOVERY FULL
GO
ALTER DATABASE [vms] SET MULTI_USER
GO
ALTER DATABASE [vms] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [vms] SET DB_CHAINING OFF
GO
EXEC sys.sp_db_vardecimal_storage_format N'vms', N'ON'
GO
USE [vms]
GO
/***** Object: User [John]  Script Date: 10/05/2010 18:44:07 *****/
CREATE USER [John] FOR LOGIN [SYSTEMDEPT\John] WITH
DEFAULT_SCHEMA=[dbo]
GO
/***** Object: User [IUSR_ANONYMOUS]  Script Date: 10/05/2010 18:44:07
*****/
CREATE USER [IUSR_ANONYMOUS] WITH DEFAULT_SCHEMA=[dbo]
GO
/***** Object: User [IUSR]  Script Date: 10/05/2010 18:44:07 *****/
CREATE USER [IUSR] FOR LOGIN [NT AUTHORITY\IUSR]
GO
/***** Object: User [Carlos]  Script Date: 10/05/2010 18:44:07 *****/
CREATE USER [Carlos] FOR LOGIN [SYSTEMDEPT\Carlos] WITH
DEFAULT_SCHEMA=[dbo]
GO
/***** Object: User [Adrian]  Script Date: 10/05/2010 18:44:07 *****/
CREATE USER [Adrian] FOR LOGIN [SYSTEMDEPT\Adrian] WITH
DEFAULT_SCHEMA=[dbo]
GO
/***** Object: Table [dbo].[LastData]  Script Date: 10/05/2010 18:44:07 *****/
SET ANSI_NULLS ON
GO

```

```

SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[LastData](
    [GPSname] [nvarchar](7) NOT NULL,
    [LastGeoReference] [geography] NOT NULL,
    [LastUTCDate] [nvarchar](6) NOT NULL,
    [LastUTCTime] [nvarchar](6) NOT NULL,
    CONSTRAINT [PK_LastData] PRIMARY KEY CLUSTERED
(
    [GPSname] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Insurance_Company]   Script Date: 10/05/2010
18:44:07 *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Insurance_Company](
    [id] [uniqueidentifier] NOT NULL,
    [DNI] [nvarchar](50) NOT NULL,
    [CorporateName] [nvarchar](250) NOT NULL,
    [Address] [nvarchar](250) NOT NULL,
    [City] [nvarchar](250) NOT NULL,
    [Country] [nvarchar](250) NOT NULL,
    CONSTRAINT [PK_Insurance_Company] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[MessageType]   Script Date: 10/05/2010 18:44:07
*****/
SET ANSI_NULLS ON

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[MessageType](
    [id] [int] NOT NULL,
    [MessageType] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_MessageType] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Transport_Company]    Script Date: 10/05/2010
18:44:07 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Transport_Company](
    [id] [uniqueidentifier] NOT NULL,
    [DNI] [nvarchar](50) NOT NULL,
    [CorporateName] [nvarchar](250) NOT NULL,
    [Address] [nvarchar](250) NULL,
    [City] [nvarchar](100) NULL,
    [State] [nvarchar](100) NULL,
    [Country] [nvarchar](100) NULL,
    [Description] [nvarchar](250) NULL,
    PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[SpatialTable]    Script Date: 10/05/2010 18:44:07 *****/
SET ANSI_NULLS ON

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[SpatialTable](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [Name] [nvarchar](50) NULL,
    [GeogCol1] [geography] NULL,
    [GeogCol2] AS ([GeogCol1].[STAsText]()),
    CONSTRAINT [PK_SpatialTable] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: StoredProcedure [dbo].[NewSettings]  Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[NewSettings]
    @id int,
    @Settings nvarchar(50),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO Settings VALUES(@id, @Settings);
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'EL CODIGO YA EXISTE';
    END CATCH
END
GO
/***** Object: Table [dbo].[RMC]  Script Date: 10/05/2010 18:44:08 *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[RMC](
    [id] [uniqueidentifier] NOT NULL,
    [GPSname] [nvarchar](17) NOT NULL,
    [UTCTime] [nvarchar](6) NOT NULL,
    [Flag] [nvarchar](1) NOT NULL,
    [GeoReference] [geography] NOT NULL,
    [Speed] [nvarchar](10) NOT NULL,
    [Course] [nvarchar](10) NOT NULL,
    [UTCDate] [nvarchar](10) NOT NULL,
    [MagneticVariation] [nvarchar](20) NOT NULL,
    [Checksum] [nvarchar](10) NOT NULL,
    CONSTRAINT [PK_RMC] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Color]    Script Date: 10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Color](
    [id] [int] NOT NULL,
    [Description] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_TB_ColorVehiculo] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO

```

```

/***** Object: Table [dbo].[CM]   Script Date: 10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[CM](
    [id] [uniqueidentifier] NOT NULL,
    [GPSname] [nvarchar](17) NOT NULL,
    [GeoReference] [geography] NOT NULL,
    [UTCDate] [nvarchar](10) NOT NULL,
    [UTCTime] [nvarchar](6) NOT NULL,
    [Passengers] [nvarchar](10) NOT NULL,
    [CO2] [nvarchar](10) NOT NULL,
    [Fuel] [nvarchar](10) NOT NULL,
    CONSTRAINT [PK__CM__3213E83F6CD828CA] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[CallReport]   Script Date: 10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[CallReport](
    [id] [int] NOT NULL,
    [Plate] [nvarchar](50) NOT NULL,
    [Date] [datetime] NOT NULL,
    [Source] [nvarchar](250) NOT NULL,
    [Destination] [nvarchar](250) NOT NULL,
    [Motive] [nvarchar](50) NOT NULL,
    [Response] [nvarchar](1000) NOT NULL,
    [Observation] [nvarchar](1000) NULL,
    CONSTRAINT [PK_TB_ReporteLlamada] PRIMARY KEY CLUSTERED
(
    [id] ASC

```

```

)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Brand]   Script Date: 10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Brand](
    [id] [nvarchar](2) NOT NULL,
    [Description] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_TB_Brand] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[GPS]   Script Date: 10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[GPS](
    [id] [int] NOT NULL,
    [GPS] [nvarchar](50) NOT NULL,
    [Status] [bit] NOT NULL,
    [Reference] [nvarchar](50) NULL,
    [IsAssigned] [bit] NOT NULL,
    CONSTRAINT [PK_TB_GPS] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

```

```

GO
/***** Object: Table [dbo].[Event]  Script Date: 10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Event](
    [id] [int] NOT NULL,
    [Event] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_Event] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[DocType]  Script Date: 10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[DocType](
    [id] [int] NOT NULL,
    [DocType] [nchar](10) NOT NULL,
    [Description] [nvarchar](50) NULL,
    CONSTRAINT [PK_TB_TipoDocumento] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: StoredProcedure [dbo].[DeleteSettings]  Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

GO
CREATE PROCEDURE [dbo].[DeleteSettings]
    @id int,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        DELETE FROM Settings WHERE id=@id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 547
            SET @Result = 'EXISTEN DEPENDENCIAS!!!';
    END CATCH
END
GO
/***** Object: UserDefinedFunction [dbo].[GetGPS]  Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[GetGPS]()
RETURNS TABLE
AS
RETURN
(
SELECT [id] as f01
    ,[GPS] as f02
    ,[Status] as f03
    ,[Reference] as f04
    ,[IsAssigned] as f05
FROM [vms].[dbo].[GPS]
)
GO
/***** Object: UserDefinedFunction [dbo].[GetColors]  Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON

```

```

GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[GetColors]()
RETURNS TABLE
AS
RETURN
(
SELECT [id] as f01
      ,[Description] as f02
FROM [vms].[dbo].[Color]
)
GO
/***** Object: UserDefinedFunction [dbo].[GetBrands]   Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[GetBrands]()
RETURNS TABLE
AS
RETURN
(
SELECT [id] as f01
      ,[Description] as f02
FROM [vms].[dbo].[Brand]
)
GO
/***** Object: StoredProcedure [dbo].[DeleteMessageType]   Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DeleteMessageType]
    @id int,
    @Result nvarchar(50) OUTPUT
AS

```

```

BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        DELETE FROM MessageType WHERE id=@id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 547
            SET @Result = 'EXISTEN DEPENDENCIAS!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[EditTransportCompany]   Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[EditTransportCompany]
    @id uniqueidentifier,
    @DNI nvarchar(50),
    @CorporateName nvarchar(250),
    @Address nvarchar(250),
    @City nvarchar(100),
    @State nvarchar(100),
    @Country nvarchar(100),
    @Description nvarchar(250),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        UPDATE Transport_Company SET
        DNI=@DNI,CorporateName=@CorporateName, [Address]=@Address,
        City=@City, [State]=@State,Country=@Country, [Description]=@Description
        WHERE id = @id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH

```

```

                SET @Result = 'NO EXISTE EL REGISTRO A MODIFICAR!!!';
            END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[EditMessageType]  Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[EditMessageType]
    @id int,
    @MessageType nvarchar(50),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        UPDATE MessageType SET MessageType=@MessageType
WHERE id = @id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'NO EXISTE EL REGISTRO A MODIFICAR!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[EditInsuranceCompany]  Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[EditInsuranceCompany]
    @id uniqueidentifier,
    @DNI nvarchar(50),
    @CorporateName nvarchar(250),
    @Address nvarchar(250),
    @City nvarchar(250),

```

```

        @Country nvarchar(250),
        @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        UPDATE Insurance_Company SET DNI=@DNI,
CorporateName=@CorporateName, [Address]=@Address, City=@City,
Country=@Country WHERE id = @id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'NO EXISTE EL REGISTRO A MODIFICAR!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[EditGPS]   Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[EditGPS]
    @id int,
    @GPS nvarchar(50),
    @Status bit,
    @Reference nvarchar(50),
    @IsAssigned bit,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        UPDATE GPS SET GPS=@GPS, [Status]=@Status,
Reference=@Reference, IsAssigned=@IsAssigned WHERE id = @id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'NO EXISTE EL REGISTRO A MODIFICAR!!!';
    END CATCH
END

```

```

        END CATCH
    END
    GO
    /***** Object: StoredProcedure [dbo].[EditEvent]   Script Date: 10/05/2010
    18:44:08 *****/
    SET ANSI_NULLS ON
    GO
    SET QUOTED_IDENTIFIER ON
    GO
    CREATE PROCEDURE [dbo].[EditEvent]
        @id int,
        @Event nvarchar(50),
        @Result nvarchar(50) OUTPUT
    AS
    BEGIN
        SET NOCOUNT ON;
        BEGIN TRY
            UPDATE [Event] SET [Event]=@Event WHERE id = @id;
            SET @Result = 'OK';
        END TRY
        BEGIN CATCH
            SET @Result = 'NO EXISTE EL REGISTRO A MODIFICAR!!!';
        END CATCH
    END
    GO
    /***** Object: StoredProcedure [dbo].[EditDocType]   Script Date: 10/05/2010
    18:44:08 *****/
    SET ANSI_NULLS ON
    GO
    SET QUOTED_IDENTIFIER ON
    GO
    CREATE PROCEDURE [dbo].[EditDocType]
        @id int,
        @DocType nchar(10),
        @Description nvarchar(50),
        @Result nvarchar(50) OUTPUT
    AS
    BEGIN
        SET NOCOUNT ON;

```

```

        BEGIN TRY
            UPDATE DocType SET DocType=@DocType, [Description]=
@Description WHERE id = @id;
            SET @Result = 'OK';
        END TRY
        BEGIN CATCH
            SET @Result = 'NO EXISTE EL REGISTRO A MODIFICAR!!!';
        END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[EditColor]   Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[EditColor]
    @id int,
    @Description nvarchar(50),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        UPDATE Color SET [Description] = @Description WHERE id = @id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'NO EXISTE EL REGISTRO A MODIFICAR!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[EditCallReport]   Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[EditCallReport]

```

```

    @id int,
    @Plate nvarchar(50),
    @Date datetime,
    @Source nvarchar(250),
    @Destination nvarchar(250),
    @Motive nvarchar(50),
    @Response nvarchar(1000),
    @Observation nvarchar(1000),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        UPDATE CallReport SET [Plate]=@Plate, [Date]=@Date,
[Source]=@Source, [Destination]=@Destination, [Motive]=@Motive,
[Response]=@Response, [Observation]=@Observation WHERE id = @id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'NO EXISTE EL REGISTRO A MODIFICAR!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[EditBrand]  Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[EditBrand]
    @id nvarchar(2),
    @Description nvarchar(50),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        UPDATE Brand SET [Description] = @Description WHERE id = @id
        SET @Result = 'OK';
    
```

```

        END TRY
        BEGIN CATCH
            SET @Result = 'NO EXISTE EL REGISTRO A MODIFICAR!!!';
        END CATCH
    END
GO
/***** Object: StoredProcedure [dbo].[DeleteGPS]   Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DeleteGPS]
    @id int,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        DELETE FROM GPS WHERE id=@id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 547
            SET @Result = 'EXISTEN DEPENDENCIAS!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[DeleteEvent]   Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DeleteEvent]
    @id int,
    @Result nvarchar(50) OUTPUT
AS
BEGIN

```

```

SET NOCOUNT ON;
BEGIN TRY
    DELETE FROM [Event] WHERE id=@id;
    SET @Result = 'OK';
END TRY
BEGIN CATCH
    IF ERROR_NUMBER() = 547
        SET @Result = 'EXISTEN DEPENDENCIAS!!!';
END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[DeleteDocType]  Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DeleteDocType]
    @id int,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        DELETE FROM DocType WHERE id=@id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 547
            SET @Result = 'EXISTEN DEPENDENCIAS!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[DeleteColor]  Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

CREATE PROCEDURE [dbo].[DeleteColor]
    @id int,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        DELETE FROM Color WHERE id=@id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 547
            SET @Result = 'EXISTEN DEPENDENCIAS!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[DeleteCallReport]   Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DeleteCallReport]
    @id int,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        DELETE FROM CallReport WHERE id=@id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 547
            SET @Result = 'EXISTEN DEPENDENCIAS!!!';
    END CATCH
END
GO

```

/***** Object: StoredProcedure [dbo].[DeleteBrand] Script Date: 10/05/2010
18:44:08 *****/

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DeleteBrand]
    @id nvarchar(2),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        DELETE FROM Brand WHERE id=@id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 547
            SET @Result = 'EXISTEN DEPENDENCIAS!!!';
    END CATCH
END
GO
```

/***** Object: StoredProcedure [dbo].[DeleteInsuranceCompany] Script Date:
10/05/2010 18:44:08 *****/

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DeleteInsuranceCompany]
    @id uniqueidentifier,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        DELETE FROM Insurance_Company WHERE id=@id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
```

```

                IF ERROR_NUMBER() = 547
                    SET @Result = 'EXISTEN DEPENDENCIAS!!!';
            END CATCH
END
GO
/***** Object: Table [dbo].[VehicleByUser]   Script Date: 10/05/2010 18:44:08
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[VehicleByUser](
    [id] [uniqueidentifier] NOT NULL,
    [idUser] [uniqueidentifier] NOT NULL,
    [GPS] [int] NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[VehicleByOperator]   Script Date: 10/05/2010 18:44:08
*****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[VehicleByOperator](
    [id] [uniqueidentifier] NOT NULL,
    [idUser] [uniqueidentifier] NOT NULL,
    [GPS] [int] NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]

```

```

) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Vehicle]   Script Date: 10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Vehicle](
    [id] [uniqueidentifier] NOT NULL,
    [Plate] [nchar](10) NOT NULL,
    [InnerRegisterDate] [nvarchar](10) NOT NULL,
    [RegisterDate] [nvarchar](10) NOT NULL,
    [Model] [nchar](10) NOT NULL,
    [Empowered] [nchar](10) NOT NULL,
    [Series] [nvarchar](50) NULL,
    [Chassis] [nvarchar](50) NULL,
    [Weight] [nchar](10) NULL,
    [Motor] [nvarchar](50) NULL,
    [EnablingDate] [nvarchar](10) NULL,
    [EnablingDuration] [int] NULL,
    [Policy] [nvarchar](10) NULL,
    [PolicyExpirationDate] [nvarchar](10) NULL,
    [idTransport_Company] [uniqueidentifier] NULL,
    [idInsurance_Company] [uniqueidentifier] NULL,
    [MembershipDate] [nvarchar](10) NULL,
    [ExpirationDate] [nvarchar](10) NULL,
    [idBrand] [nvarchar](2) NULL,
    [idLine] [nvarchar](3) NULL,
    [idColor] [int] NULL,
PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: StoredProcedure [dbo].[UpdateLastData]   Script Date: 10/05/2010
18:44:08 *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[UpdateLastData]
    @GPSname nvarchar(17),
    @Latitude nvarchar(20),
    @Longitude nvarchar(20),
    @UTCDate nvarchar(6),
    @UTCTime nvarchar(6),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        UPDATE LastData SET
            LastGeoReference=geography::STGeomFromText('POINT(' +
@Latitude + ' ' + @Longitude + ')', 4326),
            LastUTCDate=@UTCDate, LastUTCTime=@UTCTime
WHERE GPSname=@GPSname;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'Oops!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[NewInsuranceCompany]  Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[NewInsuranceCompany]
    @DNI nvarchar(50),
    @CorporateName nvarchar(250),
    @Address nvarchar(250),
    @City nvarchar(250),
    @Country nvarchar(250),

```

```

        @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO Insurance_Company VALUES(NEWID(), @DNI,
        @CorporateName, @Address, @City, @Country);
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'EL CODIGO YA EXISTE';
    END CATCH
END
GO
/***** Object: UserDefinedFunction [dbo].[GetInsuranceCompanies]  Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[GetInsuranceCompanies]()
RETURNS TABLE
AS
RETURN
(
SELECT [id] as f01
    ,[DNI] as f02
    ,[CorporateName] as f03
    ,[Address] as f04
    ,[City] as f05
    ,[Country] as f06
FROM [vms].[dbo].[Insurance_Company]
)
GO
/***** Object: UserDefinedFunction [dbo].[GetTransportCompanies]  Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

GO
CREATE FUNCTION [dbo].[GetTransportCompanies]()
RETURNS TABLE
AS
RETURN
(
SELECT [id] as f01
      ,[DNI] as f02
      ,[CorporateName] as f03
      ,[Address] as f04
      ,[City] as f05
      ,[State] as f06
      ,[Country] as f07
      ,[Description] as f08
FROM [vms].[dbo].[Transport_Company]
)
GO
/***** Object: Table [dbo].[Line]  Script Date: 10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Line](
      [idBrand] [nvarchar](2) NOT NULL,
      [id] [nvarchar](3) NOT NULL,
      [Description] [nvarchar](50) NOT NULL
) ON [PRIMARY]
GO
/***** Object: StoredProcedure [dbo].[NewGPS]  Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[NewGPS]
      @id int,
      @GPS nvarchar(50),
      @Status bit,
      @Reference nvarchar(50),

```

```

        @IsAssigned bit,
        @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO GPS VALUES(@id, @GPS, @Status, @Reference,
@IsAssigned);
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'EL CODIGO YA EXISTE';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[NewEvent]   Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[NewEvent]
    @id int,
    @Event nvarchar(50),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO [Event] VALUES(@id, @Event);
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'EL CODIGO YA EXISTE';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[NewDocType]   Script Date: 10/05/2010
18:44:08 *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[NewDocType]
    @id int,
    @DocType nchar(10),
    @Description nvarchar(50),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO DocType VALUES(@id, @DocType, @Description);
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'EL CODIGO YA EXISTE';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[NewColor]  Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[NewColor]
    @id int,
    @Description nvarchar(50),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO Color VALUES(@id, @Description);
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH

```

```

                SET @Result = 'EL CODIGO YA EXISTE!!!';
            END CATCH
        END
    GO
/***** Object: StoredProcedure [dbo].[NewCM]   Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[NewCM]
    @GPSname nvarchar(17),
    @Latitude nvarchar(20),
    @Longitude nvarchar(20),
    @UTCDate nvarchar(10),
    @UTCTime nvarchar(6),
    @Passengers nvarchar(10),
    @CO2 nvarchar(10),
    @Fuel nvarchar(10),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO CM VALUES(NEWID(), @GPSname,
geography::STGeomFromText('POINT(' + @Latitude + ' ' + @Longitude + ')',
4326), @UTCDate, @UTCTime, @Passengers, @CO2, @Fuel);
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'Oops!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[NewCallReport]   Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON

```

```

GO
CREATE PROCEDURE [dbo].[NewCallReport]
    @id int,
    @Plate nvarchar(50),
    @Date datetime,
    @Source nvarchar(250),
    @Destination nvarchar(250),
    @Motive nvarchar(50),
    @Response nvarchar(1000),
    @Observation nvarchar(1000),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO CallReport VALUES(@id, @Plate, @Date, @Source,
@Destination, @Motive, @Response, @Observation);
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'EL CODIGO YA EXISTE!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[NewBrand]   Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[NewBrand]
    @id nvarchar(2),
    @Description nvarchar(50),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO Brand VALUES(@id, @Description);

```

```

        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'EL CODIGO YA EXISTE!!!';
    END CATCH
END
GO
/***** Object: Table [dbo].[HistoryLog]   Script Date: 10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[HistoryLog](
    [id] [uniqueidentifier] NOT NULL,
    [idGPS] [int] NOT NULL,
    [idMessageType] [int] NOT NULL,
    [Message] [nvarchar](512) NOT NULL,
    [Date] [nvarchar](22) NOT NULL,
    CONSTRAINT [PK_HistoryLog] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: StoredProcedure [dbo].[NewTransportCompany]   Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[NewTransportCompany]
    @DNI nvarchar(50),
    @CorporateName nvarchar(250),
    @Address nvarchar(250),
    @City nvarchar(100),
    @State nvarchar(100),
    @Country nvarchar(100),

```

```

        @Description nvarchar(250),
        @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO Transport_Company VALUES(NEWID(), @DNI,
        @CorporateName, @Address, @City, @State, @Country, @Description);
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'EL CODIGO YA EXISTE';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[DeleteTransportCompany]   Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DeleteTransportCompany]
    @id uniqueidentifier,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        DELETE FROM Transport_Company WHERE id=@id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 547
            SET @Result = 'EXISTEN DEPENDENCIAS!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[NewRMC]   Script Date: 10/05/2010
18:44:08 *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[NewRMC]
    @GPSname nvarchar(17),
    @UTCTime nvarchar(6),
    @Flag nvarchar(1),
    @Latitude nvarchar(20),
    @Longitude nvarchar(20),
    @Speed nvarchar(10),
    @Course nvarchar(10),
    @UTCDate nvarchar(10),
    @MagneticVariation nvarchar(20),
    @Checksum nvarchar(10),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO RMC VALUES(NEWID(), @GPSname, @UTCTime,
            @Flag, geography::STGeomFromText('POINT(' + @Latitude + ' ' + @Longitude +
            ')', 4326), @Speed, @Course, @UTCDate, @MagneticVariation, @Checksum);
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'EL CODIGO YA EXISTE!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[NewMessageType]  Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[NewMessageType]
    @id int,
    @MessageType nvarchar(50),

```

```

        @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO MessageType VALUES(@id, @MessageType);
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'EL CODIGO YA EXISTE';
    END CATCH

```

END

GO

/***** Object: StoredProcedure [dbo].[NewLine] Script Date: 10/05/2010

18:44:08 *****/

SET ANSI_NULLS ON

GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE [dbo].[NewLine]

@idBrand nvarchar(2),

@id nvarchar(3),

@Description nvarchar(50),

@Result nvarchar(50) OUTPUT

AS

BEGIN

SET NOCOUNT ON;

BEGIN TRY

INSERT INTO Line VALUES(@idBrand, @id, @Description);

SET @Result = 'OK';

END TRY

BEGIN CATCH

IF ERROR_NUMBER()= 547

SET @Result = 'NO EXISTE LA MARCA';

END CATCH

END

GO

/***** Object: Table [dbo].[History] Script Date: 10/05/2010 18:44:08 *****/

SET ANSI_NULLS ON

```

GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[History](
    [id] [uniqueidentifier] NOT NULL,
    [idHistoryLog] [uniqueidentifier] NOT NULL,
    [idGPS] [int] NOT NULL,
    [GeoReference] [geography] NOT NULL,
    [Status] [nvarchar](50) NOT NULL,
    [UTCDate] [nvarchar](10) NOT NULL,
    [UTCTime] [nvarchar](6) NOT NULL,
    [Speed] [nvarchar](10) NOT NULL,
    [Passengers] [nvarchar](5) NOT NULL,
    [CO2] [nvarchar](10) NOT NULL,
    [Fuel] [nvarchar](10) NOT NULL,
    [Event] [nvarchar](1) NOT NULL,
    CONSTRAINT [PK__History__3213E83F5E8A0973] PRIMARY KEY
    CLUSTERED
    (
        [id] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[GPSVehicle]  Script Date: 10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[GPSVehicle](
    [idVehicle] [uniqueidentifier] NOT NULL,
    [idGPS] [int] NOT NULL
) ON [PRIMARY]
GO
/***** Object: StoredProcedure [dbo].[NewVehicleByUser]  Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO

```

```

SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[NewVehicleByUser]
    @idUser uniqueidentifier,
    @GPS int,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO VehicleByUser VALUES(NEWID(), @idUser, @GPS);
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'EL CODIGO YA EXISTE!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[NewVehicleByOperator]   Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[NewVehicleByOperator]
    @idUser uniqueidentifier,
    @GPS int,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO VehicleByOperator VALUES(NEWID(), @idUser,
@GPS);
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'EL CODIGO YA EXISTE!!!';
    END CATCH

```

```

END
GO
/***** Object: StoredProcedure [dbo].[NewVehicle]  Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[NewVehicle]
    @Plate nchar(10),
    @Register nvarchar(10),
    @Model nchar(10),
    @Empowered nchar(10),
    @Series nvarchar(50),
    @Chassis nvarchar(50),
    @Weight nchar(10),
    @Motor nvarchar(50),
    @Enabling nvarchar(10),
    @EnablingDuration int,
    @Policy nvarchar(10),
    @PolicyExpiration nvarchar(10),
    @idTransport_Company uniqueidentifier,
    @idInsurance_Company uniqueidentifier,
    @Membership nvarchar(10),
    @Expiration nvarchar(10),
    @idBrand nvarchar(2),
    @idLine nvarchar(3),
    @idColor int,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO Vehicle VALUES(NEWID(),
            @Plate,
            CONVERT(VARCHAR(10), GETDATE(), 101),
            @Register,
            @Model,
            @Empowered,

```

```

        @Series,
        @Chassis,
        @Weight,
        @Motor,
        @Enabling,
        @EnablingDuration,
        @Policy,
        @PolicyExpiration,
        @idTransport_Company,
        @idInsurance_Company,
        @Membership,
        @Expiration,
        @idBrand,
        @idLine,
        @idColor);
    SET @Result = 'OK';
END TRY
BEGIN CATCH
    IF ERROR_NUMBER()= 547
        SET @Result = 'NO EXISTE LA CONFIGURACION, LA
MARCA, LA LINEA O EL COLOR';
    ELSE
        SET @Result = 'EL CODIGO YA EXISTE';
END CATCH
END
GO
/***** Object: UserDefinedFunction [dbo].[GetLines]  Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[GetLines]
(
    @idBrand nvarchar(2)
)
RETURNS TABLE
AS
RETURN

```

```

(
    SELECT [idBrand] as f01
    ,[id] as f02
    ,[Description] as f03
    FROM [vms].[dbo].[Line]
    WHERE idBrand=@idBrand
)
GO
/***** Object: StoredProcedure [dbo].[NewHistoryLog]   Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[NewHistoryLog]
    @idGPS int,
    @idMessageType int,
    @Message nvarchar(512),
    @Date nvarchar(22),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @myid nvarchar(40);
    SET @myid = NEWID();
    BEGIN TRY
        INSERT INTO HistoryLog VALUES(@myid, @idGPS,
@idMessageType, @Message, @Date);
        SET @Result = @myid;
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER()= 547
            SET @Result = 'NO EXISTE EL TIPO DE MENSAJE O EL
GPS';
        ELSE
            SET @Result = 'EL CODIGO YA EXISTE';
    END CATCH
END
GO

```

/***** Object: StoredProcedure [dbo].[DeleteHistoryLog] Script Date: 10/05/2010
18:44:08 *****/

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DeleteHistoryLog]
    @id int,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        DELETE FROM HistoryLog WHERE id=@id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 547
            SET @Result = 'EXISTEN DEPENDENCIAS!!!';
    END CATCH
END
GO
```

/***** Object: StoredProcedure [dbo].[DeleteLine] Script Date: 10/05/2010
18:44:08 *****/

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DeleteLine]
    @idBrand nvarchar(2),
    @id nvarchar(3),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        DELETE FROM Line WHERE id=@id and idBrand=@idBrand;
        SET @Result = 'OK';
    END TRY
```

```

        BEGIN CATCH
            IF ERROR_NUMBER() = 547
                SET @Result = 'EXISTEN DEPENDENCIAS!!!';
        END CATCH
    END
GO

```

```

/***** Object: StoredProcedure [dbo].[DeleteVehicleByUser]  Script Date:
10/05/2010 18:44:08 *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DeleteVehicleByUser]
    @id uniqueidentifier,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        DELETE FROM VehicleByUser WHERE id=@id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 547
            SET @Result = 'EXISTEN DEPENDENCIAS!!!';
    END CATCH
END

```

```

GO
/***** Object: StoredProcedure [dbo].[DeleteVehicleByOperator]  Script Date:
10/05/2010 18:44:08 *****/

```

```

SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DeleteVehicleByOperator]
    @id uniqueidentifier,
    @Result nvarchar(50) OUTPUT
AS
BEGIN

```

```

SET NOCOUNT ON;
BEGIN TRY
    DELETE FROM VehicleByOperator WHERE id=@id;
    SET @Result = 'OK';
END TRY
BEGIN CATCH
    IF ERROR_NUMBER() = 547
        SET @Result = 'EXISTEN DEPENDENCIAS!!!';
END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[DeleteVehicle]  Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DeleteVehicle]
    @id uniqueidentifier,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        DELETE FROM Vehicle WHERE id=@id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 547
            SET @Result = 'EXISTEN DEPENDENCIAS!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[EditVehicleByUser]  Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

CREATE PROCEDURE [dbo].[EditVehicleByUser]
    @id int,
    @idUser uniqueidentifier,
    @GPS int,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        UPDATE VehicleByUser SET idUser=@idUser, GPS=@GPS
WHERE id=@id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'NO EXISTE EL REGISTRO A MODIFICAR!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[EditVehicleByOperator]   Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[EditVehicleByOperator]
    @id int,
    @idUser uniqueidentifier,
    @GPS int,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        UPDATE VehicleByOperator SET id=@id, idUser=@idUser,
GPS=@GPS;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'NO EXISTE EL REGISTRO A MODIFICAR!!!';
    END CATCH
END

```

```

        END CATCH
    END
GO
/***** Object: StoredProcedure [dbo].[EditVehicle]   Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[EditVehicle]
    @id uniqueidentifier,
    @Plate nchar(10),
    @InnerRegisterDate nvarchar(10),
    @RegisterDate nvarchar(10),
    @Model nchar(10),
    @Empowered nchar(10),
    @Series nvarchar(50),
    @Chassis nvarchar(50),
    @Weight nchar(10),
    @Motor nvarchar(50),
    @EnablingDate nvarchar(10),
    @EnablingDuration int,
    @Policy nvarchar(10),
    @PolicyExpirationDate nvarchar(10),
    @idTransport_Company uniqueidentifier,
    @idInsurance_Company uniqueidentifier,
    @MembershipDate nvarchar(10),
    @ExpirationDate nvarchar(10),
    @idBrand nvarchar(2),
    @idLine nvarchar(3),
    @idColor int,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        UPDATE Vehicle SET
            Plate=@Plate,
            InnerRegisterDate=@InnerRegisterDate,

```

```

RegisterDate=@RegisterDate,
Model=@Model,
Empowered=@Empowered,
Series=@Series,
Chassis=@Chassis,
[Weight]=@Weight,
Motor=@Motor,
EnablingDate=@EnablingDate,
EnablingDuration=@EnablingDuration,
Policy=@Policy,
PolicyExpirationDate=@PolicyExpirationDate,
idTransport_Company=@idTransport_Company,
idInsurance_Company=@idInsurance_Company,
MembershipDate=@MembershipDate,
ExpirationDate=@ExpirationDate,
idBrand=@idBrand,
idLine=@idLine,
idColor=@idColor WHERE id=@id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'NO EXISTE EL REGISTRO A MODIFICAR!!!';
    END CATCH
END
GO
/***** Object: UserDefinedFunction [dbo].[GetVehicles]  Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[GetVehicles]
(
)
RETURNS TABLE
AS
RETURN
(
SELECT DISTINCT

```

```

        Vehicle.id AS f01, Vehicle.Plate AS f02, Vehicle.InnerRegisterDate
AS f03, Vehicle.RegisterDate AS f04, Vehicle.Model AS f05, Vehicle.Empowered
AS f06,
        Vehicle.Series AS f07, Vehicle.Chassis AS f08, Vehicle.Weight AS
f09, Vehicle.Motor AS f10, Vehicle.EnablingDate AS f11, Vehicle.EnablingDuration
AS f12,
        Vehicle.Policy AS f13, Vehicle.PolicyExpirationDate AS f14,
Transport_Company.CorporateName AS f15,
Insurance_Company.CorporateName AS f16,
        Vehicle.MembershipDate AS f17, Vehicle.ExpirationDate AS f18,
Brand.Description AS f19, Line.Description AS f20, Color.Description AS f21
FROM      Vehicle INNER JOIN
        Brand ON Vehicle.idBrand = Brand.id INNER JOIN
        Color ON Vehicle.idColor = Color.id INNER JOIN
        Insurance_Company ON Vehicle.idInsurance_Company =
Insurance_Company.id INNER JOIN
        Line ON Brand.id = Line.idBrand AND Vehicle.idLine = Line.id
INNER JOIN
        Transport_Company ON Vehicle.idTransport_Company =
Transport_Company.id

)
GO
/***** Object: UserDefinedFunction [dbo].[GetVehicleByUsers]  Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[GetVehicleByUsers]()
RETURNS TABLE
AS
RETURN
(
SELECT  VehicleByUser.id AS f01, VehicleByUser.idUser AS f02, GPS.GPS AS
f03, Vehicle.Plate AS f04
FROM      VehicleByUser INNER JOIN
        GPS ON VehicleByUser.GPS = GPS.id INNER JOIN
        GPSVehicle ON GPS.id = GPSVehicle.idGPS INNER JOIN

```

```

Vehicle ON GPSVehicle.idVehicle = Vehicle.id

)
GO
/***** Object: UserDefinedFunction [dbo].[GetVehicleByOperators]  Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[GetVehicleByOperators]()
RETURNS TABLE
AS
RETURN
(
SELECT  VehicleByOperator.id AS f01, VehicleByOperator.idUser AS f02,
GPS.GPS AS f03, Vehicle.Plate AS f04
FROM    VehicleByOperator INNER JOIN
        GPS ON VehicleByOperator.GPS = GPS.id INNER JOIN
        GPSVehicle ON GPS.id = GPSVehicle.idGPS INNER JOIN
        Vehicle ON GPSVehicle.idVehicle = Vehicle.id
)
GO
/***** Object: Table [dbo].[Alarm]  Script Date: 10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Alarm](
    [id] [uniqueidentifier] NOT NULL,
    [idEvent] [int] NOT NULL,
    [idHistory] [uniqueidentifier] NOT NULL,
    [idGps] [int] NOT NULL,
    [IsNotified] [bit] NOT NULL,
    [Date] [nvarchar](10) NOT NULL,
    [NotifyDate] [nvarchar](10) NULL,
    [Operator] [nvarchar](250) NULL,
PRIMARY KEY CLUSTERED
(

```

```

        [id] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: StoredProcedure [dbo].[DeleteHistory]   Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DeleteHistory]
    @id int,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        DELETE FROM History WHERE id=@id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 547
            SET @Result = 'EXISTEN DEPENDENCIAS!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[DeleteGPSVehicle]   Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DeleteGPSVehicle]
    @idVehicle uniqueidentifier,
    @idGPS int,
    @Result nvarchar(50) OUTPUT
AS

```

```

BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        DELETE FROM GPSVehicle WHERE idVehicle=@idVehicle and
idGPS=@idGPS;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 547
            SET @Result = 'EXISTEN DEPENDENCIAS!!!';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[NewHistory]    Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[NewHistory]
    @idHistoryLog uniqueidentifier,
    @idGPS int,
    @Latitude nvarchar(20),
    @Longitude nvarchar(20),
    @Status nvarchar(50),
    @UTCDate nvarchar(10),
    @UTCTime nvarchar(6),
    @Speed nvarchar(10),
    @CO2 nvarchar(10),
    @Passengers nvarchar(5),
    @Fuel nvarchar(10),
    @Event nvarchar(1),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    DECLARE @myid nvarchar(40);
    SET @myid = NEWID();
    BEGIN TRY

```

```

        INSERT INTO History VALUES(@myid, @idHistoryLog, @idGPS,
geography::STGeomFromText('POINT(' + @Latitude + ' ' + @Longitude + ')',
4326), @Status, @UTCDate, @UTCTime, @Speed, @Passengers, @CO2,
@Fuel, @Event);
        SET @Result = @myid;
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER()= 547
            SET @Result = 'NO EXISTE EL HISTORICO LOG O EL GPS';
        ELSE
            SET @Result = 'EL CODIGO YA EXISTE';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[NewGPSVehicle]  Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[NewGPSVehicle]
    @idVehicle uniqueidentifier,
    @idGPS int,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO GPSVehicle VALUES(@idVehicle, @idGPS);
        UPDATE GPS SET IsAssigned=1 WHERE id=@idGPS;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        SET @Result = 'NO EXISTE EL CODIGO DEL VEHICULO O EL
GPS';
    END CATCH
END
GO

```

```

/***** Object: UserDefinedFunction [dbo].[GetGPSVehicles]  Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[GetGPSVehicles]()
RETURNS TABLE
AS
RETURN
(
SELECT  Vehicle.id as f01, GPS.id as f02,  Vehicle.Plate as f03, GPS.GPS as f04
FROM    GPSVehicle INNER JOIN
        GPS ON GPSVehicle.idGPS = GPS.id INNER JOIN
        Vehicle ON GPSVehicle.idVehicle = Vehicle.id
)
GO

```

```

/***** Object: UserDefinedFunction [dbo].[GetGPSByPlates]  Script Date:
10/05/2010 18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE FUNCTION [dbo].[GetGPSByPlates]()
RETURNS TABLE
AS
RETURN
(
SELECT  GPSVehicle.idGPS AS f01, Vehicle.Plate AS f02
FROM    GPSVehicle INNER JOIN
        Vehicle ON GPSVehicle.idVehicle = Vehicle.id
)
GO

```

```

/***** Object: StoredProcedure [dbo].[NewAlarm]  Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO

```

```

CREATE PROCEDURE [dbo].[NewAlarm]
    @idEvent int,
    @idHistory uniqueidentifier,
    @idGPS int,
    @IsNotified bit,
    @Date nvarchar(10),
    @NotifyDate nvarchar(10),
    @Operator nvarchar(255),
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY
        INSERT INTO Alarm VALUES(NEWID(), @idEvent, @idHistory,
        @idGPS, @isNotified, @Date, @NotifyDate, @Operator);
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER()= 547
            SET @Result = 'NO EXISTE EL EVENTO, EL HISTORICO O
EL GPS';
        ELSE
            SET @Result = 'EL CODIGO YA EXISTE';
    END CATCH
END
GO
/***** Object: StoredProcedure [dbo].[DeleteAlarm]   Script Date: 10/05/2010
18:44:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[DeleteAlarm]
    @id uniqueidentifier,
    @Result nvarchar(50) OUTPUT
AS
BEGIN
    SET NOCOUNT ON;
    BEGIN TRY

```

```

        DELETE FROM Alarm WHERE id=@id;
        SET @Result = 'OK';
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 547
            SET @Result = 'EXISTEN DEPENDENCIAS!!!';
    END CATCH
END
GO
/***** Object: ForeignKey [FK_VehicleByUser_GPS]  Script Date: 10/05/2010
18:44:08 *****/
ALTER TABLE [dbo].[VehicleByUser] WITH CHECK ADD CONSTRAINT
[FK_VehicleByUser_GPS] FOREIGN KEY([GPS])
REFERENCES [dbo].[GPS] ([id])
GO
ALTER TABLE [dbo].[VehicleByUser] CHECK CONSTRAINT
[FK_VehicleByUser_GPS]
GO
/***** Object: ForeignKey [FK_VehicleByOperator_GPS]  Script Date:
10/05/2010 18:44:08 *****/
ALTER TABLE [dbo].[VehicleByOperator] WITH CHECK ADD CONSTRAINT
[FK_VehicleByOperator_GPS] FOREIGN KEY([GPS])
REFERENCES [dbo].[GPS] ([id])
GO
ALTER TABLE [dbo].[VehicleByOperator] CHECK CONSTRAINT
[FK_VehicleByOperator_GPS]
GO
/***** Object: ForeignKey [FK_Vehicle_Brand]  Script Date: 10/05/2010 18:44:08
*****/
ALTER TABLE [dbo].[Vehicle] WITH CHECK ADD CONSTRAINT
[FK_Vehicle_Brand] FOREIGN KEY([idBrand])
REFERENCES [dbo].[Brand] ([id])
GO
ALTER TABLE [dbo].[Vehicle] CHECK CONSTRAINT [FK_Vehicle_Brand]
GO
/***** Object: ForeignKey [FK_Vehicle_Color]  Script Date: 10/05/2010 18:44:08
*****/
ALTER TABLE [dbo].[Vehicle] WITH CHECK ADD CONSTRAINT
[FK_Vehicle_Color] FOREIGN KEY([idColor])

```

```

REFERENCES [dbo].[Color] ([id])
GO
ALTER TABLE [dbo].[Vehicle] CHECK CONSTRAINT [FK_Vehicle_Color]
GO
/***** Object: ForeignKey [FK_Vehicle_Insurance_Company]  Script Date:
10/05/2010 18:44:08 *****/
ALTER TABLE [dbo].[Vehicle] WITH CHECK ADD CONSTRAINT
[FK_Vehicle_Insurance_Company] FOREIGN KEY([idInsurance_Company])
REFERENCES [dbo].[Insurance_Company] ([id])
GO
ALTER TABLE [dbo].[Vehicle] CHECK CONSTRAINT
[FK_Vehicle_Insurance_Company]
GO
/***** Object: ForeignKey [FK_Vehicle_Transport_Company]  Script Date:
10/05/2010 18:44:08 *****/
ALTER TABLE [dbo].[Vehicle] WITH CHECK ADD CONSTRAINT
[FK_Vehicle_Transport_Company] FOREIGN KEY([idTransport_Company])
REFERENCES [dbo].[Transport_Company] ([id])
GO
ALTER TABLE [dbo].[Vehicle] CHECK CONSTRAINT
[FK_Vehicle_Transport_Company]
GO
/***** Object: ForeignKey [FK_Line_Brand]  Script Date: 10/05/2010 18:44:08
*****/
ALTER TABLE [dbo].[Line] WITH CHECK ADD CONSTRAINT [FK_Line_Brand]
FOREIGN KEY([idBrand])
REFERENCES [dbo].[Brand] ([id])
GO
ALTER TABLE [dbo].[Line] CHECK CONSTRAINT [FK_Line_Brand]
GO
/***** Object: ForeignKey [FK_HistoryLog_GPS]  Script Date: 10/05/2010
18:44:08 *****/
ALTER TABLE [dbo].[HistoryLog] WITH CHECK ADD CONSTRAINT
[FK_HistoryLog_GPS] FOREIGN KEY([idGPS])
REFERENCES [dbo].[GPS] ([id])
GO
ALTER TABLE [dbo].[HistoryLog] CHECK CONSTRAINT [FK_HistoryLog_GPS]
GO

```

/***** Object: ForeignKey [FK_HistoryLog_MessageType] Script Date: 10/05/2010 18:44:08 *****/

```
ALTER TABLE [dbo].[HistoryLog] WITH CHECK ADD CONSTRAINT
[FK_HistoryLog_MessageType] FOREIGN KEY([idMessageType])
REFERENCES [dbo].[MessageType] ([id])
GO
```

```
ALTER TABLE [dbo].[HistoryLog] CHECK CONSTRAINT
[FK_HistoryLog_MessageType]
GO
```

/***** Object: ForeignKey [FK_History_GPS] Script Date: 10/05/2010 18:44:08 *****/

```
ALTER TABLE [dbo].[History] WITH CHECK ADD CONSTRAINT
[FK_History_GPS] FOREIGN KEY([idGPS])
REFERENCES [dbo].[GPS] ([id])
GO
```

```
ALTER TABLE [dbo].[History] CHECK CONSTRAINT [FK_History_GPS]
GO
```

/***** Object: ForeignKey [FK_History_HistoryLog] Script Date: 10/05/2010 18:44:08 *****/

```
ALTER TABLE [dbo].[History] WITH CHECK ADD CONSTRAINT
[FK_History_HistoryLog] FOREIGN KEY([idHistoryLog])
REFERENCES [dbo].[HistoryLog] ([id])
GO
```

```
ALTER TABLE [dbo].[History] CHECK CONSTRAINT [FK_History_HistoryLog]
GO
```

/***** Object: ForeignKey [FK_GPSVehicle_GPS] Script Date: 10/05/2010 18:44:08 *****/

```
ALTER TABLE [dbo].[GPSVehicle] WITH CHECK ADD CONSTRAINT
[FK_GPSVehicle_GPS] FOREIGN KEY([idGPS])
REFERENCES [dbo].[GPS] ([id])
GO
```

```
ALTER TABLE [dbo].[GPSVehicle] CHECK CONSTRAINT [FK_GPSVehicle_GPS]
GO
```

/***** Object: ForeignKey [FK_GPSVehicle_Vehicle] Script Date: 10/05/2010 18:44:08 *****/

```
ALTER TABLE [dbo].[GPSVehicle] WITH CHECK ADD CONSTRAINT
[FK_GPSVehicle_Vehicle] FOREIGN KEY([idVehicle])
REFERENCES [dbo].[Vehicle] ([id])
GO
```

```

ALTER TABLE [dbo].[GPSVehicle] CHECK CONSTRAINT
[FK_GPSVehicle_Vehicle]
GO
/***** Object: ForeignKey [FK_Alarm_Event]  Script Date: 10/05/2010 18:44:08
*****/
ALTER TABLE [dbo].[Alarm] WITH CHECK ADD CONSTRAINT
[FK_Alarm_Event] FOREIGN KEY([idEvent])
REFERENCES [dbo].[Event] ([id])
GO
ALTER TABLE [dbo].[Alarm] CHECK CONSTRAINT [FK_Alarm_Event]
GO
/***** Object: ForeignKey [FK_Alarm_GPS]  Script Date: 10/05/2010 18:44:08
*****/
ALTER TABLE [dbo].[Alarm] WITH CHECK ADD CONSTRAINT
[FK_Alarm_GPS] FOREIGN KEY([idGps])
REFERENCES [dbo].[GPS] ([id])
GO
ALTER TABLE [dbo].[Alarm] CHECK CONSTRAINT [FK_Alarm_GPS]
GO
/***** Object: ForeignKey [FK_Alarm_History]  Script Date: 10/05/2010 18:44:08
*****/
ALTER TABLE [dbo].[Alarm] WITH CHECK ADD CONSTRAINT
[FK_Alarm_History] FOREIGN KEY([idHistory])
REFERENCES [dbo].[History] ([id])
GO
ALTER TABLE [dbo].[Alarm] CHECK CONSTRAINT [FK_Alarm_History]
GO

```

Código Fuente Data Warehouse

Ilustración 40

```

<?xml version="1.0" encoding="utf-8"?>
<DataSource
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ddl2="http://schemas.microsoft.com/analysisservices/2003/engine/2"
xmlns:ddl2_2="http://schemas.microsoft.com/analysisservices/2003/engine/2/2"
xmlns:ddl100_100="http://schemas.microsoft.com/analysisservices/2008/engine/10

```

```

0/100"
xmlns:ddl200="http://schemas.microsoft.com/analysisservices/2010/engine/200"
xmlns:ddl200_200="http://schemas.microsoft.com/analysisservices/2010/engine/200/200"
xmlns:dwd="http://schemas.microsoft.com/DataWarehouse/Designer/1.0"
xsi:type="RelationalDataSource"
dwd:design-time-name="1dce53c2-1ffc-40e7-ac89-86139f033dfe"
xmlns="http://schemas.microsoft.com/analysisservices/2003/engine">
  <ID>Vms</ID>
  <Name>Vms</Name>
  <CreatedTimestamp>0001-01-01T05:00:00Z</CreatedTimestamp>
  <LastSchemaUpdate>0001-01-01T05:00:00Z</LastSchemaUpdate>
  <ConnectionString>Provider=SQLNCLI10.1;Data
Source=SYSTEMDEPT\SQL_SYSTEMDEPT;Integrated
Security=SSPI;Initial
Catalog=vms</ConnectionString>
  <ConnectionStringSecurity>Unchanged</ConnectionStringSecurity>
  <ImpersonationInfo>
    <ImpersonationMode>ImpersonateServiceAccount</ImpersonationMode>
    <ImpersonationInfoSecurity>Unchanged</ImpersonationInfoSecurity>
  </ImpersonationInfo>
  <Timeout>PT0S</Timeout>
</DataSource>

```

Ilustración 42

```

<?xml version="1.0" encoding="utf-8"?>
<Cube
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ddl2="http://schemas.microsoft.com/analysisservices/2003/engine/2"
xmlns:ddl2_2="http://schemas.microsoft.com/analysisservices/2003/engine/2/2"
xmlns:ddl100_100="http://schemas.microsoft.com/analysisservices/2008/engine/100/100"
xmlns:ddl200="http://schemas.microsoft.com/analysisservices/2010/engine/200"
xmlns:ddl200_200="http://schemas.microsoft.com/analysisservices/2010/engine/200/200"
xmlns:dwd="http://schemas.microsoft.com/DataWarehouse/Designer/1.0"
dwd:design-time-name="3059cb6e-f8e6-47f7-a541-a919348f2c31"
xmlns="http://schemas.microsoft.com/analysisservices/2003/engine">

```

```
<ID>Vms</ID>
<Name>Vms</Name>
<CreatedTimestamp>0001-01-01T05:00:00Z</CreatedTimestamp>
<LastSchemaUpdate>0001-01-01T05:00:00Z</LastSchemaUpdate>
<Annotations>
  <Annotation>
```

```
<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:Diagram
Layout</Name>
```

```
  <Value>
    <dds xmlns="">
      <diagram fontclsid="{0BE35203-8F91-11CE-9DE3-00AA004BB851}"
mouseiconclsid="{0BE35204-8F91-11CE-9DE3-00AA004BB851}"
defaultlayout="MSDDS.Rectilinear" defaultlineroute="MSDDS.Rectilinear"
version="7" nextobject="2" scale="100" pagebreakanchorx="0"
pagebreakanchory="0" pagebreaksize="0" pagebreaksizey="0" scrollleft="-
7627" scrolltop="-3413" gridx="150" gridy="150" marginx="5000"
margin="5000" zoom="100" x="18336" y="11245" bgcolor="15334399"
defaultpersistence="2" PrintPageNumbersMode="3" PrintMarginTop="0"
PrintMarginBottom="635" PrintMarginLeft="0" PrintMarginRight="0"
marqueeselectionmode="0" mousepointer="0" snaptogrid="0"
autotypeannotation="1" showscrollbars="0" viewpagebreaks="0"
donotforceconnectorsbehindshapes="1" backpictureclsid="{00000000-0000-
0000-0000-000000000000}">
        <font>
          <ddsxmlobjectstreamwrapper
binary="01000000900144420100065461686f6d61" />
        </font>
        <mouseicon>
          <ddsxmlobjectstreamwrapper binary="6c74000000000000" />
        </mouseicon>
      </diagram>
      <layoutmanager>
        <ddsxmlobj />
      </layoutmanager>
      <ddscontrol controlprogid="DdsShapes.DdsObjectManagedBridge.2"
tooltip="HistoryNumeric" left="-7150" top="-2805" logicalid="1"
controlid="1" masterid="0" hint1="0" hint2="0" width="3043" height="4419"
noresize="0" nomove="0" nodefaultattachpoints="0" autodrag="1"
```

```

usedefaultiddshape="1"      selectable="1"      showselectionhandles="1"
allownudging="1" isannotation="0" dontautolayout="0" groupcollapsed="0"
tabstop="1" visible="1" snaptogrid="0">
  <control>
    <ddsxmlobjectstreaminitwrapper
binary="00080000e30b000043110000" />
  </control>
  <layoutobject>
    <ddsxmlobj>
      <property name="LogicalObject" value="dbo_HistoryNumeric"
vartype="8" />
    </ddsxmlobj>
  </layoutobject>
  <shape groupshapeid="0" groupnode="0" />
</ddscontrol>
</dds>
</Value>
</Annotation>
<Annotation>

```

```

<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:ShowFri
endlyNames</Name>
  <Value>>true</Value>
</Annotation>
<Annotation>

```

```

<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:ShowRel
ationshipNames</Name>
  <Value>>false</Value>
</Annotation>
<Annotation>

```

```

<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:UseDiag
ramDefaultLayout</Name>
  <Value>>true</Value>
</Annotation>
<Annotation>

```

**<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:Diagram
ViewPortLeft</Name>
 <Value>-7627</Value>
 </Annotation>
 <Annotation>**

**<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:Diagram
ViewPortTop</Name>
 <Value>-3413</Value>
 </Annotation>
 <Annotation>**

**<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:Diagram
BoundingLeft</Name>
 <Value>-7150</Value>
 </Annotation>
 <Annotation>**

**<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:Diagram
BoundingTop</Name>
 <Value>-2805</Value>
 </Annotation>
 <Annotation>**

**<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:Diagram
Zoom</Name>
 <Value>100</Value>
 </Annotation>
 </Annotations>
 <LastProcessed>0001-01-01T05:00:00Z</LastProcessed>
 <State>Unprocessed</State>
 <Dimensions>
 <Dimension dwd:design-time-name="701e3298-9746-4f59-9408-
64e092bd44b1">
 <ID>History Numeric</ID>
 <Name>History Numeric</Name>
 <DimensionID>History Numeric</DimensionID>
 <Attributes>**

```

    <Attribute          dwd:design-time-name="fc57e05d-184f-4ef8-b49b-
782a5523a760">
      <AttributeID>UTC Date Time</AttributeID>
    </Attribute>
  </Attributes>
</Dimension>
</Dimensions>
<MeasureGroups>
  <MeasureGroup      dwd:design-time-name="ec8afbde-fe01-44ea-a04d-
4fdcaa4f0ed0">
    <ID>History Numeric</ID>
    <Name>History Numeric</Name>
    <CreatedTimestamp>0001-01-01T05:00:00Z</CreatedTimestamp>
    <LastSchemaUpdate>0001-01-01T05:00:00Z</LastSchemaUpdate>
    <LastProcessed>0001-01-01T05:00:00Z</LastProcessed>
    <State>Unprocessed</State>
    <Measures>
      <Measure          dwd:design-time-name="cfe38029-4f1a-4095-9afb-
70e1524c4fc7">
        <ID>Speed</ID>
        <Name>Speed</Name>
        <Source          dwd:design-time-name="3344b8bb-6aff-4860-8bc5-
9bc92ff49b37">
          <DataType>Double</DataType>
          <Source      xsi:type="ColumnBinding"          dwd:design-time-
name="931a6f84-0130-41ef-bfc1-9469e5bb0d16">
            <TableID>dbo_HistoryNumeric</TableID>
            <ColumnID>Speed</ColumnID>
          </Source>
        </Source>
      </Measure>
      <Measure          dwd:design-time-name="8ce300bf-15d0-4dbe-a772-
98f661c2afa2">
        <ID>Passengers</ID>
        <Name>Passengers</Name>
        <Source          dwd:design-time-name="c33ce681-ff2c-4267-8eed-
ae2f7f98b1cb">
          <DataType>Double</DataType>

```

```

    <Source      xsi:type="ColumnBinding"      dwd:design-time-
name="423b1d1e-1f8a-4f0b-8325-121dd38ad565">
      <TableID>dbo_HistoryNumeric</TableID>
      <ColumnID>Passengers</ColumnID>
    </Source>
  </Source>
</Measure>
<Measure      dwd:design-time-name="bd54e5f9-3a27-4b4c-a848-
7481d3223130">
  <ID>CO2</ID>
  <Name>CO2</Name>
  <Source      dwd:design-time-name="d4f1bcbc-752a-4196-ad92-
21142f523d2f">
    <DataType>Double</DataType>
    <Source      xsi:type="ColumnBinding"      dwd:design-time-
name="5a2ce3d0-0626-4709-8e82-9b95bbfc4f40">
      <TableID>dbo_HistoryNumeric</TableID>
      <ColumnID>CO2</ColumnID>
    </Source>
  </Source>
</Measure>
<Measure      dwd:design-time-name="991be639-1291-4ff9-b617-
f339a6bbdba7">
  <ID>Fuel</ID>
  <Name>Fuel</Name>
  <Source      dwd:design-time-name="67e0f0af-8c0d-43a0-934b-
cee8218e7131">
    <DataType>Double</DataType>
    <Source      xsi:type="ColumnBinding"      dwd:design-time-name="ffa7f5ae-
fe6e-49b8-ad83-49693a2a2f52">
      <TableID>dbo_HistoryNumeric</TableID>
      <ColumnID>Fuel</ColumnID>
    </Source>
  </Source>
</Measure>
<Measure      dwd:design-time-name="c4fa2870-9d6e-4021-bf1c-
abc09cc932ec">
  <ID>History Numeric Count</ID>
  <Name>History Numeric Count</Name>

```

```

    <AggregateFunction>Count</AggregateFunction>
    <Source          dwd:design-time-name="3db5231e-8be5-4bfb-a1a9-
b1eb714c7791">
        <DataType>Integer</DataType>
        <DataSize>4</DataSize>
        <Source  xsi:type="RowBinding"  dwd:design-time-name="7754683b-
021f-430a-bc4a-c8c296e87ec1">
            <TableID>dbo_HistoryNumeric</TableID>
        </Source>
    </Source>
</Measure>
</Measures>
<StorageMode>Molap</StorageMode>
<ProcessingMode>Regular</ProcessingMode>
<Dimensions>
    <Dimension  xsi:type="RegularMeasureGroupDimension"  dwd:design-
time-name="56de7c64-158e-4b61-ae63-009737be7776">
        <CubeDimensionID>History Numeric</CubeDimensionID>
        <Attributes>
            <Attribute          dwd:design-time-name="1fd098d0-6b9c-4802-8c67-
d7c3ebfc0851">
                <AttributeID>UTC Date Time</AttributeID>
                <KeyColumns>
                    <KeyColumn  dwd:design-time-name="0c6a6489-bbbc-42b3-8a88-
8c8adc65fb1b">
                        <DataType>Date</DataType>
                        <Source          xsi:type="ColumnBinding"          dwd:design-time-
name="95afb8dd-4ec1-42e0-a389-e1c0378da3d6">
                            <TableID>dbo_HistoryNumeric</TableID>
                            <ColumnID>UTCDateTime</ColumnID>
                        </Source>
                    </KeyColumn>
                </KeyColumns>
                <Type>Granularity</Type>
            </Attribute>
        </Attributes>
    </Dimension>
</Dimensions>
<Partitions />

```

```

    </MeasureGroup>
  </MeasureGroups>
  <Source          dwd:design-time-name="bc34f314-d45d-46e8-bdb3-
a02eea410678">
    <DataSourceViewID>Vms</DataSourceViewID>
  </Source>
</Cube>

```

Ilustración 44

```

<?xml version="1.0" encoding="utf-8"?>
<MiningStructure      xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:ddl2="http://schemas.microsoft.com/analysiservices/2003/engine/2"
xmlns:ddl2_2="http://schemas.microsoft.com/analysiservices/2003/engine/2
/2"
xmlns:ddl100_100="http://schemas.microsoft.com/analysiservices/2008/eng
ine/100/100"
xmlns:ddl200="http://schemas.microsoft.com/analysiservices/2010/engine/2
00"
xmlns:ddl200_200="http://schemas.microsoft.com/analysiservices/2010/eng
ine/200/200"
xmlns:dwd="http://schemas.microsoft.com/DataWarehouse/Designer/1.0"
dwd:design-time-name="4bbab279-ae30-4901-8d00-bd4d746a6cc6"
xmlns="http://schemas.microsoft.com/analysiservices/2003/engine">
  <ID>History Numeric</ID>
  <Name>History Numeric</Name>
  <CreatedTimestamp>0001-01-01T00:00:00Z</CreatedTimestamp>
  <LastSchemaUpdate>0001-01-01T00:00:00Z</LastSchemaUpdate>
  <Annotations>
    <Annotation>
      <Name>MDXFilterComponent</Name>
      <Value>&lt;?xml      version="1.0"      encoding="utf-16"?&gt;&lt;Filter
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.microsoft.com/SQLServer/reporting/QuerySpecificati
on"&gt;&lt;FilterItems
xmlns="http://schemas.microsoft.com/AnalysisServices/QueryDefinition"
/&gt;&lt;/Filter&gt;</Value>

```

**</Annotation>
<Annotation>**

**<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:ShowFriendlyNames</Name>
<Value>>true</Value>
</Annotation>
<Annotation>**

**<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:DiagramLayout</Name>
</Annotation>
<Annotation>**

**<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:ShowRelationshipNames</Name>
<Value>>false</Value>
</Annotation>
<Annotation>**

**<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:UseDiagramDefaultLayout</Name>
<Value>>true</Value>
</Annotation>
<Annotation>**

**<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:DiagramViewPortLeft</Name>
<Value>0</Value>
</Annotation>
<Annotation>**

**<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:DiagramViewPortTop</Name>
<Value>0</Value>
</Annotation>
<Annotation>**

```
<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:Diagram
BoundingLeft</Name>
  <Value>0</Value>
</Annotation>
<Annotation>
```

```
<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:Diagram
BoundingTop</Name>
  <Value>0</Value>
</Annotation>
<Annotation>
```

```
<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:Diagram
Zoom</Name>
  <Value>100</Value>
</Annotation>
</Annotations>
<LastProcessed>0001-01-01T00:00:00Z</LastProcessed>
<State>Unprocessed</State>
<Source      xsi:type="CubeDimensionBinding"      dwd:design-time-
name="69b9d2ce-5203-43f8-87bc-238ba87374f3">
  <DataSourceID>.</DataSourceID>
  <CubeID>Vms</CubeID>
  <CubeDimensionID>History Numeric</CubeDimensionID>
</Source>
<Columns>
  <Column    xsi:type="ScalarMiningStructureColumn"  dwd:design-time-
name="668683f5-bc24-40fe-ad07-7087e78e9504">
    <ID>UTC Date Time</ID>
    <Name>UTC Date Time</Name>
    <IsKey>true</IsKey>
    <Type>Date</Type>
    <Source    xsi:type="CubeAttributeBinding"      dwd:design-time-
name="a1463b3e-1f98-4f83-b4a9-35e810d66f63">
      <CubeID>Vms</CubeID>
      <CubeDimensionID>History Numeric</CubeDimensionID>
      <AttributeID>UTC Date Time</AttributeID>
      <Type>All</Type>
```

```

    </Source>
    <Content>Key Time</Content>
  </Column>
  <Column xsi:type="ScalarMiningStructureColumn"   dwd:design-time-
name="ea3d5873-a479-4f62-820e-7e89ae189be3">
    <ID>CO2</ID>
    <Name>CO2</Name>
    <Type>Double</Type>
    <Source xsi:type="MeasureBinding"   dwd:design-time-name="1952c290-
eede-4012-bfdd-b6cef60df68d">
      <MeasureID>CO2</MeasureID>
    </Source>
    <Content>Continuous</Content>
  </Column>
  <Column xsi:type="ScalarMiningStructureColumn"   dwd:design-time-
name="e1ca026a-57d6-426e-9c6e-06a98effc587">
    <ID>Fuel</ID>
    <Name>Fuel</Name>
    <Type>Double</Type>
    <Source xsi:type="MeasureBinding"   dwd:design-time-name="b854444d-
ce7b-46b4-95a3-2d8efc6bbc91">
      <MeasureID>Fuel</MeasureID>
    </Source>
    <Content>Continuous</Content>
  </Column>
  <Column xsi:type="ScalarMiningStructureColumn"   dwd:design-time-
name="2d115a21-1dff-4f6a-b7ae-e6b97300bfb1">
    <ID>Passengers</ID>
    <Name>Passengers</Name>
    <Type>Double</Type>
    <Source xsi:type="MeasureBinding"   dwd:design-time-name="f79804a3-
db31-40e4-b128-0c73a52266b7">
      <MeasureID>Passengers</MeasureID>
    </Source>
    <Content>Continuous</Content>
  </Column>
  <Column xsi:type="ScalarMiningStructureColumn"   dwd:design-time-
name="2fd0966e-506c-400d-bc19-5921515c54d7">
    <ID>Speed</ID>

```

```

    <Name>Speed</Name>
    <Type>Double</Type>
    <Source xsi:type="MeasureBinding" dwd:design-time-name="5c550b89-
a280-4a99-ad1a-f2997316b0b8">
      <MeasureID>Speed</MeasureID>
    </Source>
    <Content>Continuous</Content>
  </Column>
</Columns>
<MiningModels>
  <MiningModel dwd:design-time-name="961dcc8e-7577-49e8-ab9b-
c2a5ee9f7889">
    <ID>History Numeric</ID>
    <Name>History Numeric</Name>
    <CreatedTimestamp>0001-01-01T00:00:00Z</CreatedTimestamp>
    <LastSchemaUpdate>0001-01-01T00:00:00Z</LastSchemaUpdate>
    <LastProcessed>0001-01-01T00:00:00Z</LastProcessed>
    <State>Unprocessed</State>
    <Algorithm>Microsoft_Time_Series</Algorithm>
    <Columns>
      <Column dwd:design-time-name="1eda22c2-8465-4b0d-8275-
756e270bfb12">
        <ID>UTC Date Time</ID>
        <Name>UTC Date Time</Name>
        <SourceColumnID>UTC Date Time</SourceColumnID>
        <Usage>Key</Usage>
      </Column>
      <Column dwd:design-time-name="f666a8f7-21e2-4d34-a8e4-
68f2dd8e5d33">
        <ID>CO2</ID>
        <Name>CO2</Name>
        <SourceColumnID>CO2</SourceColumnID>
        <ModelingFlags>
          <ModelingFlag>REGRESSOR</ModelingFlag>
        </ModelingFlags>
      </Column>
      <Column dwd:design-time-name="9644ce04-e549-477a-a068-
e5f48fda1818">
        <ID>Fuel</ID>

```

```

    <Name>Fuel</Name>
    <SourceColumnID>Fuel</SourceColumnID>
    <ModelingFlags>
      <ModelingFlag>REGRESSOR</ModelingFlag>
    </ModelingFlags>
  </Column>
  <Column          dwd:design-time-name="7e6ef8db-9df7-4286-8a74-
b6c96e510ddc">
    <ID>Passengers</ID>
    <Name>Passengers</Name>
    <SourceColumnID>Passengers</SourceColumnID>
    <ModelingFlags>
      <ModelingFlag>REGRESSOR</ModelingFlag>
    </ModelingFlags>
  </Column>
  <Column          dwd:design-time-name="1b980e85-285e-4b84-85f2-
17f965a70bc6">
    <ID>Speed</ID>
    <Name>Speed</Name>
    <SourceColumnID>Speed</SourceColumnID>
    <Usage>Predict</Usage>
    <ModelingFlags>
      <ModelingFlag>REGRESSOR</ModelingFlag>
    </ModelingFlags>
  </Column>
</Columns>
<AllowDrillThrough>true</AllowDrillThrough>
</MiningModel>
</MiningModels>
</MiningStructure>

```

Ilustración 45

```

<?xml version="1.0" encoding="utf-8"?>
<MiningStructure          xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

```

```

xmlns:ddl2="http://schemas.microsoft.com/analysisservices/2003/engine/2"
xmlns:ddl2_2="http://schemas.microsoft.com/analysisservices/2003/engine/2/2"
xmlns:ddl100_100="http://schemas.microsoft.com/analysisservices/2008/engine/100/100"
xmlns:ddl200="http://schemas.microsoft.com/analysisservices/2010/engine/200"
xmlns:ddl200_200="http://schemas.microsoft.com/analysisservices/2010/engine/200/200"
xmlns:dwd="http://schemas.microsoft.com/DataWarehouse/Designer/1.0"
dwd:design-time-name="468ed1b5-0cec-4209-b87a-5419b4d4679c"
xmlns="http://schemas.microsoft.com/analysisservices/2003/engine">
  <ID>History Numeric By Date</ID>
  <Name>History Numeric By Date</Name>
  <CreatedTimestamp>0001-01-01T00:00:00Z</CreatedTimestamp>
  <LastSchemaUpdate>0001-01-01T00:00:00Z</LastSchemaUpdate>
  <Annotations>
    <Annotation>
      <Name>MDXFilterComponent</Name>
      <Value>&lt;?xml version="1.0" encoding="utf-16"?&gt;&lt;Filter
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="http://schemas.microsoft.com/SQLServer/reporting/QuerySpecification" &gt;&lt;FilterItems
xmlns="http://schemas.microsoft.com/AnalysisServices/QueryDefinition" /&gt;&lt;/Filter&gt;</Value>
    </Annotation>
  </Annotation>

  <Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:ShowFriendlyNames</Name>
  <Value>true</Value>
  </Annotation>
</Annotation>

  <Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:DiagramLayout</Name>
  </Annotation>
</Annotation>

```

**<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:ShowRelationshipNames</Name>
<Value>>false</Value>
</Annotation>
<Annotation>**

**<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:UseDiagramDefaultLayout</Name>
<Value>>true</Value>
</Annotation>
<Annotation>**

**<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:DiagramViewPortLeft</Name>
<Value>0</Value>
</Annotation>
<Annotation>**

**<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:DiagramViewPortTop</Name>
<Value>0</Value>
</Annotation>
<Annotation>**

**<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:DiagramBoundingLeft</Name>
<Value>0</Value>
</Annotation>
<Annotation>**

**<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:DiagramBoundingTop</Name>
<Value>0</Value>
</Annotation>
<Annotation>**

<Name>http://schemas.microsoft.com/DataWarehouse/Designer/1.0:DiagramZoom</Name>

```

    <Value>100</Value>
  </Annotation>
</Annotations>
<LastProcessed>0001-01-01T00:00:00Z</LastProcessed>
<State>Unprocessed</State>
<Source      xsi:type="CubeDimensionBinding"      dwd:design-time-
name="3a3387d6-5faa-4469-8039-9b88d8d9ceb2">
  <DataSourceID>.</DataSourceID>
  <CubeID>Vms</CubeID>
  <CubeDimensionID>History Numeric</CubeDimensionID>
  <Filter></Filter>
</Source>
<Columns>
  <Column  xsi:type="ScalarMiningStructureColumn"  dwd:design-time-
name="f4d1358b-2703-48cd-9970-dff92eb3e080">
    <ID>UTC Date Time</ID>
    <Name>UTC Date Time</Name>
    <IsKey>true</IsKey>
    <Type>Date</Type>
    <Source      xsi:type="CubeAttributeBinding"      dwd:design-time-
name="e58c625d-2292-44ba-8d16-331dad824936">
      <CubeID>Vms</CubeID>
      <CubeDimensionID>History Numeric</CubeDimensionID>
      <AttributeID>UTC Date Time</AttributeID>
      <Type>All</Type>
    </Source>
    <Content>Key</Content>
  </Column>
  <Column  xsi:type="ScalarMiningStructureColumn"  dwd:design-time-
name="6d476e70-49f7-4b09-a261-cccdf16f3224">
    <ID>Speed</ID>
    <Name>Speed</Name>
    <Type>Double</Type>
    <Source  xsi:type="MeasureBinding"  dwd:design-time-name="c90956cb-
c7a1-410e-a518-df871d636896">
      <MeasureID>Speed</MeasureID>
    </Source>
    <Content>Discretized</Content>
  </Column>

```

```

</Columns>
<MiningModels>
  <MiningModel          dwd:design-time-name="67f89793-441d-4523-afe2-
1c1324ba1d4f">
    <ID>History Numeric 1</ID>
    <Name>History Numeric 1</Name>
    <CreatedTimestamp>0001-01-01T00:00:00Z</CreatedTimestamp>
    <LastSchemaUpdate>0001-01-01T00:00:00Z</LastSchemaUpdate>
    <LastProcessed>0001-01-01T00:00:00Z</LastProcessed>
    <State>Unprocessed</State>
    <Algorithm>Microsoft_Association_Rules</Algorithm>
    <Columns>
      <Column          dwd:design-time-name="799461c9-5ec8-4608-a779-
eb9a23442e11">
        <ID>UTC Date Time</ID>
        <Name>UTC Date Time</Name>
        <SourceColumnID>UTC Date Time</SourceColumnID>
        <Usage>Key</Usage>
      </Column>
      <Column          dwd:design-time-name="ee52eb02-f4ce-48e6-a69e-
6624edce029b">
        <ID>Speed</ID>
        <Name>Speed</Name>
        <SourceColumnID>Speed</SourceColumnID>
        <Usage>Predict</Usage>
      </Column>
    </Columns>
  </MiningModel>
</MiningModels>
<ddl100_100:HoldoutMaxPercent>99</ddl100_100:HoldoutMaxPercent>
</MiningStructure>

```