



Universidad de las Ciencias Informáticas  
Facultad 6

Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas:



Título: Desarrollo del sistema para la captura y almacenamiento en tiempo real de tramas GPS para el centro GEYSED.

Autor: Yunier Aguilera Betancourt  
Tutor: Ing. Claritza Sánchez Veranes

La Habana, Junio 2013  
"Año 55 de la Revolución"



*“Es que todos los derechos del hombre son los derechos de la Revolución. Es que la libertad de opinar, la libertad de escribir, la libertad de hablar, la libertad de reunirse y la libertad de creer, son libertades sagradas de nuestra Revolución”*

A handwritten signature in black ink, which appears to be 'Fidel Castro'. The signature is stylized and fluid, with a long horizontal line extending from the end of the name.

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Centro Geysed de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Yunier Aguilera Betancourt

Claritza Sánchez Veranes

---

Autor

---

Tutor

## **Datos de contacto**

**Autor:** Yunier Aguilera Betancourt.

**Centro Estudiantil:** Universidad de las Ciencias Informáticas (UCI).

**Correo Electrónico:** [yabetancourt@estudiantes.uci.cu](mailto:yabetancourt@estudiantes.uci.cu)

**Tutor:** Ing. Claritza Sánchez Veranes.

**Formación Académica:** Ingeniera en Ciencias Informáticas.

**Centro Laboral:** Universidad de las Ciencias Informáticas (UCI).

**Correo Electrónico:** [csanchezv@uci.cu](mailto:csanchezv@uci.cu)

# Agradecimientos

---

## Agradecimientos

*Primeramente quiero agradecerle a mi adorable madre, por ser lo más grande que tengo en la vida, gracias mami por tu amor incondicional, por estar a mi lado, porque sin ti este día nunca hubiese llegado, por ser la mujer luchadora y trabajadora que siempre has sido.*

*A mi familia por todo el amor y el apoyo incondicional que me han brindado a lo largo de estos años, en especial a mi viejuca Margot, que ha sido mi segunda madre, a mi primo Iván Marichal y a mi tío Tomas Marichal, los cuales han sido padres para mí y todo un modelo a seguir, a Yarelis, por su gran corazón y apoyo, a la mejor madrina del mundo, para ti Migdalia, a mi papá Tomasito, a mi tío Manolito, a mi tía Pancha y a mi gordita hermosa Martha Lucia por haber confiado siempre en mí. A mi tía Hildita y a sus hijas Anabel y Annia.*

*A mi tutora Claritza Sánchez Veranes por su apoyo sin medida mediante todo este tiempo para el desarrollo exitoso de la presente investigación, por su tiempo y su preocupación.*

*A los miembros del tribunal por sus sugerencias, lo que permitió que este trabajo tuviera mayor calidad.*

*A los profesores Liester Cruz Castro y Aimé Esther Guzmán.*

*A mis abuelos, a mi hermana y al resto de mi familia que es bastante larga pero no por eso se les quiere menos.*

*A todos los amigos que compartieron conmigo estos inolvidables años universitarios, en especial a Jesús A. Caballero, Yunier López (cabeza), Carlos Oviedo, Erick, Karel Piorno, Yadir Hernández, Aldo Scull, Dailenka Mejías, Julio C. Rodríguez, Yainelis Gallardo y Yensaire (Yeni).*

*A todas mis amistades del barrio, en especial a Ernesto, Netty, Luigi, Carlos, Bonal, Yudith y Emilito, los cuales más que amigos para mí son como hermanos.*

*A todos muchas gracias de todo corazón.*

# *Dedicatoria*

---

## **Dedicatoria**

*A mi madre: por su cariño, su amor, su confianza, por ser la mujer más maravillosa de este mundo, por inculcarme ser mejor cada día y superarme siempre, por todo su esfuerzo y dedicación le dedico este trabajo.*

## **Resumen**

En la actualidad el mundo se encuentra afrontando diversos avances tecnológicos, los cuales surgen y revolucionan constantemente. Uno de los aspectos más significativos a tratar es el desarrollo de aplicaciones informáticas destinadas al procesamiento y almacenamiento de información espacial. Dentro de las mismas han surgido los Sistemas de Información Geográfica (SIG), los cuales permiten almacenar y analizar datos espaciales además de servir de apoyo para la toma de decisiones en esferas como el transporte. Otras de las tecnologías surgidas son los Sistemas de Posicionamiento Global (GPS), los cuales permiten obtener la posición en el espacio tanto de vehículos como de objetos o personas. En la actualidad numerosas empresas han optado por utilizar estas tecnologías para optimizar sus procesos, como puede ser el proceso de control de flotas.

En el presente trabajo se exponen los resultados de todo el proceso de investigación, que cumple con el objetivo de desarrollar el Sistema para la captura y almacenamiento en tiempo real de tramas GPS para el centro GEYSED, para de esta manera gestionar los datos de las tramas GPS emitidas por dispositivos móviles, además de la inserción en una Base de Datos de grandes cantidades de tramas en poco tiempo manteniendo la integridad de los datos, en cualquier sistema control de flotas. Finalmente se le realizaron pruebas al sistema para avalar el correcto funcionamiento del mismo, además de quedar descritas recomendaciones que posibilitarán la continuidad y perfeccionamiento para su uso futuro.

**Palabras Claves:** SIG, GPS, control de flotas, tramas GPS, Base de Datos.

# Índice de figuras y tablas

---

<b>Fig. 1 Estructura de RUP.....</b>	<b>21</b>
<b>Fig.2 Características de PostgreSQL. ....</b>	<b>25</b>
<b>Fig.3 Resultados para consultas complejas.....</b>	<b>26</b>
<b>Fig. 4 Trama GPS en formato NMEA.....</b>	<b>30</b>
<b>Fig. 5 Estructura sentencia RMC. ....</b>	<b>31</b>
<b>Fig. 6 Diagrama Modelo de Dominio.....</b>	<b>33</b>
<b>Fig. 7 Diagrama de Casos de Uso del Sistema.....</b>	<b>37</b>
<b>Fig. 8 Imagen que representa el modelo Cliente-Servidor.....</b>	<b>43</b>
<b>Fig. 9 Diagrama de Clases del Diseño.....</b>	<b>46</b>
<b>Fig. 10 Diagrama Entidad-Relación. ....</b>	<b>48</b>
<b>Fig. 11 Diagrama de despliegue.....</b>	<b>49</b>
<b>Fig. 12 Diagrama de Componentes.....</b>	<b>50</b>
<b>Tabla 1. Descripción de los actores del sistema.....</b>	<b>36</b>
<b>Tabla 2. Descripción textual del CUS Verificar formato de trama GPS. ....</b>	<b>39</b>
<b>Tabla 3. Descripción textual del CUS Validar datos de trama GPS.....</b>	<b>40</b>
<b>Tabla 4. Descripción textual del CUS Insertar datos de tramas GPS. ....</b>	<b>67</b>



# Índice de contenido

---

<b>Agradecimientos</b> .....	<b>I</b>
<b>Dedicatoria</b> .....	<b>II</b>
<b>Resumen</b> .....	<b>III</b>
<b>Introducción</b> .....	<b>1</b>
<b>Capítulo 1: Fundamentación Teórica del sistema para la captura y almacenamiento en tiempo real de tramas GPS para el centro GEYSED</b> .....	<b>6</b>
1.1 Introducción.....	6
1.2 Conceptos asociados al dominio del problema.....	6
1.3 Descripción actual del problema .....	10
1.4 Análisis de mecanismos existentes para el control de concurrencia en los principales Sistemas Gestores de Base de Datos.....	11
1.4.1 Oracle.....	11
1.4.2 PostgreSQL.....	13
1.4.3 MySQL.....	15
1.5 Conclusiones parciales.....	16
<b>Capítulo 2: Características del sistema para la captura y almacenamiento en tiempo real de tramas de GPS para el centro GEYSED</b> .....	<b>18</b>
2.1 Introducción.....	18
2.2 Metodología de Desarrollo de Software.....	18
2.2.1 Proceso Unificado de Desarrollo .....	19
2.3 Lenguaje Unificado de Modelado (UML v2.0).....	21
2.4 Herramienta Case.....	22

# Índice de contenido

---

2.5 Sistema Gestor de Base de Datos (SGBD).....	24
2.5.1 PostgreSQL v9.1.....	24
2.5.2 PostGIS v1.5.....	26
2.6 Framework de desarrollo.....	27
2.7 Lenguaje de programación.....	28
2.7.1 C++.....	28
2.8 Data Generator for PostgreSQL v3.0.1.3.....	29
2.9 Protocolo Standard NMEA.....	29
2.9.1 Sentencia RMC del estándar NMEA.....	30
2.10 Conclusiones parciales.....	31
<b>Capítulo 3: Presentación de la solución propuesta.....</b>	<b>32</b>
3.1 Introducción.....	32
3.2 Modelo de dominio.....	32
3.3 Glosario de términos del dominio.....	33
3.4 Especificación de Requisitos.....	34
3.4.1 Requisitos Funcionales (RF).....	35
3.4.2 Requisitos No Funcionales (RNF).....	35
3.5 Descripción del sistema.....	36
3.5.1 Descripción de los actores del sistema.....	36
3.5.2 Diagrama de Casos de Uso del Sistema.....	36
3.5.3 Descripción de los Casos de Uso del Sistema.....	37
3.6 Conclusiones parciales.....	40
<b>Capítulo 4: Construcción y validación de la solución propuesta.....</b>	<b>42</b>

# Índice de contenido

---

4.1 Introducción.....	42
4.2 Arquitectura de software.....	42
4.2.1 El modelo Cliente-Servidor.....	42
4.2.2 Patrón arquitectónico en capas.....	44
4.3 Modelo de diseño.....	45
4.3.1 Diagramas de clase del diseño.....	45
4.3.2 Patrones de diseño.....	46
4.3.2.1 Patrones GRASP.....	46
4.3.2.2 Patrones GOF.....	47
4.4 Diseño de la Base de Datos.....	47
4.4.1 Diagrama Entidad-Relación.....	47
4.5 Modelo de despliegue.....	49
4.5.1 Diagrama de despliegue.....	49
4.6 Modelo de implementación.....	49
4.6.1 Diagrama de componentes.....	49
4.7 Pruebas de sistema.....	50
4.7.1 Pruebas de Caja Blanca.....	50
4.7.2 Pruebas de rendimiento.....	58
4.8 Conclusiones parciales.....	59
<b>Conclusiones.....</b>	<b>60</b>
<b>Recomendaciones.....</b>	<b>61</b>
<b>Bibliografía.....</b>	<b>62</b>
<b>Anexos.....</b>	<b>65</b>

# *Introducción*

---

## **Introducción**

Con el transcurso del tiempo, la era de la informatización se hace cada vez más potente. Surgen nuevos adelantos en la informática y cada día que transcurre se aprecia más la dependencia del hombre hacia los ordenadores. Con el avance científico y tecnológico, el desarrollo de la información y el conocimiento, han sido determinantes en los resultados económicos y beneficios sociales, debido a que han provocado un impulso de los mismos, así como de las necesidades de progreso en todas las esferas de la sociedad: dígase en la economía, como la política, la salud, los deportes, la educación, etcétera. Debido a estos avances tecnológicos se ha multiplicado considerablemente el uso de dispositivos móviles, por esa causa numerosas empresas en telefonía digital y las comunicaciones, desarrollan una varia gama de programas que integran funcionalidades para la mejora de los servicios que estos ofrecen. Un ejemplo de estos lo constituyen los Sistemas de Información Geográfica (SIG).

Un Sistema de Información Geográfica (SIG) es una integración organizada de hardware, software y datos geográficos diseñada para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y de gestión.(1) En nuestros días los Sistemas de Información Geográfica han evolucionado mundialmente a gran escala, siendo muy eficientes en campos como la salud, la geografía, el transporte, entre otros. Hoy en día, gracias a los adelantos tecnológicos han surgido tecnologías para facilitar el manejo de los datos que necesitan los Sistemas de Información Geográfica, como son las Bases de Datos y los Sistemas de Posicionamiento Global (GPS).

El Sistema de Posicionamiento Global (GPS) es un Sistema Global de Navegación por Satélite 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. (2) Además de la posición, también permite conocer la velocidad del movimiento, la orientación del desplazamiento y la traza del recorrido que se ha efectuado.

En Cuba, los fines principales de este seguimiento, está constituido por la obtención de un mayor control para el ahorro del combustible utilizado por los vehículos, de los cuales antes de este tiempo, no se tenía un control exacto de este preciado tesoro. Cuba, con un proyecto de desarrollo que tiene como pilares la justicia social, la participación popular, la equidad y la solidaridad, ha diseñado e iniciado la aplicación de estrategias que permiten convertir los conocimientos y las Tecnologías de la información y las

# Introducción

---

Comunicaciones (TICs), las cuales permiten gestionar la información en cualquier ámbito, en instrumentos a disposición del avance y las profundas transformaciones revolucionarias.

La Universidad de las Ciencias Informáticas (UCI), se encuentra entre las principales impulsoras en Cuba, de aplicaciones que usan la tecnología SIG, GPS y Base de Datos, y con el objetivo de aumentar la productividad ha creado centros de desarrollo, que son conceptualmente pequeñas empresas dirigidas a la producción de software en una rama determinada, para desplegar productos, servicios y soluciones informáticas. Una de ellas es el centro Geoinformática y Señales Digitales (GEYSED).

El centro GEYSED se encuentra con un gran obstáculo, debido a que no posee un sistema para la obtención de manera propicia de los datos del posicionamiento de dispositivos móviles, como los movimientos, la velocidad y la posición exacta de sus vehículos, lo que significa que el seguimiento de estos dispositivos no se realiza en tiempo real. Por esta razón la información es capturada de forma diferida por medios de ficheros que son proporcionados por los dispositivos.

Otras de las limitantes que presenta está constituida por las posibles sobrecargas de flujo de información en el momento de almacenar dicha información en la Base de Datos, debido a que no existe un mecanismo para controlar el acceso a la Base de Datos de manera regulada y que a su vez se brinde la información indispensable en el sistema para su correcto funcionamiento en el momento necesario. Debido a esa sobrecarga el sistema podría responder con lentitud ante las peticiones de los usuarios, y puede existir el caso de que el mismo usuario haya pedido una información a la Base de Datos y que por motivos de retraso esta aún no se encuentre insertada en la misma.

Teniendo en cuenta lo expuesto anteriormente surge el siguiente **problema a resolver**: ¿Cómo mantener actualizada, en todo momento, la información asociada a las flotas en el centro GEYSED, sin que esto afecte negativamente la concurrencia en el proceso de almacenamiento de la información?

Para darle solución al problema de la investigación identificado se presenta el siguiente **objetivo general**: Desarrollar un sistema para la captura y almacenamiento en tiempo real de la información emitida por Dispositivos Móviles para el centro GEYSED.

# Introducción

---

Basándose en el problema definido se plantea como **objeto de estudio**: Los procesos de captura y almacenamiento de información de dispositivos móviles, siendo el **campo de acción**: la captura y almacenamiento de información de tramas GPS.

Por todo lo anterior se expone la siguiente **idea a defender**: Con la elaboración de un sistema para la captura y almacenamiento en tiempo real de la información emitida por Dispositivos Móviles para el centro GEYSED se evitará la sobrecarga del flujo de información en el instante de almacenar los datos provenientes de los GPS. A partir de los elementos anteriormente expuestos, se plantean las siguientes tareas de investigación:

1. Caracterizar el formato de Tramas GPS NMEA.
2. Caracterizar la tecnología existente referente al trabajo con los dispositivos GPS y el tratamiento de la información que estos emiten.
3. Caracterizar la arquitectura base de la propuesta.
4. Elaborar la documentación técnica correspondiente.
5. Implementar la solución, según la documentación técnica generada.
6. Validar que la solución construida cumpla con los requisitos propuestos mediante el uso de técnicas para este fin.

Atendiendo a lo antes planteado se esperan como **posibles resultados**:

1. Documentación técnica asociada al proceso ingenieril.
2. Sistema para la captura y almacenamiento de información en tiempo real del posicionamiento de dispositivos móviles.

Con el desarrollo de la presente investigación científica se han utilizado un conjunto de **métodos de investigación** los cuales se muestran a continuación:

## **Métodos Teóricos:**

# Introducción

---

- **Analítico-Sintético**: Este método se utilizó para el análisis y estudio de los documentos relacionados con los dispositivos móviles, los Sistemas de Información Geográfica (SIG) y las tramas GPS, para luego sintetizarlos para la confección de la solución propuesta.
- **Histórico-Lógico**: Este método se utilizó para el estudio de los trabajos e investigaciones anteriores y coger estos como base para el desarrollo de esta investigación.

## **Métodos Empíricos:**

- **Entrevista**: Se realizaron entrevistas a profesores del antiguo proyecto Control de Flotas de la facultad 6 con el objetivo de recoger información necesaria para el desarrollo de la investigación. La entrevista se llevó a cabo de manera no estructurada, siendo más flexible y abierta, dando la posibilidad de profundizar en algún tema de interés que emergiera durante la misma.

La presente investigación científica se estructura en cuatro capítulos que a continuación se describen:

**Capítulo 1:** Fundamentación teórica del sistema para la captura y almacenamiento en tiempo real de tramas GPS para el centro GEYSED. En este capítulo se realiza un estudio acerca del manejo de la concurrencia en algunos de los principales Sistemas Gestores de Bases de Datos existentes en la actualidad, así como los principales conceptos asociados para una mayor comprensión de la situación problemática planteada.

**Capítulo 2:** Características del sistema para la captura y almacenamiento en tiempo real de tramas GPS para el centro GEYSED. En este capítulo se exponen las herramientas y tecnologías que se utilizaron para la construcción del módulo, la metodología, lenguaje de modelado, herramienta CASE más idónea para la construcción de los artefactos correspondientes a cada uno de los flujos de trabajo, entre otras.

**Capítulo 3:** Análisis y diseño del sistema para la captura y almacenamiento en tiempo real de tramas GPS para el centro GEYSED. En este capítulo se describirá todo el diseño del sistema, según la metodología escogida previamente para darle solución al problema. Se definirán los requisitos funcionales y no funcionales, se identificarán los casos de uso del sistema y se realizarán las descripciones textuales de los casos de usos.

# *Introducción*

---

**Capítulo 4:** Construcción y validación de la solución propuesta. En este capítulo se tratarán aspectos relacionados con la implementación del sistema en cuestión y las pruebas realizadas al mismo para validar su correcto funcionamiento.



# *Capítulo 1: Fundamentación Teórica*

---

## **Capítulo 1: Fundamentación Teórica del sistema para la captura y almacenamiento en tiempo real de tramas GPS para el centro GEYSED.**

### **1.1 Introducción.**

En este capítulo se presentarán los conceptos básicos asociados al dominio del problema y que se hacen necesarios para lograr una mayor comprensión de la investigación. Se realizará una detallada descripción acerca del objeto de estudio a través de sus características para lograr mayor conocimiento sobre el tema a tratar. Además se realizará un estudio acerca del proceso de manejo de la concurrencia en algunos de los principales Sistemas Gestores de Bases de Datos existentes en la actualidad.

### **1.2 Conceptos asociados al dominio del problema.**

Para lograr una mayor comprensión de los temas que serán tratados en la presente investigación, se relacionan a continuación un conjunto de conceptos asociados al dominio del problema.

#### **Sistema de Información Geográfica (SIG)**

Se define como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestres, construido para satisfacer necesidades concretas de información. En el sentido más estricto, es cualquier sistema de información capaz de integrar, almacenar, editar, analizar, compartir y mostrar la información geográficamente referenciada. Los SIG son herramientas que permiten a los usuarios crear consultas interactivas, analizar la información espacial, editar datos, mapas y presentar los resultados de todas estas operaciones. El SIG funciona como una base de datos con información geográfica (datos alfanuméricos) que se encuentra asociada por un identificador común con los objetos gráficos de un mapa digital. (3)

#### **Base de Datos**

Una base de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada ó estructurada. Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulan ese conjunto de datos. (4)

# Capítulo 1: Fundamentación Teórica

---

Entre las principales características de los sistemas de base de datos podemos mencionar:

- ✓ Redundancia mínima.
- ✓ Acceso a través de lenguajes de programación estándar.
- ✓ Respaldo y recuperación.
- ✓ Seguridad de acceso y auditoría.
- ✓ Consultas complejas optimizadas.
- ✓ Integridad de los datos.
- ✓ Acceso concurrente por parte de múltiples usuarios.
- ✓ Independencia lógica y física de los datos.

## **Sistemas Gestores de Bases de Datos (SGBD)**

Un sistema gestor de bases de datos no es más que el software que permite la utilización y/o la actualización de los datos almacenados en una o varias bases de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez. Los programas de aplicación operan sobre los datos almacenados en la base de datos utilizando las facilidades que brindan los SGBD, los que en la mayoría de los casos, poseen lenguajes especiales de manipulación de la información que facilitan el trabajo de los usuarios. (5)

Existen un conjunto de objetivos generales que deben cumplir todas los SGBD, de modo que faciliten el proceso de diseño de aplicaciones y que los tratamientos sean más eficientes y rápidos, dando la mayor flexibilidad posible a los usuarios, dentro de los cuales podemos citar:

- ✓ Independencia de los datos y los programas de aplicación.
- ✓ Minimización de la redundancia.
- ✓ Integración y sincronización de las bases de datos.

# Capítulo 1: *Fundamentación Teórica*

---

- ✓ Integridad de los datos.
- ✓ Seguridad y recuperación.
- ✓ Facilidad de manipulación de la información.
- ✓ Control centralizado.

## **Sistema de Posicionamiento Global (GPS)**

Es un sistema que permite conocer en todo momento y en cualquier punto del globo, la localización exacta de dicho punto con un margen de error del orden de unos pocos metros o menos. Para ello, se basan en el envío de señales entre un dispositivo situado en el punto concreto y una red de satélites, pudiendo establecerse la posición exacta mediante las características de dicha transmisión. El funcionamiento del sistema se basa en la triangulación de la posición mediante las señales procedentes de un cierto número de los satélites. Esta posición se calcula no únicamente en sus coordenadas x e y, sino también en z, es decir en elevación. La posición de los satélites es conocida en todo momento, y los propios satélites informan de ella a los receptores a través de los mensajes de navegación. En base a esas posiciones orbitales, el proceso de triangulación que se lleva a cabo en el sistema GPS no se basa en el trabajo con ángulos, sino con distancias. (6)

En función de sus características y de la forma en que operan, podemos distinguir los siguientes tipos de receptores GPS:

- ✓ Receptores secuenciales.
- ✓ Receptores continuos.
- ✓ Receptores con canales multiplexados.

## **Información geográfica**

Se denomina Información Geográfica a aquellos datos espaciales georreferenciados requeridos como parte de las operaciones científicas, administrativas o legales. Dichos datos espaciales suelen llevar una información alfanumérica asociada. (7)

# *Capítulo 1: Fundamentación Teórica*

---

## **Flota**

Conjunto de vehículos que realizan la misma actividad y normalmente son propiedad de una compañía: una flota de taxis. (8) Estos vehículos pueden ser tanto terrestres como aéreos o marítimos.

## **Control de flotas**

El control de flotas no es más que un grupo de acciones llevadas a cabo por una organización para aumentar la productividad del proceso de logística a un menor costo. Este control proporciona un procesamiento automático de los tiempos de conducción, descanso y servicio, tiempos de espera, informes de ruta, factor de carga o información de estado. Estos datos permiten calcular el rendimiento y optimizar la planificación logística. (9)

## **Dispositivos móviles**

Se entiende por dispositivo móvil aquel que es portátil, que puede acceder a la web y que está pensado para ser usado en movimiento. Para la mayoría de los propósitos, el término dispositivo móvil se utiliza para implicar web habilitadas para terminales celulares o teléfonos y asistentes digitales personales. (10) En el contexto de la investigación se utilizará el término dispositivos móviles para referirnos a uno o más vehículos.

## **Tramas GPS**

Para las telecomunicaciones, una trama consiste en una unidad o módulo de transmisión de información. Este concepto es similar y equivale a la idea de paquete de datos en el grado de enlazamiento de datos del modelo OSI. Está compuesta por una cabecera (que incluye campos de control de protocolo), datos (aquello que se quiere transmitir en un nivel de comunicación superior) y una cola (donde se establece un chequeo de errores). (11)

Una trama GPS no es más que un conjunto de datos que viajan mediante protocolos de comunicación, enviados del equipo GPS a la computadora y contiene la información geográfica de un objeto.

## **1.3 Descripción actual del problema**

En la actualidad el control de flotas constituye una necesidad para muchas empresas existentes en el mundo, debido al gran número de vehículos que desean controlar, con el objetivo de mantener un mayor control y poder contar con información actualizada en todo momento acerca de sus conductores y del combustible asignado a las flotas, garantizando de esta manera un ahorro considerable de dinero y tiempo.

No en todos los países del mundo se cuenta con tecnología avanzada para el control de flotas, entre estos países se encuentra Cuba, debido a que para adquirir dichas tecnologías se necesita pagar altos precios, lo que obliga a que este proceso se realice de manera manual, es decir, mediante hojas de ruta, y además de no realizarse en tiempo real para de manera más oportuna tomar decisiones en el momento. Otra de las limitaciones que puede presentar este proceso, es la poca confiabilidad en el cumplimiento de las rutas y consumo de combustible asignado a los vehículos, ya que si los datos no son ciertos, entonces la toma de decisiones estará basada en datos ficticios y sus resultados no se ajustarán con la realidad. El proceso de gestión de la información para el control de flotas sería más eficiente si estuviera acompañado de un SIG, de esta manera se tendría un mejor acceso a toda la información, además de poder brindar la posibilidad de visualizar el proceso.

Debido a que los datos del posicionamiento de los dispositivos móviles, como los movimientos, la velocidad y la posición exacta no se obtienen de manera propicia, el seguimiento de estos no se realiza en tiempo real. Por esta razón la información es capturada de forma diferida por medios de ficheros que son proporcionados por los dispositivos.

Al mismo tiempo existe sobrecargas de flujo de información en el momento de almacenar y acceder a dicha información en la Base de Datos, debido a que no existe un mecanismo para controlar el acceso a la Base de Datos de manera regulada y que a su vez se brinde la información indispensable en el sistema para su correcto funcionamiento en el momento necesario. Debido a esa sobrecarga el sistema responde con lentitud ante las peticiones de los usuarios.

# Capítulo 1: *Fundamentación Teórica*

---

## **1.4 Análisis de mecanismos existentes para el control de concurrencia en los principales Sistemas Gestores de Base de Datos.**

Una preocupación primaria de un sistema de gestión de base de datos es cómo controlar la concurrencia, que es el acceso simultáneo a los mismos datos por muchos usuarios. Sin controles adecuados de concurrencia, los datos pueden ser modificados, poniendo en peligro la integridad de los mismos y además propiciando posibles caídas del sistema por la gran cantidad de usuarios conectados a la vez realizando peticiones al sistema. A continuación se explica cómo manejan la concurrencia algunos de los principales sistemas gestores de bases de datos, centrándonos en estos sistemas ya que aquí es donde está la esencia del problema a resolver y no en los sistemas de software para el control de las flotas. Cabe destacar que se realizó un estudio del arte de los sistemas controles de flotas que utilizan la tecnología GPS pero no se encontró ningún elemento que mostrara como los mismos manejan el problema de la concurrencia.

### **1.4.1 Oracle.**

Oracle maneja la concurrencia de los datos y la integridad entre las transacciones utilizando sus mecanismos de bloqueo y un modelo de consistencia multiversión. Debido a que los mecanismos de bloqueo de Oracle están vinculados estrechamente al control de la transacción, los diseñadores de aplicaciones deben definir solamente las transacciones correctamente, y Oracle gestiona automáticamente el bloqueo.

Para describir el comportamiento de transacción consistente cuando las transacciones se ejecutan al mismo tiempo, se ha definido un modelo de aislamiento de transacción llamado serialización. El modo de comportamiento de la transacción serializable trata de asegurar que las transacciones se ejecuten de tal forma que parezcan ser ejecutadas de una en una, en vez de simultáneamente.

Oracle permite una transacción serializable para modificar una fila de datos sólo si se puede determinar que los cambios previos a la fila fueron realizados por transacciones que han finalizado cuando la transacción serializable comenzó. Para hacer este proceso eficiente, Oracle utiliza la información de control almacenada en bloques de datos, los cuales indican qué filas en el bloque contienen cambios confirmados y no confirmados. En este sentido, el bloque contiene un historial reciente de las transacciones que afectaron a cada fila.

# Capítulo 1: Fundamentación Teórica

---

## Fenómenos prevenibles y niveles de aislamiento de transacción (Preventable Phenomena and Transaction Isolation Levels).

El estándar ANSI / ISO (SQL92) define cuatro niveles de aislamiento de transacciones con diferentes grados de impacto en el rendimiento de procesamiento de la transacción. Estos niveles de aislamiento se definen en términos de tres fenómenos que se deben evitar entre las operaciones que se ejecutan simultáneamente:

- ✓ **Lecturas sucias**: Una transacción lee datos que han sido escritos por otra transacción que no ha finalizado todavía.
- ✓ **Lecturas no repetibles**: Una transacción relea los datos que previamente han sido leídos y descubre que otra transacción finalizada ha modificado o borrado los datos.
- ✓ **Lecturas fantasmas**: Una transacción vuelve a ejecutar una consulta que devuelve un conjunto de filas que satisfacen una condición de búsqueda y encuentra que otra transacción finalizada ha insertado filas que satisfacen la condición.

Oracle proporciona tres niveles de aislamiento:

- ✓ **Read-committed**: Nivel de aislamiento por defecto. Cada consulta de una transacción solo ve los datos que fueron confirmados antes de que la consulta comenzara.
- ✓ **Serializable transactions**: Solamente se ven los cambios realizados por transacciones confirmadas + cambios efectuados por ella misma.
- ✓ **Read-only**: Ven datos confirmados antes de empezar y no permiten modificaciones de los mismos.

## Lectura consistente, registros "deshacer" y transacciones (Read Consistency, Undo Records, and Transactions).

Para gestionar el modelo de consistencia multiversión, Oracle crea una lectura consistente de los datos cuando a una tabla se le realiza una consulta (lectura) y, simultáneamente, se actualiza (escritura). Cuando una actualización ocurre, los valores originales de los datos, cambiados por la actualización, son almacenados en los registros de "deshacer" de la base de datos. Mientras la actualización sigue siendo parte de una transacción no confirmada, cualquier usuario que luego consulte los datos modificados verá los valores de los datos originales. Solo cuando se confirma una transacción, los datos de la transacción

# Capítulo 1: Fundamentación Teórica

---

realizada se hacen permanentes. Las declaraciones que se inicien después de la transacción del usuario sólo ven los cambios realizados por la transacción confirmada.

## **Mecanismos de bloqueo (Locking Mechanisms).**

Oracle también utiliza bloqueos para resolver problemas asociados con el acceso concurrente a los datos, la consistencia y la integridad. Los bloqueos son mecanismos destinados a evitar la interacción destructiva entre los usuarios que acceden a los datos. Los bloqueos se utilizan para garantizar la coherencia y la integridad. Coherencia significa que los datos que un usuario está viendo o modificando, no se cambian por otros usuarios hasta que el usuario haya finalizado con los datos. La integridad significa que las bases de datos y las estructuras reflejan todos los cambios realizados en la secuencia correcta. Además garantizan la integridad de los datos al tiempo que permite el máximo acceso concurrente a los mismos por un número ilimitado de usuarios.

## **Bloqueo automático (Automatic Locking).**

Los bloqueos en Oracle se realizan de forma automática y no requiere de ninguna acción por parte del usuario. El bloqueo implícito se produce para las sentencias de SQL como sea necesario, dependiendo de la acción solicitada. El administrador de bloqueos en Oracle automáticamente bloquea los datos a nivel de fila. Dicho administrador mantiene varios tipos de bloqueo de filas en dependencia del tipo de operación establecida por el bloqueo. Los dos tipos generales de bloqueos son exclusivos (exclusive) y compartidos (share). Sólo un bloqueo exclusivo se puede colocar en un recurso (como una fila o una tabla), sin embargo, muchos bloqueos compartidos pueden ser colocados en un único recurso. Ambos tipos de bloqueos siempre permiten consultas en el recurso bloqueado pero prohíben otras actividades en el mismo recurso (como las actualizaciones y eliminaciones).

## **Interbloqueos (Deadlocks).**

Puede ocurrir cuando uno o más usuarios están esperando por datos bloqueados por los otros usuarios. Oracle automáticamente detecta situaciones de punto muerto y las resuelve revirtiendo una de las declaraciones que participan en el interbloqueo, liberando así un conjunto de filas bloqueadas. (12)

## **1.4.2 PostgreSQL.**

### **Control de Concurrencia Multi-Versión (MVCC).**



# Capítulo 1: Fundamentación Teórica

---

PostgreSQL proporciona un completo conjunto de herramientas para desarrolladores para gestionar el acceso simultáneo a los datos. Internamente, los datos mantienen la coherencia con un modelo multiversión (control de concurrencia multiversión, MVCC). Esto significa que mientras se consulta una base de datos, cada transacción ve una instantánea de los datos (una versión de la base de datos), independientemente del estado actual de los datos subyacentes. Esto protege a la transacción de la visualización de datos incoherentes que podría ser causado por otra u otras actualizaciones de transacciones concurrentes en la misma fila de datos, proporcionando un aislamiento de transacción para cada sesión de base de datos.

En PostgreSQL, se puede solicitar cualquiera de los cuatro niveles de aislamiento de transacciones estándar. Pero en el interior, hay solo tres niveles de aislamiento distintos, que corresponden a los niveles de lectura confirmada (Read Committed), lectura repetible (Repeatable Read) y Serializable.

Las facilidades de bloqueos a nivel de tabla y de fila también se encuentran disponibles en PostgreSQL para aplicaciones que generalmente no necesitan un completo aislamiento de la transacción y prefieren manejar explícitamente determinados puntos de conflicto. Sin embargo, un uso adecuado de MVCC generalmente proporciona un mejor rendimiento que los bloqueos.

## **Bloqueos explícitos (Explicit Locking).**

PostgreSQL ofrece varios modos de bloqueo para controlar el acceso concurrente a los datos en las tablas. Estos modos pueden ser usados por la aplicación para controlar los bloqueos en situaciones en que el MVCC no muestra el resultado esperado. Además, la mayoría de los comandos de PostgreSQL automáticamente adquieren bloqueos de modos apropiados para asegurarse de que las tablas de referencia no se eliminen o se modifiquen de manera incompatible mientras se ejecuta el comando.

## **Bloqueos a nivel de tabla (Table-level Locks).**

La siguiente lista muestra los modos de bloqueos a nivel de tabla disponibles por PostgreSQL.

- ✓ Access Share
- ✓ Row share
- ✓ Row exclusive
- ✓ Share Update Exclusive
- ✓ Share Row Exclusive

# Capítulo 1: Fundamentación Teórica

---

- ✓ Access Exclusive
- ✓ Share
- ✓ Exclusive

## **Bloqueos a nivel de fila (Row-level Locks).**

Además de los bloqueos a nivel de tabla, existen bloqueos a nivel de fila, que pueden ser también bloqueos exclusivos o compartidos. Un bloqueo exclusivo en una fila específica se adquiere automáticamente cuando las filas se actualizan o se eliminan. El bloqueo se mantiene hasta que la transacción se confirma o se deshace, como los bloqueos a nivel de tabla. Los bloqueos a nivel de fila no afectan a los datos de la consulta, sino que sólo bloquean a los escritores de la misma fila. Una vez que el bloqueo a nivel de fila se adquiere, la transacción puede actualizar la fila varias veces sin temor a que ocurran conflictos.

## **Interbloqueos (Deadlocks).**

El uso de bloqueos explícitos puede aumentar la probabilidad de interbloqueos, cuando dos o más transacciones necesitan obtener bloqueos que otras transacciones poseen. Un ejemplo se manifiesta cuando una transacción posee un bloqueo exclusivo en la tabla A pero solicita un bloqueo exclusivo de la tabla B, encontrándose otra transacción con un bloqueo exclusivo de la tabla B y necesitando un bloqueo exclusivo de la tabla A. PostgreSQL detecta automáticamente la situación de interbloqueo y la resuelve abortando una de las transacciones en cuestión, permitiéndole a la otra completarse satisfactoriamente.

(13)

### **1.4.3 MySQL.**

#### **Control de concurrencia en MySQL.**

MySQL maneja bloqueo a nivel de fila. Los bloqueos a nivel de fila son mucho más eficaces cuando se necesita realizar una gran cantidad de inserciones o actualizaciones en la tabla. Sin embargo, los mismos solo están disponibles para los tipos de tabla de transacción segura (BDB e InnoDB). MySQL incorpora la función de bloqueo a nivel de tablas, que está disponible para todos los tipos de tabla. Existen dos tipos de bloqueos de tabla: los bloqueos de lectura y los bloqueos de escritura. Los bloqueos de lectura solo permiten realizar lecturas sobre la tabla, quedando bloqueadas las operaciones de escritura.

# Capítulo 1: Fundamentación Teórica

---

## Bloqueo InnoDB.

InnoDB implementa un bloqueo a nivel de fila estándar, donde hay dos tipos de bloqueos:

- ✓ **Compartido:** le permite a una transacción leer datos de una fila.
- ✓ **Exclusivo:** le permite a una transacción actualizar o eliminar datos de una fila.

Si una transacción A sostiene un bloqueo exclusivo (X) sobre una tupla t, entonces una solicitud de otra transacción B para establecer un bloqueo de cualquier tipo sobre la tupla no puede ser atendida inmediatamente. En lugar de eso, la transacción B debe esperar a que la transacción A libere el bloqueo exclusivo en la tupla.

InnoDB implementa semáforos a nivel de fila. Pero utiliza un sistema de control de concurrencia denominado MVCC (Multi-Version Concurrency Control) que reduce el tiempo de gestión. Adicionalmente, InnoDB soporta bloqueo de granularidad múltiple (multiple granularity locking), el cual permite que existan simultáneamente bloqueos en registros y bloqueos en tablas enteras. Para hacer práctico el nivel de bloqueo de granularidad múltiple, se emplean tipos adicionales de bloqueo, llamados bloqueos de intención (intention locks). Los bloqueos de intención son bloqueos de tabla en InnoDB. La idea detrás de los mismos es que una transacción indique qué tipo de bloqueo (compartido o exclusivo) requerirá más tarde sobre una fila de esa tabla. En InnoDB se utilizan dos tipos de bloqueos de intención (asumiendo que la transacción T ha solicitado un bloqueo del tipo indicado en la tabla R):

- ✓ **Intención compartida:** La transacción T trata de establecer bloqueos compartidos(S) en tuplas individuales de la tabla R.
- ✓ **Intención exclusiva:** La transacción T trata de establecer bloqueos exclusivos en todas las tuplas de la tabla.

## 1.5 Conclusiones parciales.

Luego de haberse dado a conocer algunos conceptos para lograr una mejor comprensión sobre la investigación y haberse realizado un estudio acerca de los mecanismos para el control de concurrencia en los principales gestores de bases de datos, se llega a la conclusión de que todos los estudiados poseen mecanismos para controlar la concurrencia de transacciones en aras de mantener la integridad de los

## *Capítulo 1: Fundamentación Teórica*

---

datos. No obstante, cabe destacar, que se desarrollará un algoritmo que sirva de apoyo en el proceso de inserción de grandes cantidades de información provenientes de los GPS, de manera regulada en la Base de Datos, para de esta manera lograr una inserción ordenada en pequeños intervalos de tiempo y asegurar que la información pueda ser accedida por muchos usuarios sin que se vean afectados por el alto volumen de información arribada al sistema suministrada por los GPS.

# *Capítulo 2: Características del sistema.*

---

## **Capítulo 2: Características del sistema para la captura y almacenamiento en tiempo real de tramas GPS para el centro GEYSED.**

### **2.1 Introducción.**

En este capítulo se explica detalladamente el uso de las tecnologías y herramientas a utilizar en el desarrollo del sistema para la captura en tiempo real de tramas GPS para el centro GEYSED. Se definirá cual será la metodología que guiará el proceso de desarrollo del sistema, así como el lenguaje de modelado para lograr un mayor entendimiento del proceso a través de diagramas, haciendo uso de las herramientas CASE<sup>1</sup>. Además se definirá el sistema gestor de base de datos, el lenguaje de programación a utilizar y el entorno de desarrollo donde se desarrollará la solución que se desea ofrecer.

### **2.2 Metodología de Desarrollo de Software.**

Una metodología de desarrollo de software es una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto (software). (14) En el proceso de desarrollo de software la metodología define quién debe hacer qué, cuándo y cómo debe hacerlo para obtener los distintos productos parciales y finales. Una metodología de desarrollo de software tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo.

Existen dos clasificaciones de las metodologías, ágiles o ligeras y tradicionales o robustas. Las metodologías ágiles permiten incorporar cambios sustanciales con rapidez en el desarrollo de software sin que se afecte el desarrollo del producto. Dichas metodologías están destinadas a pequeños proyectos, con una documentación corta y centrada en lo fundamental. Además el cliente forma parte del equipo de desarrollo, lo que significa que para tener éxito debe haber una interacción constante entre ambas partes. Las metodologías tradicionales o robustas, se centran en la definición detallada de los procesos, tareas y herramientas a utilizar, y requiere una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodologías son más eficaces y necesarias cuanto mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear, donde una gran organización es requerida. (14)

---

<sup>1</sup> CASE (Computer Aided Software Engineering): Aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.

## *Capítulo 2: Características del sistema.*

---

### **2.2.1 Proceso Unificado de Desarrollo**

El Proceso Unificado es un proceso de desarrollo de software. Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyectos. (15)

RUP es un proceso para el desarrollo de un proyecto de un software que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto. Su propósito es asegurar que la producción de software de alta calidad se ajuste a las necesidades de sus usuarios finales con unos costos y calendario predecibles.

Las características principales de RUP son:

- ✓ **Guiado por casos de uso:** Un caso de uso es una facilidad que el software debe proveer a sus usuarios. Los casos de uso reemplazan la antigua especificación funcional tradicional y constituyen la guía fundamental establecida para las actividades a realizar durante todo el proceso de desarrollo incluyendo el diseño, la implementación y las pruebas del sistema.
- ✓ **Centrado en la arquitectura:** La arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas software, sistemas operativos, manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. Es como una radiografía del sistema, lo suficientemente completa como para que todos los implicados en el desarrollo tengan una idea clara de qué es lo que están construyendo, pero lo suficientemente simple como para que si quitamos algo una parte importante del sistema quede sin especificar.
- ✓ **Iterativo e incremental:** Para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini proyecto cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo. (16)

RUP divide el proceso en cuatro fases y define seis flujos de trabajos de ingeniería y tres flujos de trabajo de apoyo.

## *Capítulo 2: Características del sistema.*

---

### **Fases de RUP:**

- ✓ **Inicio:** Durante esta fase se desarrolla una descripción del producto final a partir de una buena idea y se presenta el análisis del negocio para el producto. Se identifican y se priorizan los riesgos más importantes y se estima el proyecto de manera aproximada.
- ✓ **Elaboración:** Durante esta fase se especifican en detalle la mayoría de los casos de usos del producto y se diseña la arquitectura del sistema. Además se realizan los casos de uso más críticos que se identificaron en la fase de Inicio.
- ✓ **Construcción:** En esta fase se crea el producto, la línea base de la arquitectura crece hasta convertirse en el sistema completo.
- ✓ **Transición:** Esta fase cubre el período durante el cual el producto se convierte en una versión beta. Conlleva actividades como la formación del cliente, el proporcionar una línea de ayuda y asistencia y la corrección de los defectos que se encuentren tras la entrega. (16)

## Capítulo 2: Características del sistema.

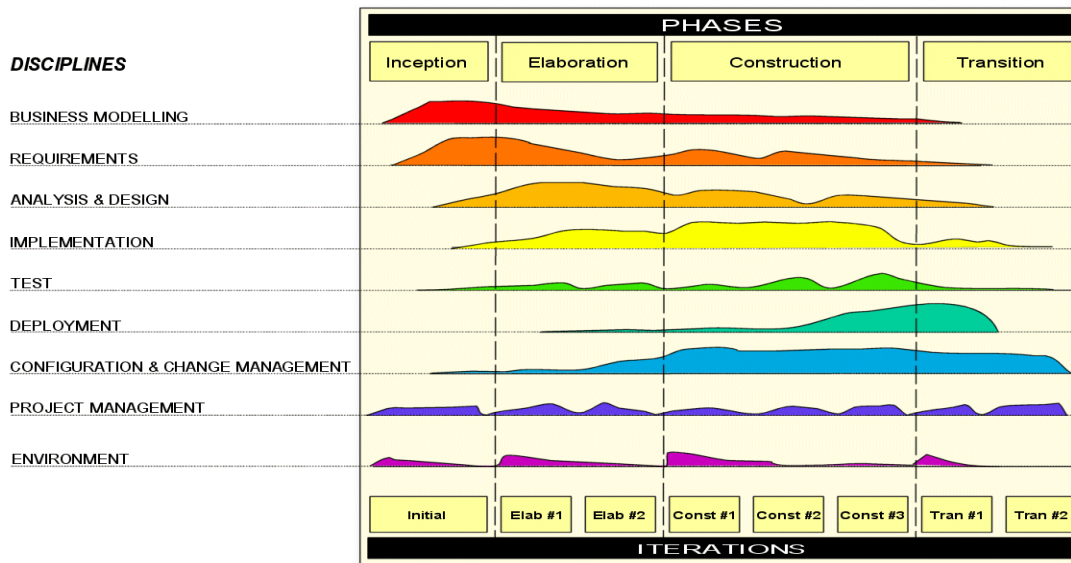


Fig. 1 Estructura de RUP. (37)

Atendiendo a las características antes mencionadas se decidió utilizar la metodología RUP ya que se necesita una documentación en cada fase que es necesaria para el desarrollo del software. También el uso de RUP permite guiar el proceso de desarrollo, trazando tareas organizadas jerárquicamente, además que se obtienen artefactos que tributan al desarrollo del software. RUP reduce el riesgo de no sacar el producto en el calendario previsto y presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores. También se selecciona debido a que los requisitos están definidos claramente desde el comienzo y no sufrirán cambios posteriores. Además el cliente no forma parte del equipo de desarrollo, lo que significa que para tener éxito no debe haber una interacción constante entre ambas partes. Hay que tener en cuenta que esta metodología también es la definida por el proyecto.

### 2.3 Lenguaje Unificado de Modelado (UML v2.0).

Desde los inicios de la informática, se han estado utilizando distintas formas de representar los diseños de una forma más bien personal o con algún modelo gráfico. La falta de estandarización en la manera de



## Capítulo 2: Características del sistema.

---

representar gráficamente un modelo impedía que los diseños gráficos realizados se pudieran compartir fácilmente entre distintos diseñadores.

Se necesitaba por tanto un lenguaje no sólo para comunicar las ideas a otros desarrolladores, sino también para servir de apoyo en los procesos de análisis de un problema. Con este objetivo se creó el **Lenguaje Unificado de Modelado** (UML: Unified Modeling Language). UML se ha convertido en ese estándar tan ansiado para representar y modelar la información con la que se trabaja en las fases de análisis y especialmente de diseño.

UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y unas reglas para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema.

Entre los principales objetivos de UML se encuentran:

- ✓ **Visualizar:** UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- ✓ **Especificar:** UML permite especificar cuáles son las características de un sistema antes de su construcción.
- ✓ **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.
- ✓ **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión. (17)

Teniendo en cuenta las características mostradas anteriormente, se optó por utilizar UML como lenguaje de modelado para el desarrollo de la aplicación, debido a que posee una notación gráfica muy expresiva que permite representar en mayor o menor medida todas las fases de un proyecto informático desde el análisis con los casos de uso hasta la implementación y configuración con los diagramas de despliegue. Brinda mayor rigor en las especificaciones, permite realizar una verificación y validación del modelo realizado, además de permitir generar código a partir de los modelos y a la inversa.

### 2.4 Herramienta Case.

Las herramientas CASE (Computer Aided Systems Engineering ) cuyo significado en español es ingeniería de sistemas asistida por ordenador, es la aplicación de tecnologías informáticas a las actividades, las técnicas y las metodologías propias de desarrollo de sistemas, con el objetivo de

## Capítulo 2: Características del sistema.

---

automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas. (18) Además estas herramientas ayudan en aspectos del ciclo de vida de desarrollo de un software en tareas como el proceso de realizar un diseño del proyecto, cálculo de costes, implementación de parte del código automáticamente con el diseño dado, entre otras. En el mundo, actualmente se encuentran disponibles varias herramientas CASE para el modelado de software utilizando UML, destacándose entre ellas Visual Paradigm y Rational Rose.

### 2.4.1 Visual Paradigm for UML v8.0 Enterprise Edition v5.0.

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Permite representar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Presenta licencia gratuita y comercial. Es fácil de instalar, actualizar, compatible entre ediciones y además es multiplataforma (19). Entre sus principales características se encuentran:

- ✓ Soporte de UML versión 2.0.
- ✓ Ingeniería inversa - Código a modelo, código a diagrama.
- ✓ Generación de código - Modelo a código, diagrama a código.
- ✓ Generación de bases de datos - Transformación de diagramas de Entidad-Relación en tablas de base de datos.
- ✓ Ingeniería inversa de bases de datos - Desde Sistemas Gestores de Bases de Datos (DBMS) existentes a diagramas de Entidad-Relación.
- ✓ Importación y exportación de ficheros XMI<sup>2</sup>.
- ✓ Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.

Con la utilización de esta herramienta se hace más factible el trabajo al equipo de desarrollo. Por las ventajas citadas anteriormente se decide utilizarla, además porque ayuda a la construcción de aplicaciones con mayor calidad, permite exportar los diagramas en formatos de imagen, permite ajustar

---

<sup>2</sup> XMI: Especificación para el intercambio de diagramas entre diferentes herramientas CASE.

## *Capítulo 2: Características del sistema.*

---

rápido los cambios efectuados en los modelos realizados, posee licencia libre y comercial, existe amplia documentación sobre la herramienta, es multiplataforma y los diagramas se encuentran clasificados por categorías, ofreciendo así una mayor organización a la hora de trabajar.

### **2.5 Sistema Gestor de Base de Datos (SGBD).**

Un Sistema Gestor de Base de Datos puede definirse de la siguiente manera: Un software que proporciona servicios para la creación, el almacenamiento, el procesamiento y la consulta de la información almacenada en bases de datos de forma segura y eficiente. Un SGBD actúa como un intermediario entre las aplicaciones y los datos. (20) Existen en la actualidad varios SGBD, entre los cuales se destacan Oracle, MySQL y PostgreSQL.

#### **2.5.1 PostgreSQL v9.1.**

PostgreSQL es el más avanzado servidor de base de datos de código abierto. PostgreSQL utiliza un modelo cliente/servidor de comunicación. Un servidor PostgreSQL está continuamente corriendo, esperando peticiones de los clientes, el servidor procesa la solicitud y devuelve el resultado al cliente. (21)

Funciona en los principales sistemas operativos, incluyendo Linux, UNIX, Mac OS X y Windows. Tiene soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados. Esto incluye la mayor parte de SQL: 2008 tipos de datos. También soporta almacenamiento de objetos binarios grandes, como imágenes, sonidos o vídeo. Cuenta con interfaces nativas de programación para C, C++, Java, .NET, Perl, Python, Ruby, ODBC<sup>3</sup>, entre otros, y una documentación excepcional. PostgreSQL cuenta con características avanzadas como Control de Concurrencia Multi-Version (MVCC), puntos en el tiempo de recuperación y un optimizador de consultas. Puede manejar grandes volúmenes de datos además de permitir un gran número de conexiones a la aplicación. (22) Este número condicionado por las características de hardware del propio servidor de Bases de Datos. Además permite que se le incorpore la extensión PostGIS para el manejo de datos espaciales. A continuación se presenta una figura con otras características notables de PostgreSQL:

---

<sup>3</sup> ODBC (Open Database Connectivity): El objetivo de ODBC es hacer posible acceder a cualquier dato desde cualquier aplicación, sin importar qué sistema de gestión de bases de datos (DBMS) almacene los datos.

## Capítulo 2: Características del sistema.

---

Límite	Valor
Máximo tamaño base de dato	Ilimitado (Depende de tu sistema de almacenamiento)
Máximo tamaño de tabla	32 TB
Máximo tamaño de fila	1.6 TB
Máximo tamaño de campo	1 GB
Máximo número de filas por tabla	Ilimitado
Máximo número de columnas por tabla	250 - 1600 (dependiendo del tipo)
Máximo número de índices por tabla	Ilimitado

Fig.2 Características de PostgreSQL. (22)

En PostgreSQL el parámetro **max\_connections** indica el número máximo de conexiones concurrentes a la base de datos. Por defecto su valor es 100 pero podría ser menor o mayor en dependencia de las características de hardware del servidor. El aumento de este parámetro puede hacer que PostgreSQL solicite más memoria compartida del sistema operativo.

PostgreSQL trabaja bajo el concepto de “*process per user*”, lo que significa que se le asigna un solo proceso cliente a cada conexión. Un sistema multiproceso consume entre 500 Kb y 1Mb por conexión. (40)

Uno de los problemas al cual las personas que desarrollan sistemas se tiene que enfrentar es elegir el sistema de gestión de base de datos que muestre mejor rendimiento y ofrezca menor tiempo de respuesta a la hora de ejecutar las consultas.

A continuación se muestra un estudio realizado por Ramón Aliendre, Juan Paz y Josefina Juaniquina en cuanto al rendimiento entre PostgreSQL, Oracle y SQL Server donde realizaron pruebas con diferentes cantidades de registros (100 000, 500 000 y 1 000000), y las pruebas presentaban diferentes tipos de consulta (simple, media y compleja). Para esta prueba se creó una base de datos con cinco tablas y se fueron analizando los resultados obtenidos a partir del tipo de consulta y la cantidad de registros. (35)



Fig.3 Resultados para consultas complejas. (35)

En la figura anterior se muestra el resultado obtenido para la consulta compleja, en la misma se observa que el SGBD Oracle es muy robusto y rápido, pero además es visible que PostgreSQL cuanto mayor es el número de registros y más compleja es la consulta sus resultados son más rápidos.

### 2.5.2 PostGIS v1.5.

PostGIS agrega soporte para objetos geográficos. En efecto, PostGIS habilita espacialmente al servidor PostgreSQL, lo que le permite ser utilizado como base de datos espacial para los Sistemas de Información Geográfica (SIG). Además es liberado bajo la Licencia Pública General de GNU. (23) Mediante PostGIS se pueden usar todos los objetos que aparecen en la especificación OpenGIS como son los puntos, líneas, polígonos y multilíneas.

Por las características antes mencionadas se selecciona a PostgreSQL como el SGBD más adecuado para el desarrollo de la presente investigación, además por su fácil integración con la extensión PostGIS para el manejo de información geográfica, existe la posibilidad de acceder a su código fuente sin costo alguno por ser una herramienta libre bajo licencia BSD, posee una comunidad muy amplia donde se puede encontrar información variada y actualizada a todo momento, también por su fácil administración y poseer características avanzadas como Control de Concurrencia Multiversión (MVCC), además por su excelente rendimiento a la hora de trabajar con grandes volúmenes de datos.

## *Capítulo 2: Características del sistema.*

---

### **2.6 Framework de desarrollo.**

En el desarrollo de software, un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, un framework puede incluir soporte de programas y librerías entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Un framework agrega funcionalidad extendida a un lenguaje de programación, esta automatiza muchos de los patrones de programación para orientarlos a un determinado propósito. Un framework proporciona una estructura al código y permite que los desarrolladores escriban código más entendible y mantenible. Además hace la programación más fácil, convirtiendo complejas funciones en sencillas instrucciones. (24)

#### **2.6.1 Qt 4.8.**

Qt es un marco de desarrollo integral con herramientas diseñadas para simplificar la creación de aplicaciones e interfaces de usuario para el escritorio, embebidos y plataformas móviles. Con Qt, se puede reutilizar el código de manera eficiente y es multiplataforma. El sistema modular de C++, las bibliotecas de clases, y las herramientas para programadores permiten a los desarrolladores crear aplicaciones para una plataforma, además de poder generar y ejecutar fácilmente el despliegue en otra. (25) Entre las principales características de Qt Creator se encuentran (26):

- ✓ Posee un avanzado editor de código C++.
- ✓ Soporta los lenguajes: C#/.NET, Python, Ada, Pascal, Perl, PHP y Ruby.
- ✓ Posee una GUI integrada y diseñador de formularios.
- ✓ Herramienta para proyectos y administración.
- ✓ Ayuda integrada sensible al contexto.
- ✓ Depurador visual.
- ✓ Resaltado y auto-completado de código.
- ✓ Soporte para refactorización de código.

## *Capítulo 2: Características del sistema.*

---

Qt Creator es distribuido bajo tres tipos de licencias: Qt Commercial Developer License, Qt GNU LGPL v. 2.1, Qt GNU GPL v. 3.0 y está disponible para las plataformas: Linux, Mac OSX; Windows, Windows CE, Symbian y Maemo.

Se decide usar en el desarrollo del sistema como framework de desarrollo Qt Creator por las ventajas antes mencionadas, además por la amplia comunidad online con la que se puede interactuar en busca de información actualizada, por poseer una licencia LGPL, lo que implica que los cambios a Qt deben de ser compartidos, por su integración con base de datos en PostgreSQL, combina edición, depuración, gestión de proyectos, localización y herramientas de compilación. Tiene todo lo necesario para crear programas para computadoras y Smartphone. Es multiplataforma lo que permite trabajar tanto en Windows como Linux o Mac OS. Es un producto estable, de probada calidad y existe mucha documentación sobre el tema.

### **2.7 Lenguaje de programación.**

Los lenguajes de programación son un conjunto de reglas, herramientas y condiciones que nos permiten crear programas o aplicaciones dentro de una computadora. Todo lenguaje de programación posee reglas acerca de cómo se deben escribir las sentencias y de qué forma. A su vez, estos se dividen en tres grandes grupos: los lenguajes de máquina, los de bajo nivel y los de alto nivel. (27)

#### **2.7.1 C++.**

C++ es un lenguaje de programación de propósito general diseñado para hacer que la programación sea más agradable. A excepción de pequeños detalles, C++ es un superconjunto del lenguaje de programación C. Además de las facilidades proporcionadas por C, C++ ofrece servicios flexibles y eficientes para definir nuevos tipos. Un programador puede dividir una aplicación en trozos manejables, definiendo nuevos tipos que se aproximen a los conceptos de la aplicación.

También contiene mejoras que no están directamente relacionadas con las clases, incluyendo las constantes simbólicas, la sustitución inline (petición al compilador para sustituir la llamada a la función directamente por su código) de la gestión de las funciones, los argumentos por omisión de funciones, nombres de funciones sobrecargadas y tipos de referencia. C++ y sus bibliotecas estándar están diseñadas para la portabilidad. Además las bibliotecas de C se pueden utilizar desde un programa en

## *Capítulo 2: Características del sistema.*

---

C++, y la mayoría de las herramientas que soportan la programación en C se pueden utilizar con C++. El lenguaje es seguro, expresivo y reduce la necesidad de centrarse en técnicas de bajo nivel. (28)

Se propone como lenguaje de programación para el desarrollo de la aplicación el lenguaje C++ debido a las ventajas mencionadas anteriormente. Es un lenguaje de programación orientado a objetos y que consume muy pocos recursos de hardware. Es muy potente y robusto para la creación de sistemas complejos. Es un lenguaje muy empleado a nivel mundial, con muchos años de creado lo que significa que existe mucha documentación sobre el tema para el aprendizaje del mismo.

### **2.8 Data Generator for PostgreSQL v3.0.1.3.**

Es una herramienta para la generación de datos de pruebas en Base de Datos PostgreSQL, brindando la posibilidad de guardar y editar los scripts. Permite poblar varias tablas con datos de pruebas al mismo tiempo y establecer parámetros de generación para cada tipo de dato de manera sencilla. Entre las principales características de Data Generator for PostgreSQL se encuentran:

- Interfaz de usuario amigable.
- Posibilidad de guardar y editar los datos generados en el script SQL sin ejecutar consultas en el servidor.
- Posibilidad de obtener una vista previa de los datos generados.
- Control automático sobre la integridad referencial para las tablas vinculadas.
- Posibilidad de guardar todos los parámetros de generación establecidos en la sesión del asistente actual en un archivo de configuración. (41)

### **2.9 Protocolo Standard NMEA.**

NMEA es un protocolo que se usa para la navegación tanto marítima (por la que realmente se creó) como terrestre. Una vez que un GPS sabe dónde está (conoce las coordenadas geográficas de su posición), es posible suministrar dicha información a un ordenador (o cualquier otro tipo de equipo especializado, como los "plotters" de los barcos). (29)

Los receptores GPS de comunicación se definen dentro de esta especificación. La mayoría de los programas informáticos que proporcionan información de posición en tiempo real entienden y esperan que los datos estén en formato NMEA que es el formato universal de tramas GPS, pero existen otros según la



## Capítulo 2: Características del sistema.

marca del dispositivo y el fabricante. Estos datos incluyen la completa PVT (posición, velocidad, tiempo), solución calculada por el receptor GPS. La idea de NMEA es enviar una línea de datos llamadas sentencias, que son totalmente autónomas e independientes de las otras sentencias. Cada sentencia comienza con un "\$", termina con un salto de línea y no puede tener más de 80 caracteres de texto visible, además de que los datos están contenidos dentro de esta misma línea de elementos separados por comas, luego le siguen las letras GP para los datos recibidos de un receptor GPS, y a continuación tres letras para identificar el tipo de sentencia, de las cuales existen varias como la RMC, que es la que aporta la información más completa para el trabajo con flotas. Existe una disposición para una suma de comprobación al final de cada sentencia, que pudiera o no ser verificada por la unidad que lee los datos. La comprobación consiste en un OR lógico de todos los caracteres que se encuentran entre el '\$' y '\*'. (30)

```
$PTYCVER,TYCO Electronics - 101-01 (11:29:15 Jan 24 2005)<CR><LF>
$PSTMVER,GPS Version 4.20.3 release ARM (13:25:29 Dec 23 2004)<CR><LF>
$PSTMVER,Bootloader Version 2.00<CR><LF>
$GPGGA,131729.000,0211.3571,S,07953.4670,W,0.00,99.0,0020.5,M,12.4,M,,*52<CR><LF>
$GPRMC,131729.000,V,0211.3571,S,07953.4670,W,0.0,0.0,280707,0.0,W*68<CR><LF>
$GPGSA,A,1,,,,,,,,,99.0,99.0,99.0*00<CR><LF>
$GPGSV,3,1,12,01,13,261.00,05,36,118.00,06,30,178.00,07,32,190.00*73<CR><LF>
$GPGSV,3,2,12,09,25,032.00,12,23,100.00,14,29,298.00,18,27,011.00*76<CR><LF>
$GPGSV,3,3,12,21,85,026.00,24,53,137.00,30,37,141.00,31,23,216.00*70<CR><LF>
$GPGGA,131730.000,0211.3571,S,07953.4670,W,0.00,99.0,0020.5,M,12.4,M,,*5A<CR><LF>
$GPRMC,131730.000,V,0211.3571,S,07953.4670,W,0.0,0.0,280707,0.0,W*60<CR><LF>
$GPGSA,A,1,,,,,,,,,99.0,99.0,99.0*00<CR><LF>
$GPGSV,3,1,12,01,13,261.00,05,36,118.00,06,30,178.00,07,32,190.00*73<CR><LF>
$GPGSV,3,2,12,09,25,032.00,12,23,100.00,14,29,298.00,18,27,011.00*76<CR><LF>
$GPGSV,3,3,12,21,85,026.00,24,53,137.00,30,37,141.00,31,23,216.00*70<CR><LF>
$GPGGA,131731.000,0211.3571,S,07953.4670,W,0.00,99.0,0020.5,M,12.4,M,,*5B<CR><LF>
$GPRMC,131731.000,V,0211.3571,S,07953.4670,W,0.0,0.0,280707,0.0,W*61<CR><LF>
$GPGSA,A,1,,,,,,,,,99.0,99.0,99.0*00<CR><LF>
$GPGSV,3,1,12,01,13,261.00,05,36,118.00,06,30,178.00,07,32,190.00*73<CR><LF>
$GPGSV,3,2,12,09,25,032.00,12,23,100.00,14,29,298.00,18,27,011.00*76<CR><LF>
$GPGSV,3,3,12,21,85,026.00,24,53,137.00,30,37,141.00,31,23,216.00*70<CR><LF>
$GPGGA,131732.000,0211.3571,S,07953.4670,W,0.00,99.0,0020.5,M,12.4,M,,*58<CR><LF>
$GPRMC,131732.000,V,0211.3571,S,07953.4670,W,0.0,0.0,280707,0.0,W*62<CR><LF>
$GPGSA,A,1,,,,,,,,,99.0,99.0,99.0*00<CR><LF>
$GPGSV,3,1,12,01,13,261.00,05,36,118.00,06,30,178.00,07,32,190.00*73<CR><LF>
$GPGSV,3,2,12,09,25,032.00,12,23,100.00,14,29,298.00,18,27,011.00*76<CR><LF>
$GPGSV,3,3,12,21,85,026.00,24,53,137.00,30,37,141.00,31,23,216.00*70<CR><LF>
$GPGGA,131733.000,0211.3571,S,07953.4670,W,0.00,99.0,0020.5,M,12.4,M,,*59<CR><LF>
```

Fig. 4 Trama GPS en formato NMEA. (38)

### 2.9.1 Sentencia RMC del estándar NMEA.

Debido a que la sentencia RMC ofrece la información más completa para lograr el control de flotas será con la cual se trabajará en la presente investigación. A continuación se muestra una figura que describe detalladamente los campos que componen a la sentencia RMC. Entre los datos más relevantes de la misma se encuentran la hora, fecha, latitud, longitud, hemisferios y la velocidad.

## Capítulo 2: Características del sistema.

---

### Estructura de la sentencia:

```
$GPRMC, hhmss, estado, latitud, N, longitud, E, velocidad, rumbo, ddmmyy, mv, mvE, modo*suma_chequeo
```

### Ejemplo:

```
$GPRMC, 083559.00, A, 4717.11437, N, 00833.91522, E, 0.004, 77.52, 091202, , , A*57
```

Fig. 5 Estructura sentencia RMC. (31)

Una vez capturadas las tramas GPS se analizan una a una, para ello se almacenan en estructuras de datos cada campo por separado, luego se le aplican algoritmos matemáticos para su conversión y posteriormente se guardan en la Base de Datos.

### 2.10 Conclusiones parciales.

Luego de haberse realizado un estudio, teniendo en cuenta las características de las herramientas y tecnologías, las cuales se incluyen dentro del grupo de las soberanas, pudiendo afirmar que el uso de las mismas es de gran importancia ya que favorecen al desarrollo eficiente de la solución propuesta con un bajo costo, se decidió utilizar en el desarrollo del sistema para la captura y almacenamiento en tiempo real de tramas GPS, RUP como metodología de desarrollo de software, Visual Paradigm como herramienta CASE, UML como lenguaje de modelado, PostgreSQL como gestor de base de datos con la extensión PostGIS para el soporte de datos geográficos, como lenguaje de programación C++ utilizando como framework de desarrollo Qt Creator y la herramienta Data Generator para realizar las pruebas de rendimiento.

## *Capítulo 3: Presentación de la solución propuesta.*

---

### **Capítulo 3: Presentación de la solución propuesta.**

#### **3.1 Introducción.**

En el presente capítulo se realiza la descripción de la solución propuesta. Se describen los procesos asociados al negocio a partir de un modelo de dominio y se especifican detalladamente los requisitos funcionales y no funcionales que debe presentar el sistema para cumplir el objetivo planteado. Se definen los casos de usos del sistema y los actores que interactúan con los mismos, elaborándose una descripción de cada uno de ellos. Además se representa mediante diagramas los casos de usos del sistema. Todos los artefactos descritos en el presente capítulo están definidos según la fase de desarrollo orientada por RUP.

#### **3.2 Modelo de dominio.**

El modelo de dominio se emplea cuando los procesos no están bien definidos, cuando no se tiene un cliente o todos los elementos necesarios, como en este caso un receptor GPS o un especialista en este tipo de tecnologías que apoye a la investigación. Un modelo de dominio captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las cosas que existen o los eventos que suceden en el entorno en el que trabaja el sistema. El modelo de dominio se describe mediante diagramas UML (especialmente mediante diagramas de clases). Estos diagramas muestran a los clientes, usuarios, revisores y a otros desarrolladores las clases del dominio y cómo se relacionan unas con otras mediante asociaciones. (15) Este modelo provee a los usuarios, desarrolladores y clientes un vocabulario común para entender el contexto en el que se desarrolla el sistema en cuestión.

## Capítulo 3: Presentación de la solución propuesta.

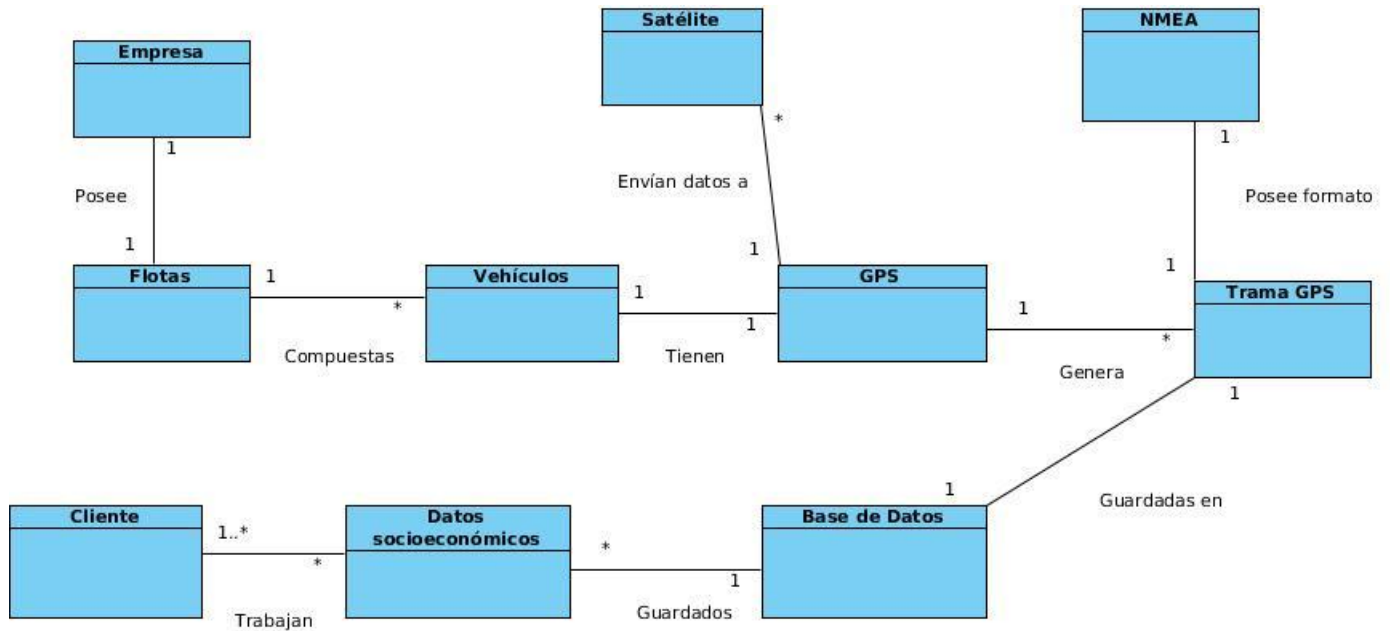


Fig. 6 Diagrama Modelo de Dominio.

El diagrama de la Fig.6 está basado en una empresa que contiene flotas, las cuales están compuestas por varios vehículos. Los mismos poseen GPS para conocer la posición exacta donde se encuentran ubicados, entre otros datos relevantes como la velocidad y el rumbo, esto se obtiene mediante una conexión entre los receptores GPS con los satélites. Estos dispositivos generan tramas GPS, que no son más que el conjunto de datos que almacenan la información referente a los vehículos. A estas tramas luego de ser capturadas, se le comprueba la integridad de sus datos y después se guardan en la Base de Datos. El formato de las mismas es NMEA, que es el estándar mundial de tramas GPS. Los clientes también pueden interactuar con los datos socioeconómicos almacenados en la Base de Datos.

### 3.3 Glosario de términos del dominio.

**Empresa:** Es la entidad donde se utilizará el sistema.

**Flotas:** Constituyen el conjunto de vehículos pertenecientes a la empresa.

**Vehículos:** Es el medio de locomoción que permite el traslado de un lugar a otro.

## *Capítulo 3: Presentación de la solución propuesta.*

---

**GPS:** Las siglas GPS se corresponden con Global Positioning System que significa Sistema de Posicionamiento Global. Es el dispositivo que permitirá conocer la posición de un objeto móvil, a todo momento, gracias a la recepción de señales emitidas por una red de satélites.

**Satélite:** Son objetos puestos en órbita mediante la intervención humana. Dispositivos que trazan órbitas alrededor de nuestro planeta o de otro, y que tienen como objetivo trasladar equipamientos que permiten recoger y retransmitir información.

**Trama GPS:** Son el conjunto de datos recogidos por el GPS, las cuales almacenan información referente a las flotas como la velocidad, trayectoria, latitud, velocidad, entre otros.

**NMEA:** Conocido mundialmente como National Marine Electronics Association (NMEA), es el protocolo estándar mundial usado en dispositivos GPS, que permite la organización y almacenamiento de los datos en una trama GPS, es decir, es el formato que define la estructura de la información cuando se recibe por el receptor GPS.

**Base de Datos:** Sistema computarizado encargado de almacenar, gestionar y mantener disponible la información de la empresa para su posterior uso.

**Cliente:** Es el usuario que mantiene una interacción directa con la entidad, es decir que utilizará los servicios brindados por la misma.

**Datos socioeconómicos:** Son los datos asociados a los vehículos que controla el sistema.

### **3.4 Especificación de Requisitos.**

El propósito fundamental de la definición de requisitos es guiar el desarrollo por el camino correcto. Esto se consigue mediante la descripción de los requisitos del sistema. Los requisitos se dividen en dos grupos: los requisitos funcionales, que son capacidades o condiciones que el sistema debe cumplir, y los requisitos no funcionales que son las condiciones o cualidades que el producto debe tener. A continuación se muestra el listado de los requisitos funcionales y los requisitos no funcionales propuestos para el desarrollo del sistema.

## *Capítulo 3: Presentación de la solución propuesta.*

---

### **3.4.1 Requisitos Funcionales (RF).**

**RF 1.** Verificar formato de trama GPS.

**RF 2.** Validar datos de trama GPS.

**RF 3.** Insertar datos espaciales asociados a las tramas GPS en la Base de Datos.

### **3.4.2 Requisitos No Funcionales (RNF).**

#### **Confiabilidad**

**RNF 1.** Los datos insertados en la Base de Datos provenientes de las tramas GPS deben de ser válidos.

#### **Hardware**

**RNF 2.** El servidor de BD debe tener 2GB o más de memoria RAM y 80GB ó más de capacidad de disco duro.

**RNF 3.** El sistema requiere un microprocesador igual o superior al Pentium Dual Core, con una velocidad de procesamiento de 2.00 GHz o más.

#### **Software**

**RNF 4.** Se utilizará PostgreSQL como Sistema Gestor de Base de Datos.

**RNF 5.** Se utilizará PostGIS como extensión de PostgreSQL para el soporte de datos espaciales.

**RNF 6.** Se utilizará sistema operativo GNU/Linux ó Windows.

#### **Requisitos Legales, de Derechos de Autor y otros**

**RNF 7.** Las herramientas de desarrollo son todas de software libres.

**RNF 8.** Como producto, el sistema para la captura y almacenamiento en tiempo real de tramas GPS se distribuirá bajo las normativas legales establecidas en el registro comercial emitidas por las entidades jurídicas de la Universidad de las Ciencias Informáticas.

## Capítulo 3: Presentación de la solución propuesta.

---

### Eficiencia

**RNF 9.** El tiempo de lectura de las tramas GPS estará dado por la cantidad de tramas que se reciban, entre mayor cantidad de datos, mayor será el tiempo de procesamiento de los mismos.

### Disponibilidad

**RNF 10.** La lectura y manejo de las tramas GPS será garantizada en todo momento, mientras no exista un fallo con el generador de tramas GPS.

### 3.5 Descripción del sistema.

#### 3.5.1 Descripción de los actores del sistema.

Actor	Descripción
Sistema	Es el actor que realiza las acciones de captura, validación y almacenamiento de las tramas GPS en la Base de Datos.

Tabla 1. Descripción de los actores del sistema.

#### 3.5.2 Diagrama de Casos de Uso del Sistema.

Un diagrama de casos de uso del sistema es un modelo del sistema que contiene actores, casos de usos y las relaciones entre ellos. Además describe lo que hace el sistema para cada tipo de usuario, cada uno de estos representados mediante uno o más actores.

#### Listado de Casos de Usos de Sistema

- ✓ **CUS 1:** Verificar formato de trama GPS (RF1).
- ✓ **CUS 2:** Validar datos de tramas GPS (RF2).
- ✓ **CUS 3:** Insertar datos de tramas GPS (RF3).

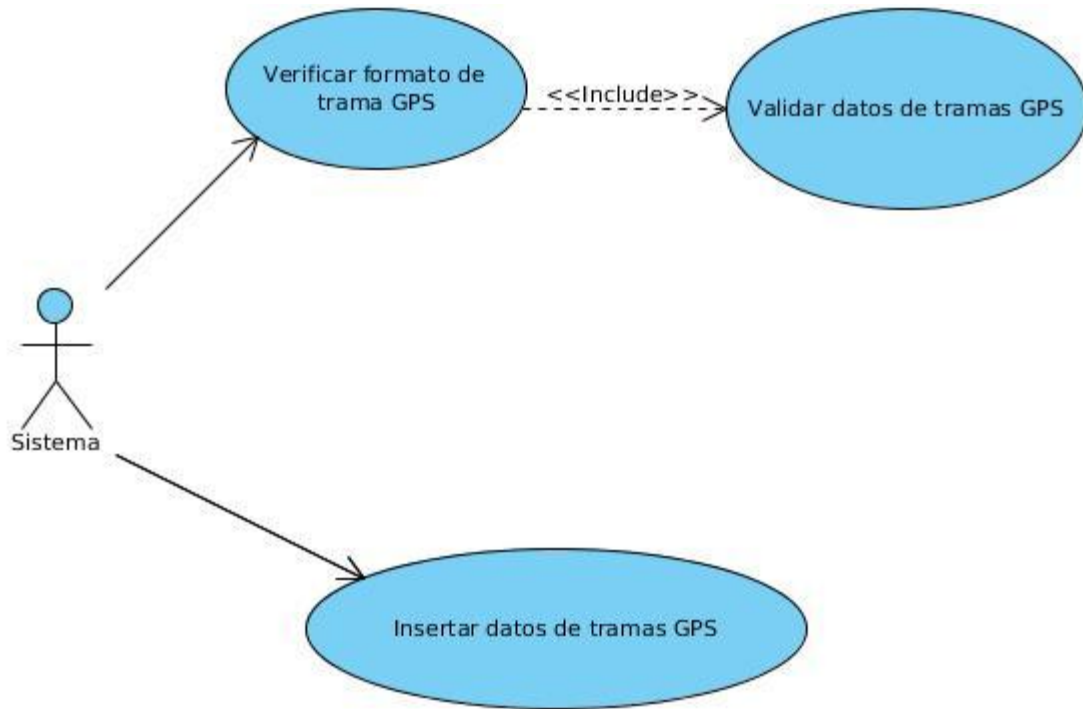


Fig. 7 Diagrama de Casos de Uso del Sistema.

### 3.5.3 Descripción de los Casos de Uso del Sistema.

Un caso de uso especifica una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable de valor para un actor concreto. (15)

#### ➤ CUS 1. Verificar formato de trama GPS.

<b>Objetivo</b>	Verificar que el formato de la trama GPS es NMEA.
<b>Actores</b>	Sistema
<b>Resumen</b>	El caso de uso comienza una vez que se captura la trama GPS. Luego se verifica el formato de la misma y si todo está correctamente verificado u ocurre algún error, termina el caso de uso.
<b>Referencia</b>	RF1
<b>Complejidad</b>	Media



## *Capítulo 3: Presentación de la solución propuesta.*

---

<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	El dispositivo GPS ha enviado la trama.	
<b>Postcondiciones</b>	La trama GPS está en formato NMEA y no le faltan datos.	
<b>Flujo normal de eventos</b>		
<b>Flujo básico Verificar formato de trama GPS</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
1		Captura la trama proveniente del GPS y la almacena en una estructura de datos.
2		Verifica que la trama GPS tenga formato NMEA comenzando por el carácter \$ todas sus sentencias.
3		Verifica que a la trama NMEA no le falta la sentencia RMC.
<b>Flujos alternos</b>		
<b>Evento 1. La trama no ha sido recibida.</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
1		Termina el caso de uso.
<b>Evento 2. La trama GPS no tiene formato NMEA.</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
1		Cancela la verificación de los datos.
2		Termina el caso de uso.
<b>Evento 3. La trama GPS no tiene sentencia RMC.</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
1		Cancela la verificación de los datos.

## Capítulo 3: Presentación de la solución propuesta.

2		Termina el caso de uso.
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	
<b>Requisitos no funcionales</b>		
<b>Entradas</b>		
<b>Salidas</b>		

Tabla 2. Descripción textual del CUS Verificar formato de trama GPS.

### ➤ CUS 2. Validar datos de trama GPS.

<b>Objetivo</b>	Verificar que todos los datos de la trama GPS sean correctos.	
<b>Actores</b>	Sistema	
<b>Resumen</b>	El caso de uso comienza una vez que se ha validado el formato de la trama GPS. Luego se verifica la integridad de los datos de la misma y si todo está correctamente verificado u ocurre algún error, termina el caso de uso.	
<b>Referencia</b>	RF2	
<b>Complejidad</b>	Media	
<b>Prioridad</b>	Crítico	
<b>Precondiciones</b>	La trama GPS se encuentra en formato NMEA.	
<b>Postcondiciones</b>	Los datos de la trama GPS son correctos.	
<b>Flujo normal de eventos</b>		
<b>Flujo básico Validar datos de trama GPS</b>		
	<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1		Accede a la sentencia RMC de la trama NMEA.

## *Capítulo 3: Presentación de la solución propuesta.*

2		Verifica que a la sentencia RMC no le falten campos.
3		Verifica que no existan errores de cálculo de posición en la sentencia RMC.
<b>Flujos alternos</b>		
<b>Evento 1. Le faltan campos a la sentencia RMC.</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
1		Cancela la validación de la trama GPS.
2		Termina el caso de uso.
<b>Evento 2. Existen errores de cálculo de posición en la sentencia RMC.</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
1		Cancela la validación de la trama GPS.
2		Termina el caso de uso.
<b>Relaciones</b>	<b>CU Incluidos</b>	Verificar formato de trama GPS
	<b>CU Extendidos</b>	
<b>Requisitos no funcionales</b>		
<b>Entradas</b>		
<b>Salidas</b>		

**Tabla 3. Descripción textual del CUS Validar datos de trama GPS.**

La descripción del CUS 3. Insertar datos de tramas GPS se puede encontrar en los Anexos de este documento.

### **3.6 Conclusiones parciales.**

Con el desarrollo de este capítulo se ha obtenido como resultado el modelo de dominio, en el cual se presentan los conceptos que intervienen en los procesos que se realizan. Se identificaron los requisitos funcionales y no funcionales, los cuales servirán como base para construir un sistema que cumpla con todas las restricciones y funcionalidades especificadas, garantizando que el mismo tenga una mayor

## *Capítulo 3: Presentación de la solución propuesta.*

---

calidad. Se ha elaborado el diagrama de casos de uso del sistema, el cual presenta el actor y su relación con las funcionalidades que va a desempeñar en el sistema. Además fueron elaboradas las descripciones textuales de los casos de uso para lograr una mayor comprensión del funcionamiento del sistema.

# *Capítulo 4: Construcción de la solución propuesta.*

---

## **Capítulo 4: Construcción y validación de la solución propuesta.**

### **4.1 Introducción.**

Este capítulo tiene como objetivo la construcción y validación de la solución propuesta anteriormente. Para ello se presentan una serie de artefactos del diseño de la solución propuesta como son el diagrama de clases del diseño, el modelo de datos, el modelo de despliegue y el modelo de implementación. También se describen los patrones de diseño utilizados en el desarrollo del sistema. Además se realizarán las pruebas adecuadas para validar los requisitos funcionales descritos anteriormente, con el objetivo de garantizar el correcto funcionamiento del sistema. Las pruebas de Caja Blanca y las pruebas de volumen son las que se le efectuarán al sistema.

### **4.2 Arquitectura de software.**

Con el término arquitectura designamos la estructura de alto nivel de los subsistemas y de los componentes, así como de sus interfaces, conexiones e interacciones. La arquitectura se compone de un conjunto de esquemas, subsistemas, clases, asignaciones de responsabilidades y colaboraciones entre objetos que cumplen con las funciones del sistema. (33)

Un sistema software requiere una arquitectura para que los desarrolladores puedan progresar hasta tener una visión común. Se necesita una arquitectura para comprender el sistema, organizar el desarrollo, fomentar la reutilización y hacer evolucionar el sistema. (15)

#### **4.2.1 El modelo Cliente-Servidor.**

En el modelo cliente servidor, el cliente envía un mensaje solicitando un determinado servicio a un servidor (hace una petición), y este envía uno o varios mensajes con la respuesta (provee el servicio). En un sistema distribuido cada máquina puede cumplir el rol de servidor para algunas tareas y el rol de cliente para otras. Usualmente la mayoría del trabajo pesado se hace en el proceso llamado servidor y los procesos clientes sólo se ocupan de la interacción con el usuario.

El cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor, se le conoce con el término front-end. Normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten

## Capítulo 4: Construcción de la solución propuesta.

---

construir interfaces gráficas de usuario (GUI), además de acceder a los servicios distribuidos en cualquier parte de una red. Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- ✓ Interactuar con el usuario.
- ✓ Procesar la lógica de la aplicación y hacer validaciones locales.
- ✓ Generar requerimientos de bases de datos.
- ✓ Recibir resultados del servidor.

El servidor es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se le conoce con el término back-end. El mismo normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos. Las funciones que lleva a cabo el proceso servidor se resumen en los siguientes puntos:

- ✓ Aceptar los requerimientos de bases de datos que hacen los clientes.
- ✓ Procesar requerimientos de bases de datos.
- ✓ Formatear datos para transmitirlos a los clientes.
- ✓ Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

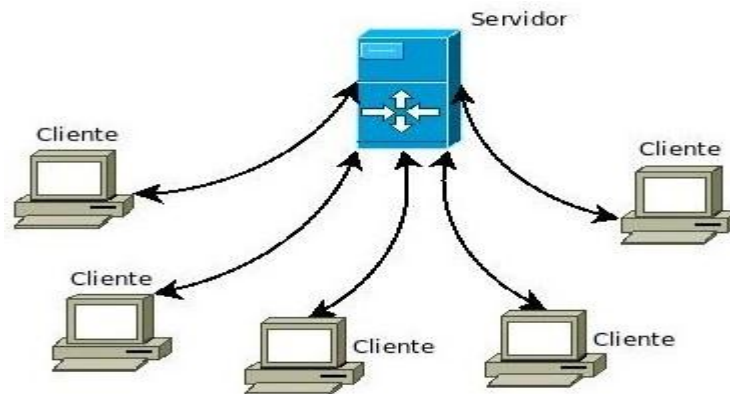


Fig. 8 Imagen que representa el modelo Cliente-Servidor. (39)

## *Capítulo 4: Construcción de la solución propuesta.*

---

### **Ventajas del modelo Cliente-Servidor:**

- ✓ Posibilidad de utilizar máquinas considerablemente más baratas que las requeridas por una solución centralizada, basada en sistemas grandes. Además, se pueden utilizar componentes, tanto de hardware como de software, de varios fabricantes, lo cual contribuye considerablemente a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.
- ✓ En el uso de interfaces gráficas para el usuario, el esquema Cliente/Servidor presenta la ventaja con respecto a uno centralizado, de que no es siempre necesario transmitir información gráfica por la red pues esta puede residir en el cliente, lo cual permite aprovechar mejor el ancho de banda de la red.
- ✓ La estructura inherentemente modular facilita además la integración de nuevas tecnologías y el crecimiento de la infraestructura computacional, favoreciendo así la escalabilidad de las soluciones.
- ✓ El esquema Cliente/Servidor contribuye además, a proporcionar, a los diferentes departamentos de una organización, soluciones locales, pero permitiendo la integración de la información relevante a nivel global.
- ✓ La plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma, posibilitando conectar clientes y servidores independientemente de sus plataformas. (34)

La implementación del sistema se realizará bajo una arquitectura cliente-servidor debido a que se trata de una comunicación por socket mediante el protocolo TCP<sup>4</sup>, donde muchos dispositivos móviles se encontrarán constantemente enviando información, la cual será capturada por el sistema que se propone. Los clientes estarán representados por cada dispositivo móvil con su correspondiente GPS y el sistema propuesto actuará como servidor, al ser el mismo quien reciba toda la información asociada a las tramas GPS proveniente de los distintos dispositivos móviles.

### **4.2.2 Patrón arquitectónico en capas.**

---

<sup>4</sup> TCP: Protocolo de Control de Transmisión usado para la transmisión de datos.

## *Capítulo 4: Construcción de la solución propuesta.*

---

Para la construcción del sistema propuesto se decidió utilizar el patrón arquitectónico en capas. Este patrón descompone una aplicación en un conjunto de capas independientes y ordenadas jerárquicamente. Cada nivel o capa usa los servicios de la capa inmediatamente inferior y ofrece servicios a la inmediatamente superior.

Para ello se definió la utilización de dos capas, que tienen como objetivo fundamental separar la lógica del negocio de la capa de acceso a datos. Dentro del modelo en capas se utilizan solo dos, porque en el sistema no es necesario mostrar información a ningún usuario. Entre sus principales ventajas se puede decir que admite optimizaciones y proporciona una amplia reutilización. A continuación se muestran las capas:

- Capa lógica del negocio: Es donde se procesa la información y se comunica con la capa de acceso a datos para almacenar la información proveniente de los GPS.
- Capa de acceso a datos: Es la encargada de almacenar los datos asociados a las tramas GPS.

### **4.3 Modelo de diseño.**

El modelo de diseño es el modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación tienen un impacto en el sistema a considerar. Además, el modelo de diseño sirve de abstracción a la implementación del sistema. (15)

Es válido aclarar que en la solución se implementó un generador de tramas GPS, ya que no se cuenta con un dispositivo GPS para la obtención de las mismas, esto constituye un valor agregado con lo que va a contar el sistema, que sirve para comprobar el funcionamiento de la aplicación.

#### **4.3.1 Diagramas de clase del diseño.**



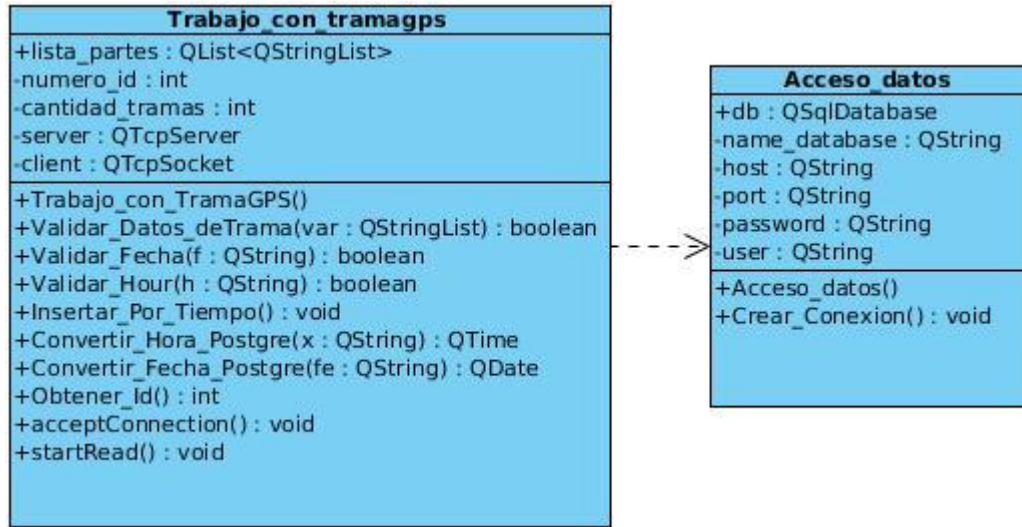


Fig. 9 Diagrama de Clases del Diseño.

### 4.3.2 Patrones de diseño.

Un patrón es una pareja de problema/solución con un nombre y que es aplicable a otros contextos, con una sugerencia sobre la manera de usarlos en situaciones nuevas.

#### 4.3.2.1 Patrones GRASP.

GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns (patrones generales de software para asignar responsabilidades). El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente el software orientado a objetos.

**Patrón Experto:** Se basa en asignar una responsabilidad a la clase que cuenta con la información necesaria para cumplir dicha responsabilidad. Este patrón permite conservar el encapsulamiento, ya que los objetos se valen de su propia información para realizar lo que se le oriente.

**Patrón Creador:** El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El objetivo fundamental de este patrón es encontrar un creador que debemos conectar con el objeto producido en cualquier evento. Al escogerlo como creador, se da soporte al bajo acoplamiento.

## *Capítulo 4: Construcción de la solución propuesta.*

---

**Patrón Bajo Acoplamiento:** Este patrón asigna una responsabilidad para mantener el bajo acoplamiento. La idea es tratar de que una clase no dependa de muchas otras, así esa clase no tendrá muchas dependencias, lo que facilitará la reutilización de la misma y se reducirá el impacto de los cambios.

**Patrón Alta Cohesión:** Una clase con alta cohesión tienen un número relativamente pequeño de métodos, con funcionalidades altamente relacionadas, y no realiza mucho trabajo, colaborando con otros objetos para compartir el esfuerzo si la tarea es extensa. Las clases con alta cohesión son relativamente fáciles de mantener, entender y reutilizar.

**Patrón Controlador:** Para manejar y controlar los eventos del sistema. Con la utilización de este patrón se logra separar la lógica de negocio de la capa de presentación. De esta manera se logra un mayor control sobre el sistema y se favorece la reutilización de código. (33)

### **4.3.2.2 Patrones GOF.**

Los patrones GOF (Gang Of Four o Pandilla de los Cuatro) son patrones de diseño orientado a objetos y deben su nombre a los creadores de los mismos.

**Patrón Observador:** El patrón observador define una dependencia del tipo “uno a muchos” entre objetos, se definen objetos observadores y un objeto observado, con el fin de detectar algún cambio en el objeto observado y reproducirlo en los observadores (33). Se evidencia el uso de este patrón en la aplicación cuando son utilizados los SIGNAL y SLOT, que son los mecanismos de manejos de eventos en el framework QT.

### **4.4 Diseño de la Base de Datos.**

El diseño de una base de datos consiste en definir la estructura de los datos que debe tener la base de datos de un sistema de información determinado. En el caso relacional, esta estructura será un conjunto de esquemas de relación con sus atributos, dominios de atributos, claves primarias, claves foráneas, entre otros. El diseño de base de datos se descompone en tres partes: diseño conceptual, diseño lógico y diseño físico.

#### **4.4.1 Diagrama Entidad-Relación.**

## Capítulo 4: Construcción de la solución propuesta.

Modelo de datos basado en una percepción del mundo real que esta compuesto por un conjunto de objetos básicos llamados entidades, atributos que son propiedades o características de las entidades y las relaciones entre estos objetos.

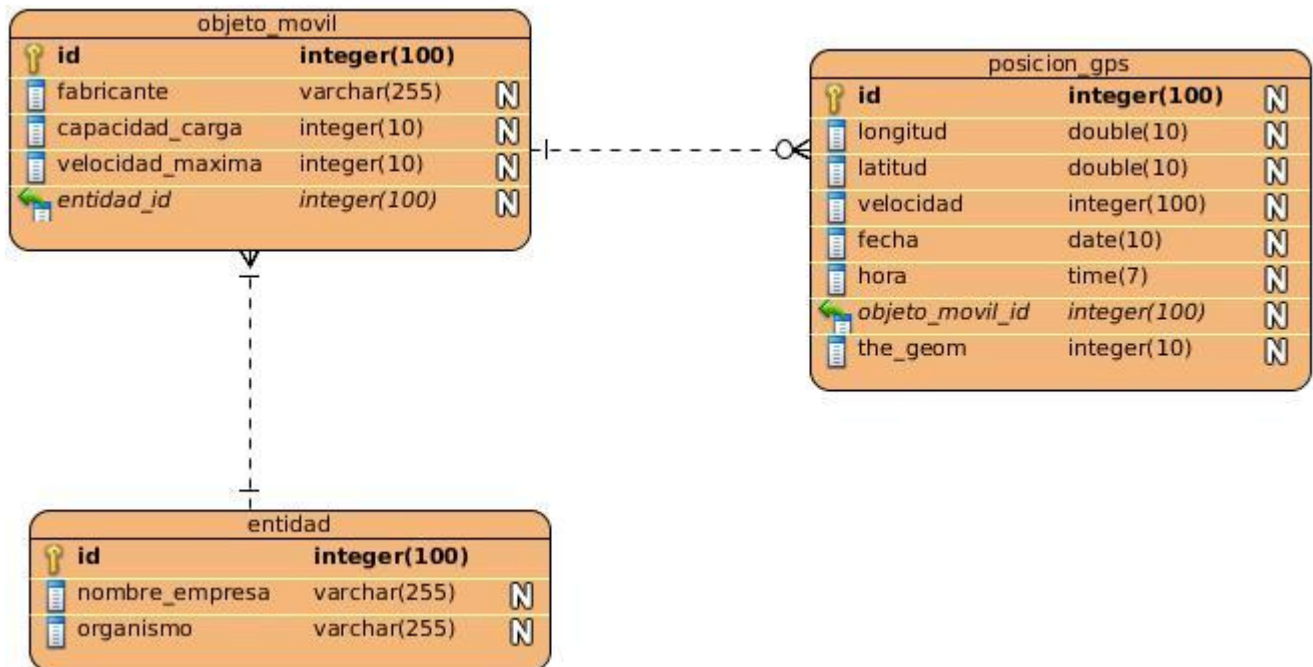


Fig. 10 Diagrama Entidad-Relación.

El diagrama Entidad-Relación mostrado anteriormente consta de tres tablas las cuales se describen a continuación:

- **Tabla “objeto\_movil”:** Se almacena el número que identifica al vehículo dentro de la flota, el nombre del fabricante del mismo, la capacidad de carga que posee y la velocidad máxima a la que puede arribar en su recorrido, además presenta un atributo que identifica a la entidad a la que pertenece dicho auto.
- **Tabla “entidad”:** Se almacena en la misma el número que identifica a la entidad a la cual se le está supervisando la flota, el nombre de la empresa y el organismo al cual pertenece.

## Capítulo 4: Construcción de la solución propuesta.

- **Tabla “posicion\_gps”:** Se almacena el número que identifica a la posición, la longitud y la latitud donde se encontraba el auto en ese instante de tiempo, la velocidad con la cual transitaba, la fecha y la hora en que se capturó la posición, además de un número que identifica a que vehículo pertenecen estos datos y un campo geométrico de tipo punto que contiene el par longitud-latitud.

### 4.5 Modelo de despliegue.

Con el Modelo de Despliegue se representará el sistema a desarrollar desde un punto de vista físico, visualizando los recursos de ejecución que requiere la aplicación (computador, memoria, dispositivo, capacidad de procesamiento).

#### 4.5.1 Diagrama de despliegue.

El diagrama de despliegue muestra las relaciones físicas entre los componentes hardware y software en el sistema propuesto. Se representa como un conjunto de nodos unidos por conexiones de comunicación.

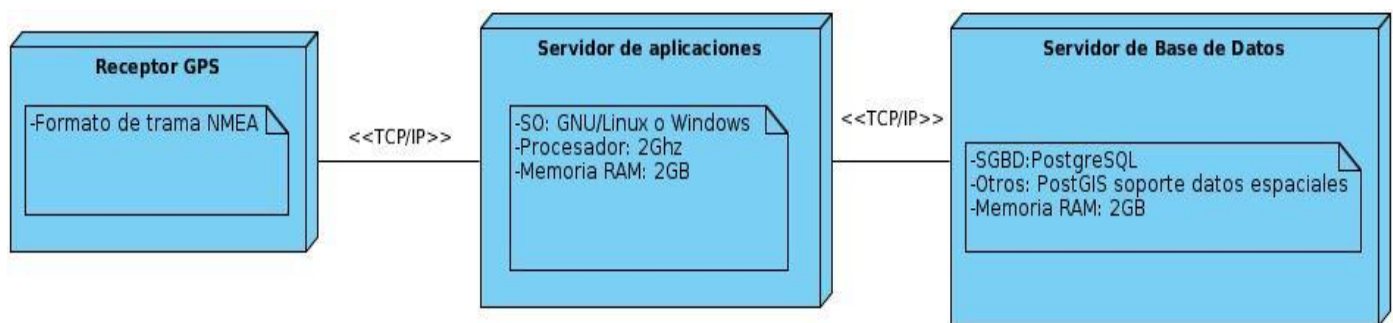


Fig. 11 Diagrama de despliegue.

### 4.6 Modelo de implementación.

El modelo de implementación describe cómo los elementos del modelo de diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables, entre otros. Además describe como se organizan los componentes de acuerdo con los mecanismos de estructuración y popularización disponibles en el entorno de implementación y en el lenguaje o los lenguajes de programación utilizados, y como dependen los componentes unos de otros. (15)

#### 4.6.1 Diagrama de componentes.

## Capítulo 4: Construcción de la solución propuesta.

---

Un diagrama de componentes muestra las dependencias lógicas entre componentes software, sean éstos componentes fuentes, binarios o ejecutables.

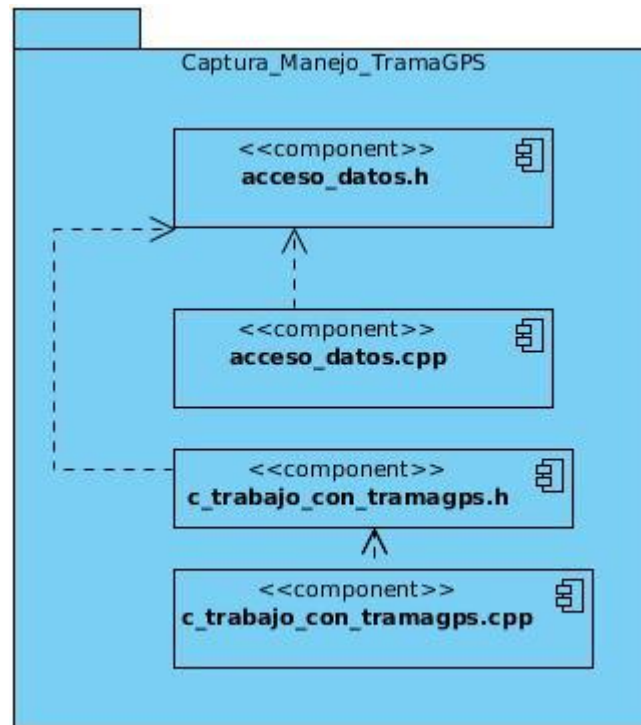


Fig. 12 Diagrama de Componentes.

### 4.7 Pruebas de sistema.

Las pruebas son un conjunto de actividades en las cuales un sistema es ejecutado bajo condiciones específicas con el objetivo de descubrir errores. Es un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones del diseño y de la codificación. Se ha seleccionado para la aplicación de pruebas al sistema las prueba de caja blanca, debido a que el mismo no cuenta con interfaz de usuario, además de las pruebas de volumen.

#### 4.7.1 Pruebas de Caja Blanca.

La prueba de caja blanca, es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo; que se ejerciten todas las decisiones

## Capítulo 4: Construcción de la solución propuesta.

---

lógicas en sus vertientes verdadera y falsa; se ejecuten todos los bucles en sus límites y se ejerciten las estructuras internas de datos para asegurar su validez. (36)

### 4.7.1.1 Técnicas de pruebas de Caja Blanca.

La prueba del camino básico es una técnica de prueba de caja blanca propuesta inicialmente por Tom McCabe. El método del camino básico permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa. (36)

Los pasos del diseño de pruebas a seguir mediante el camino básico son:

1. A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.
2. Se calcula la complejidad ciclomática del grafo de flujo.
3. Se determina el conjunto básico de caminos independientes.
4. Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

### 4.7.1.2 Pruebas al CUS Validar datos de trama GPS

```
bool Trabajo_con_TramaGPS::Validar_Datos_deTrama(QStringList var)
{
    bool flag=false;      1

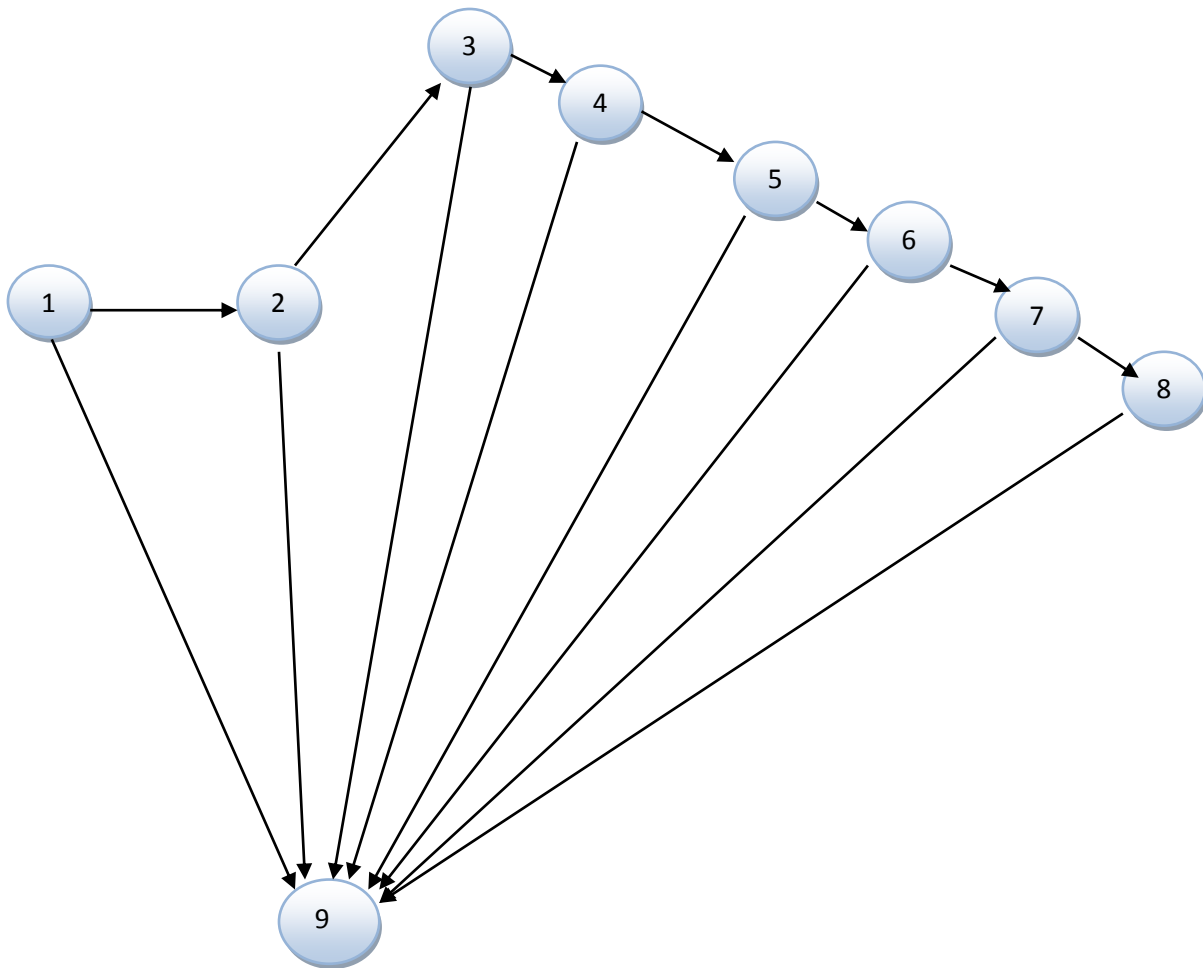
    if(Validar_Hour(var.at(1)))      1
    {
        if(var.at(2)=="A")      2
        {
            if((var.at(3).toDouble()<=9999.99) && (var.at(5).toDouble() <=9999.99))      3
            {
                if((var.at(4)=="N")|| (var.at(4)=="S"))      4
                {
                    if((var.at(6)=="E")|| (var.at(6)=="W"))      5
                    {
```

## Capítulo 4: Construcción de la solución propuesta.

---

```
    if(var.at(7).toInt()>=0)    6
    {
        if(Validar_Fecha(var.at(9)))    7
        {
            flag=true;    8
        }
    }
}
}
}
}
return flag;    9
}
```

**Construcción del grafo de flujo asociado al CUS Validar datos de trama GPS**



**Complejidad ciclomática del grafo de flujo asociado al CUS2.**

$$V(G) = \text{Número de Aristas} - \text{Número de Nodos} + 2$$

$$V(G) = 15 - 9 + 2$$

$$V(G) = 8$$

**Conjunto básico de caminos independientes.**

**Camino 1:** 1-9



## Capítulo 4: Construcción de la solución propuesta.

---

**Camino 2:** 1-2-9

**Camino 3:** 1-2-3-9

**Camino 4:** 1-2-3-4-9

**Camino 5:** 1-2-3-4-5-9

**Camino 6:** 1-2-3-4-5-6-9

**Camino 7:** 1-2-3-4-5-6-7-9

**Camino 8:** 1-2-3-4-5-6-7-8-9

**Conjunto de pruebas.**

<b>Prueba Camino 1</b>	Se muestran datos relacionados con la prueba.
<b>Flag inicial</b>	False
<b>Hora</b>	106402
<b>Salida</b>	False
<b>Resultados esperados</b>	El método ha recorrido el camino adecuado debido a que el formato de la hora no es correcto.

<b>Prueba Camino 2</b>	Se muestran datos relacionados con la prueba.
<b>Flag inicial</b>	False
<b>Hora</b>	100402
<b>Estado</b>	B
<b>Salida</b>	False

## Capítulo 4: Construcción de la solución propuesta.

---

<b>Resultados esperados</b>	El método ha recorrido el camino adecuado debido a que el estado de la trama GPS no es correcto.
-----------------------------	--

<b>Prueba Camino 3</b>	Se muestran datos relacionados con la prueba.
<b>Flag inicial</b>	False
<b>Hora</b>	100402
<b>Estado</b>	A
<b>Latitud/Longitud</b>	10000.03
<b>Salida</b>	False
<b>Resultados esperados</b>	El método ha recorrido el camino adecuado debido a que los valores de la latitud o la longitud no son válidos.

<b>Prueba Camino 4</b>	Se muestran datos relacionados con la prueba.
<b>Flag inicial</b>	False
<b>Hora</b>	100402
<b>Estado</b>	A
<b>Latitud/Longitud</b>	1235.65
<b>Indicador de Hemisferio(N/S)</b>	C
<b>Salida</b>	False
<b>Resultados esperados</b>	El método ha recorrido el camino adecuado debido a que el indicador de hemisferio no es válido.

## Capítulo 4: Construcción de la solución propuesta.

---

<b>Prueba Camino 5</b>	Se muestran datos relacionados con la prueba.
<b>Flag inicial</b>	False
<b>Hora</b>	100402
<b>Estado</b>	A
<b>Latitud/Longitud</b>	1235.65
<b>Indicador de Hemisferio(N/S)</b>	N
<b>Indicador de Hemisferio(E/W)</b>	Z
<b>Salida</b>	False
<b>Resultados esperados</b>	El método ha recorrido el camino adecuado debido a que el indicador de hemisferio no es válido.

<b>Prueba Camino 6</b>	Se muestran datos relacionados con la prueba.
<b>Flag inicial</b>	False
<b>Hora</b>	100402
<b>Estado</b>	A
<b>Latitud/Longitud</b>	1235.65
<b>Indicador de Hemisferio(N/S)</b>	N
<b>Indicador de Hemisferio(E/W)</b>	E
<b>Velocidad</b>	-23

## Capítulo 4: Construcción de la solución propuesta.

---

<b>Salida</b>	False
<b>Resultados esperados</b>	El método ha recorrido el camino adecuado debido a que el valor de la velocidad es incorrecto.

<b>Prueba Camino 7</b>	Se muestran datos relacionados con la prueba.
<b>Flag inicial</b>	False
<b>Hora</b>	100402
<b>Estado</b>	A
<b>Latitud/Longitud</b>	1235.65
<b>Indicador de Hemisferio(N/S)</b>	N
<b>Indicador de Hemisferio(E/W)</b>	E
<b>Velocidad</b>	80
<b>Fecha</b>	0405999
<b>Salida</b>	False
<b>Resultados esperados</b>	El método ha recorrido el camino adecuado debido a que el valor de la fecha es incorrecto.

<b>Prueba Camino 8</b>	Se muestran datos relacionados con la prueba.
<b>Flag inicial</b>	False
<b>Hora</b>	100402

## Capítulo 4: Construcción de la solución propuesta.

---

<b>Estado</b>	A
<b>Latitud/Longitud</b>	1235.65
<b>Indicador de Hemisferio(N/S)</b>	N
<b>Indicador de Hemisferio(E/W)</b>	E
<b>Velocidad</b>	80
<b>Fecha</b>	040513
<b>Flag</b>	True
<b>Salida</b>	True
<b>Resultados esperados</b>	El método ha recorrido el camino adecuado debido a que todos los datos son correctos.

El resto de las pruebas se puede encontrar en los Anexos de este documento.

### 4.7.2 Pruebas de rendimiento.

En la Ingeniería del Software, las pruebas de rendimiento son aquellas que son realizadas para determinar que tan rápido un sistema realiza una tarea bajo ciertas condiciones pre-planificadas de trabajo. Estas pruebas también son utilizadas para validar y verificar diferentes aspectos de la calidad de software, como por ejemplo, escalabilidad, fiabilidad y el buen uso de los recursos. Las pruebas de rendimiento constituyen un subconjunto de la Ingeniería de Pruebas, la cual se esfuerza en mejorar el rendimiento, basándose en el diseño y la arquitectura de un sistema, antes de la realización del proceso de codificación.

Existen diferentes tipos de pruebas de rendimiento, las cuales son muy útiles, siempre y cuando sean usadas a través de herramientas que permitan visualizar los detalles técnicos y gráficos altamente informativos de los resultados de las pruebas. Entre los tipos de pruebas de rendimiento más importantes se encuentran las pruebas de cargas y las pruebas de volumen. (42)

## *Capítulo 4: Construcción de la solución propuesta.*

---

### **4.7.2.1 Pruebas de volumen.**

Las pruebas de volumen son pruebas típicas de entornos que utilicen bases de datos. Las mismas se realizan para analizar el comportamiento de la base de datos con volúmenes de datos almacenados para verificar si alcanza un límite de almacenamiento que pueda causar fallas.

Para el desarrollo de la prueba de volumen la base de datos se pobló con datos introducidos utilizando la herramienta Data Generator para PostgreSQL. En esta prueba se pobló específicamente la tabla posición\_gps, con 10 000 y 20 000 tuplas, ya que la misma es la que más información contendrá debido a que es donde se almacenará la información referente a las tramas GPS.

Luego de haberse aplicado la prueba, la Base de Datos no presentó problemas de límite de capacidad, desbordamientos de columnas, atributos o tipos de datos. Esto garantiza que el gestor utilizado para el desarrollo de la misma soporta el cúmulo de información que será insertada cuando la base de datos esté en explotación.

### **4.8 Conclusiones parciales.**

Como resultado de este capítulo se describió la construcción de la solución propuesta, mostrándose la arquitectura seleccionada para la realización de la aplicación. Se desarrolló el modelo de diseño que sirvió de base para la implementación, el modelo de datos que garantizó el funcionamiento adecuado de los datos del sistema, el diagrama de despliegue donde quedo definido el entorno donde se va a desplegar la aplicación, y el diagrama de componentes, donde se ofrece una visión general de cómo queda finalmente implementada la aplicación mostrándose la misma en términos de componentes. Además se culmina con la fase de pruebas para determinar la eficiencia de la aplicación, en este caso se llevaron a cabo pruebas de Caja Blanca y pruebas de volumen a la Base de Datos, mostrando ambos resultados satisfactorios.

### **Conclusiones**

Luego de concluido todo el proceso de construcción del Sistema para la captura y almacenamiento en tiempo real de tramas GPS para el centro GEYSED, como objetivo fundamental de esta investigación, se arriba a la conclusión de que el sistema propuesto garantiza el control del acceso a la Base de Datos en el momento de almacenar la información contenida en las tramas GPS. La implementación fue realizada en su totalidad por herramientas libres, lo que posibilitará la eliminación de las barreras presupuestales. La aplicación fue guiada por una metodología de desarrollo de software, lo que permitió obtener una detallada documentación que sirve como guía y como base para futuras mejoras. Una vez que esta propuesta sea integrada a cualquier sistema control de flotas se garantizará la toma de decisiones de manera instantánea. Por todo lo anterior planteado se da cumplimiento a las tareas acordadas previamente, dando respuesta al objetivo general de la investigación.

## **Recomendaciones**

Una vez cumplidos los objetivos del presente trabajo y en correspondencia con los resultados obtenidos se recomienda:

- Implementar una comunicación con receptores GPS donde se empleen otras sentencias dentro del formato NMEA.
- Integrar el Sistema para la captura y almacenamiento en tiempo real de tramas GPS para el centro GEYSED a cualquier sistema control de flotas donde se utilice tecnología GPS.



## Bibliografía

1. Laboratorio Unidad Pacífico Sur CIESAS. SISTEMAS DE INFORMACIÓN GEOGRÁFICA. [En línea] [Citado el: 16 de Octubre de 2012.] <http://langleruben.wordpress.com/%C2%BFque-es-un-sig/>.
2. **EcuRed, Colectivo de.** Enciclopedia Cubana en la Red. [En línea] [Citado el: 16 de Octubre de 2012.] <http://www.ecured.cu/index.php/GPS>.
3. **Ecured, Colectivo de.** Enciclopedia Cubana en la Red. [En línea] [Citado el: 16 de Octubre de 2012.] [http://www.ecured.cu/index.php/Sistema\\_de\\_Informaci%C3%B3n\\_Geogr%C3%A1fica](http://www.ecured.cu/index.php/Sistema_de_Informaci%C3%B3n_Geogr%C3%A1fica).
4. **Ecured, Colectivo de.** Enciclopedia Cubana en la Red. [En línea] [Citado el: 16 de Octubre de 2012.] [http://www.ecured.cu/index.php/Bases\\_de\\_datos](http://www.ecured.cu/index.php/Bases_de_datos).
5. **García, Rosa María Mato.** *Diseño de BASES DE DATOS.* 1999.
6. **Olaya, Víctor.** *Sistemas de Información Geográfica.* 2010.
7. **Olmos, Pablo Sastres.** *Sistemas de Información Geográfica (SIG): Técnicas básicas para estudios de biodiversidad.* 2010.
8. Transics. Gestion de Flotas. [En línea] [Citado el: 10 de Noviembre de 2012.] <http://www.transics.com/es/resources/glossary/gestion-de-flotas.html>.
9. The Free Dictionary. [En línea] [Citado el: 1 de Noviembre de 2012.] <http://es.thefreedictionary.com/flota>.
10. **W3C.** W3C Working Group Note. [En línea] W3C, 31 de Octubre de 2007. [Citado el: 13 de Noviembre de 2012.] <http://www.w3.org/TR/dd-landscape/#sec-mobiledevice>.
11. Definicion.de . *Trama.* [En línea] [Citado el: 2 de Noviembre de 2012.] <http://www.definicion.de/trama>.
12. **Oracle Corporation.** *Oracle9i.Database Concepts,Release 2 (9.2).* California : s.n., 2002.
13. **The PostgreSQL Global Development Group.** *PostgreSQL 9.1.0 Documentation.* California : s.n., 2012.
14. **Colectivo de Ecured.** Enciclopedia Cubana en la Red. [En línea] [Citado el: 14 de Noviembre de 2012.] [http://www.ecured.cu/index.php/Metodolog%C3%ADas\\_de\\_desarrollo\\_de\\_software](http://www.ecured.cu/index.php/Metodolog%C3%ADas_de_desarrollo_de_software).
15. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El Proceso Unificado de Desarrollo de Software.* [ed.] Andrés Otero. [trad.] Salvador Sánchez. Madrid : PEARSON EDUCACIÓN S.A, 2000.

## Bibliografía

---

16. RUP. [En línea] 10 de Septiembre de 2010. [Citado el: 12 de Diciembre de 2012.] <http://rup-ing-software.blogspot.com/>.
17. **Orallo, Enrique Hernández.** *El Lenguaje Unificado de Modelado (UML)*.
18. HERRAMIENTAS CASE. [En línea] [Citado el: 10 de Enero de 2013.] <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r88866.DOC>.
19. Plusformacion. Herramientas CASE para el proceso de desarrollo de Software. [En línea] [Citado el: 12 de Enero de 2013.] <http://www.plusformacion.com/Recursos/r/Herramientas-CASE-para-proceso-desarrollo-Software..>
20. Connexions. Los Sistemas de Gestión de Bases de Datos son software especializado. [En línea] 23 de Septiembre de 2008. [Citado el: 12 de Enero de 2012.] <http://cnx.org/content/m17543/latest/>.
21. **Momjian, Bruce.** *PostgreSQL Introduction and Concepts*. s.l. : ADDISON-WESLEY, 2001.
22. PostgreSQL. The world's most advanced open source database. [En línea] PostgreSQL. [Citado el: 13 de Enero de 2013.] <http://www.postgresql.org/about/>.
23. **OSGeo Project.** PostGIS. What is PostGIS? [En línea] [Citado el: 13 de Enero de 2013.] <http://postgis.refractor.net/>.
24. **Gutierrez, Andres Felipe.** *Kumbia PHP Framework. Porque programar debería ser más fácil*.
25. **Digia.** Introducing Qt 5.0. One framework to rule them all. [En línea] [Citado el: 15 de Enero de 2013.] <http://qt.digia.com/product/>.
26. **Pixelco.** Diseño, desarrollo y tecnología. Qt Creator – Completo entorno de desarrollo multiplataforma. [En línea] 10 de Junio de 2010. [Citado el: 14 de Enero de 2013.] <http://pixelcoblog.com/qt-creator-completo-entorno-de-desarrollo-multiplataforma/>.
27. **Bonanata, Maximiliano.** *Programación y Algoritmos. Aprenda a trabajar con los lenguajes C y Pascal*. Buenos Aires : MP Ediciones S.A, 2003.
28. **Stroustrup, Bjarne.** *The C++ Programming Language*. s.l. : Addison-Wesley, 1997.
29. **Costanera Uno S.A.** Costanera Uno. Sinonimo de Nautica. [En línea] [Citado el: 5 de Marzo de 2013.] [http://www.costanerauno.com.ar/preguntasfrecuentes/Qu%C3%A9+es+y+qu%C3%A9+significa+NME%20A\\_15\\_86.htm..](http://www.costanerauno.com.ar/preguntasfrecuentes/Qu%C3%A9+es+y+qu%C3%A9+significa+NME%20A_15_86.htm..)
30. NMEA Data. Introduction. [En línea] [Citado el: 5 de Marzo de 2013.] <http://www.gpsinformation.org/dale/nmea.htm..>

## Bibliografía

---

31. **Floristan, Liset Cabrera.** *Trabajo de Diploma: Módulo de Comunicación para el Sistema de Control de Flotas.* La Habana : Universidad de las Ciencias Informaticas, 2012.
32. **Cespedes, Alexa, y otros.** sabd15N1. [En línea] 2013. [Citado el: 26 de Febrero de 2013.] <http://sabd15n1.wikispaces.com/MySQL>.
33. **Larman, Craig.** *UML y Patrones. Introduccion al analisis y diseño orientado a objetos.* [ed.] Pablo Eduardo Roig Vazquez y Magdiel Gomez Marina. [trad.] Luz Maria Hernandez Rodriguez. Naucalpan de Juarez : PRENTICE HALL, 1999. 970-17-0261-1.
34. [En línea] [Citado el: 26 de Marzo de 2013.] [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lis/marquez\\_a\\_bm/capitulo5.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/marquez_a_bm/capitulo5.pdf).
35. **Rondón, Yoandri Quintana.** *Desarrollo de la rquitectura del proyecto Captura y Catalogación de Medias.* La Habana : Universidad de las Ciencias Informaticas, 2009.
36. **S.Pressman, Roger.** *Ingenieria del Software. Un enfoque practico.* Quinta Edicion. s.l. : McGraw-Hill, 2002.
37. [En línea] [Citado el: 6 de Enero de 2013.] <http://projects.staffs.ac.uk>.
38. ImageShack. [En línea] 2013. [Citado el: 15 de Febrero de 2013.] <http://imageshack.us/photo/my-images/687/capturagps.gif/sr=1>.
39. [En línea] [Citado el: 3 de Marzo de 2013.] <http://e-ducativa.catedu.es>.
40. **NEIRA, MARIO CISTERNA.** *METODOS DE OPTIMIZACION DE CONSULTAS PARA EL LENGUAJE SQL.* Santiago de Chile : Universidad de Santiago de Chile. Facultad de Ciencias. , 2002.
41. **EMS.** SQL Manager.net. Database Management Solutions . [En línea] 2013. [Citado el: 27 de Mayo de 2013.] <http://www.sqlmanager.net/products/postgresql/datagenerator>.
42. Corporación Sybven. [En línea] 2013. [Citado el: 14 de Mayo de 2013.] [http://www.corporacionsybven.com/portal/index.php?option=com\\_content&view=article&id=246..](http://www.corporacionsybven.com/portal/index.php?option=com_content&view=article&id=246..)

## Anexos

### ➤ CUS 3. Insertar datos de tramas GPS.

<b>Objetivo</b>	Extraer de la trama GPS los datos necesarios para el control de flotas, convertirlos a formato de datos espaciales para luego insertarlos en la Base de Datos.
<b>Actores</b>	Sistema
<b>Resumen</b>	El caso de uso inicia cuando el sistema realiza la extracción de los datos necesarios de la sentencia RMC de la trama GPS en formato NMEA, siendo interpretados y convertidos sus valores en dependencia de la proyección cartográfica a utilizar, para luego insertarlos en la Base de Datos.
<b>Referencia</b>	RF3
<b>Complejidad</b>	Alta
<b>Prioridad</b>	Crítico
<b>Precondiciones</b>	La trama GPS se encuentra en formato NMEA y no le faltan datos.
<b>Postcondiciones</b>	Se insertaron en la Base de Datos los datos siguientes: latitud, longitud, velocidad, fecha, hora y el campo espacial "POINT" formado por los valores de la latitud y la longitud.
<b>Flujo normal de eventos</b>	
<b>Flujo básico general</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1	Realiza varias acciones:

## Anexos

	<p>-Extrae datos de tramas GPS. Ver Sección 1: “Extraer datos de tramas GPS”.</p> <p>-Convierte datos de tramas GPS. Ver Sección 2: “Convertir datos de tramas GPS”.</p> <p>-Guarda datos asociados a la trama GPS en la Base de Datos. Ver Sección 3: “Guardar datos asociados a la trama GPS en la Base de Datos”.</p>
2	Termina el caso de uso.

### Sección 1: Extraer datos de tramas GPS.

#### Flujo básico Extraer datos de trama GPS

Acción del Actor	Respuesta del Sistema
1	Se almacena la primera sentencia completa en una estructura de datos.
2	Se validan todos los datos de la sentencia RMC.
3	Extrae de la sentencia RMC los datos por separados de latitud, longitud, velocidad, fecha y hora en su formato original.
4	Guarda los datos extraídos en estructuras de datos.

### Sección 2: Convertir datos de tramas GPS.

#### Flujo básico Convertir datos de tramas GPS.

Acción del Actor	Respuesta del Sistema
1	Obtiene cada dato de la trama GPS por separado: longitud, latitud, velocidad, fecha y hora.
2	Aplica los algoritmos correspondientes a cada dato

## Anexos

		para su conversión según el sistema de proyección cartográfica a emplear.
<b>Sección 3: Guardar datos asociados a la trama GPS en la Base de Datos.</b>		
<b>Flujo básico Guardar datos asociados a la trama GPS en la Base de Datos.</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
<b>1</b>		Inserta los datos en la Base de Datos.
<b>Flujos alternos</b>		
<b>Evento 1. No hay conexión con la Base de Datos.</b>		
<b>Acción del Actor</b>		<b>Respuesta del Sistema</b>
<b>1</b>		El sistema guarda los datos relativos a la trama que está analizando en ese momento en una estructura de datos para su posterior inserción.
<b>Relaciones</b>	<b>CU Incluidos</b>	
	<b>CU Extendidos</b>	
<b>Requisitos no funcionales</b>		
<b>Entradas</b>		
<b>Salidas</b>		

Tabla 4. Descripción textual del CUS Insertar datos de tramas GPS.

### Pruebas al CUS Insertar datos de tramas GPS.

```
void Trabajo_con_TramaGPS::Insertar_Por_Tiempo()
```

```
{
```

## Anexos

---

```
int count = 20;           1
QStringList var;         1
while (count > 0 && !lista_partes.empty())  1
{
    var = lista_partes.first();           2
    lista_partes.removeFirst();          2
    count--;                             2
    if(var.length()===13 && var.at(0)=="$GPRMC")  2
    {
        if(Validar_Datos_deTrama(var))    3
        {
            if(obj->db.open())             4
            {
                QSqlQuery *consulta = new QSqlQuery(obj->db);  5
                QString nombre=var.at(0);    5
                QString hora_bruta=var.at(1);  5
                QTime hora=Convertir_Hora_Postgre(hora_bruta);  5
                QString lat=var.at(3);       5
                double lat_grados=lat.mid(0,2).toDouble();  5
                double lat_min=lat.mid(2,2).toDouble();  5
            }
        }
    }
}
```

## Anexos

---

```
double lat_seg=lat.mid(5,2).toDouble();           5
double latitud= lat_grados+ lat_min*(1/60)+lat_seg*(1/3600);           5
QString cuadrante_latitud=var.at(4);             5
if(cuadrante_latitud=="S")                       5
{
    latitud =-1 * latitud;                       6
}
QString lon=var.at(5);                           7
double lon_grados=lon.mid(0,2).toDouble();       7
double lon_min=lon.mid(2,2).toDouble();          7
double lon_seg=lon.mid(5,2).toDouble();          7
double longitud=lon_grados+lon_min*(1/60)+lon_seg*(1/3600);           7
QString cuadrante_longitud=var.at(6);           7
if(cuadrante_longitud=="W")                     7
{
    longitud = -1 * longitud;                     8
}
int velocidad_bruta_en_nudos=var.at(7).toInt();  9
double vel= velocidad_bruta_en_nudos * 1.852;   9
int velocidad_final=round(vel);                 9
```

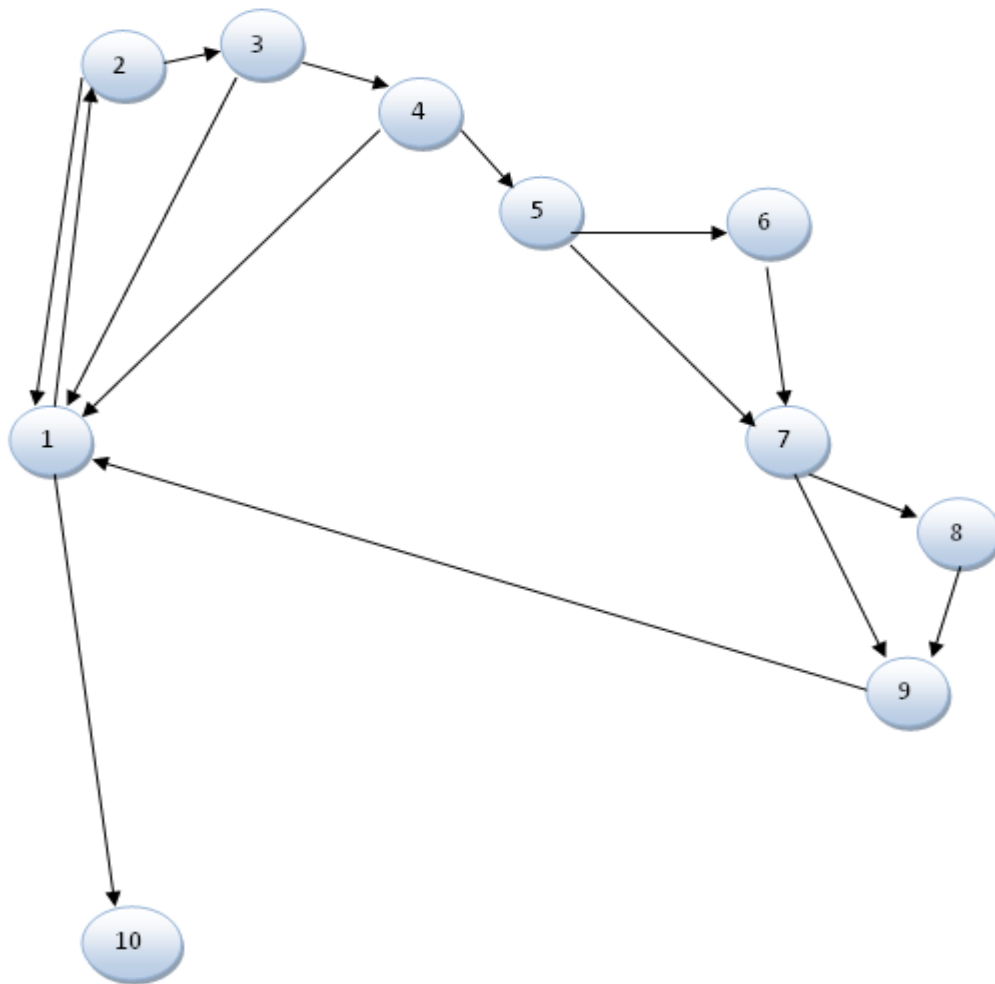


## Anexos

---

```
QString fecha_bruta=var.at(9);          9
    QDate date=Convertir_Fecha_Postgre(fecha_bruta);    9
    consulta->exec(QString("INSERT          INTO          auto
(latitud,longitud,velocidad,fecha,hora,objeto_movil_id,the_geom)          VALUES
("+QString::number(latitud)+"','"+QString::number(longitud)+"','"+QString::number(velocidad_final)+"",
"+date.toString("yyyy/MM/d")+ "','"+hora.toString("HH:mm:ss")+ "','1',
"%1);").arg("ST_SetSRID(ST_MakePoint("+QString::number(latitud)+"','"+QString::number(latitud)+""),4326)")
);          9
    obj->db.close();          9
}
}
}
}
}
}          10
```

**Construcción del grafo de flujo asociado al CUS Insertar datos de tramas GPS.**



**Conjunto básico de caminos independientes.**

**Camino 1:** 1-10

**Camino 2:** 1-2-1-10

**Camino 3:** 1-2-3-1-10

**Camino 4:** 1-2-3-4-1-10

**Camino 5:** 1-2-3-4-5-7-9-1-10

**Camino 6:** 1-2-3-4-5-6-7-9-1-10

## Anexos

---

**Camino 7:** 1-2-3-4-5-6-7-8-9-1-10

**Conjunto de pruebas.**

<b>Prueba Camino 1</b>	Se muestran datos relacionados con la prueba.
<b>Count</b>	20
<b>Lista Var</b>	Vacía
<b>Condición while</b>	Lista_partes.empty() = true
<b>Resultados esperados</b>	El método ha recorrido el camino adecuado debido a que la lista encargada de almacenar el conjunto de tramas está vacía.

<b>Prueba Camino 2</b>	Se muestran datos relacionados con la prueba.
<b>Count</b>	20
<b>Lista Var</b>	Vacía
<b>Condición while</b>	Count >0 y Lista_partes.empty() = false
<b>Lista Var</b>	\$GPRMC,010045.37,A,0000.002,N,00000.005,E,21.60,61.35,040112,0.0,E,A*34
<b>Longitud Lista Var y 1ra Posición de Lista Var</b>	var.length()==12 y var.at(0)=="\$GPRMC"

## Anexos

---

<b>Condición while</b>	Count <=0 ó Lista_partes.empty() = true
<b>Resultados esperados</b>	El método ha recorrido el camino adecuado debido a que la lista encargada de almacenar el conjunto de tramas está vacía ó el valor de la variable Count es menor o igual a cero.

<b>Prueba Camino 3</b>	Se muestran datos relacionados con la prueba.
<b>Count</b>	20
<b>Lista Var</b>	Vacía
<b>Condición while</b>	Count >0 y Lista_partes.empty() = false
<b>Lista Var</b>	\$GPRMC,010045.37,A,0000.002,N,00000.005,E,21.60,61.35,040112,0.0,E,A*34
<b>Longitud Lista Var y 1ra Posición de Lista Var</b>	var.length()==13 y var.at(0)=="\$GPRMC"
<b>Validar datos</b>	False
<b>Condición while</b>	Count <=0 ó Lista_partes.empty() = true

## Anexos

---

<b>Resultados esperados</b>	El método ha recorrido el camino adecuado debido a que la lista encargada de almacenar el conjunto de tramas está vacía ó el valor de la variable Count es menor o igual a cero.
-----------------------------	--

<b>Prueba Camino 4</b>	Se muestran datos relacionados con la prueba.
<b>Count</b>	20
<b>Lista Var</b>	Vacía
<b>Condición while</b>	Count >0 y Lista_partes.empty() = false
<b>Lista Var</b>	\$GPRMC,010045.37,A,0000.002,N,00000.005,E,21.60,61.35,040112,0.0,E,A*34
<b>Longitud Lista Var y 1ra Posición de Lista Var</b>	var.length()==13 y var.at(0)=="\$GPRMC"
<b>Validar datos</b>	True
<b>Conexión a la BD</b>	False
<b>Condición while</b>	Count <=0 ó Lista_partes.empty() = true

## Anexos

---

<b>Resultados esperados</b>	El método ha recorrido el camino adecuado debido a que la lista encargada de almacenar el conjunto de tramas está vacía ó el valor de la variable Count es menor o igual a cero.
-----------------------------	--

<b>Prueba Camino 5</b>	Se muestran datos relacionados con la prueba.
<b>Count</b>	20
<b>Lista Var</b>	Vacía
<b>Condición while</b>	Count >0 y Lista_partes.empty() = false
<b>Lista Var</b>	\$GPRMC,010045.37,A,0000.002,N,00000.005,E,21.60,61.35,040112,0.0,E,A*34
<b>Longitud Lista Var y 1ra Posición de Lista Var</b>	var.length() == 13 y var.at(0) == "\$GPRMC"
<b>Validar datos</b>	True
<b>Conexión a la BD</b>	True
<b>Cuadrante Latitud</b>	N
<b>Cuadrante</b>	E

## Anexos

---

<b>Longitud</b>	
<b>Ejecución de la query</b>	True
<b>Condición while</b>	Count <=0 ó Lista_partes.empty() = true
<b>Resultados esperados</b>	El método ha recorrido el camino adecuado debido a que la lista encargada de almacenar el conjunto de tramas está vacía ó el valor de la variable Count es menor o igual a cero.

<b>Prueba Camino 6</b>	Se muestran datos relacionados con la prueba.
<b>Count</b>	20
<b>Lista Var</b>	Vacía
<b>Condición while</b>	Count >0 y Lista_partes.empty() = false
<b>Lista Var</b>	\$GPRMC,010045.37,A,0000.002,N,00000.005,E,21.60,61.35,040112,0.0,E,A*34
<b>Longitud Lista Var y 1ra Posición de Lista Var</b>	var.length()==13 y var.at(0)=="\$GPRMC"
<b>Validar datos</b>	True

## Anexos

---

<b>Conexión a la BD</b>	True
<b>Cuadrante Latitud</b>	S
<b>Valor latitud</b>	-5638.06
<b>Cuadrante Longitud</b>	E
<b>Ejecución de la query</b>	True
<b>Condición while</b>	Count <=0 ó Lista_partes.empty() = true
<b>Resultados esperados</b>	El método ha recorrido el camino adecuado debido a que la lista encargada de almacenar el conjunto de tramas está vacía ó el valor de la variable Count es menor o igual a cero.

<b>Prueba Camino 7</b>	Se muestran datos relacionados con la prueba.
<b>Count</b>	20
<b>Lista Var</b>	Vacía
<b>Condición while</b>	Count >0 y Lista_partes.empty() = false
<b>Lista Var</b>	\$GPRMC,010045.37,A,0000.002,N,00000.005,E,21.60,61.35,040112,0.0,E,A*34



## Anexos

---

<b>Longitud Lista Var y 1ra Posición de Lista Var</b>	var.length()==13 y var.at(0)=="\$GPRMC"
<b>Validar datos</b>	True
<b>Conexión a la BD</b>	True
<b>Cuadrante Latitud</b>	S
<b>Valor latitud</b>	-5638.06
<b>Cuadrante Longitud</b>	W
<b>Valor longitud</b>	-0015.80
<b>Ejecución de la query</b>	True
<b>Condición while</b>	Count <=0 ó Lista_partes.empty() = true
<b>Resultados esperados</b>	El método ha recorrido el camino adecuado debido a que la lista encargada de almacenar el conjunto de tramas está vacía ó el valor de la variable Count es menor o igual a cero.

# *Anexos*

---