

Universidad de las Ciencias Informáticas

Facultad 15



Título: Implementación de un Sistema de Gestión de Información del proceso de Pruebas Técnicas a vehículos del CID del Transporte de las FAR.

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor(es): Cdte: Ernesto Fajardo González

Cdte: Yeiciel Gómez Acosta

Tutor: Tte. Ing. Enier Maceo Cisnero.

Co. Tutor: Tte. Ing. Henry Ernesto Bermúdez Pérez.

Ciudad de La Habana, junio de 2010

“Año 52 de la Revolución”

Declaración de Tutoría

Declaración de Autoría.

Declaramos que somos los autores de la presente tesis con título: **Implementación de un Sistema de Gestión de Información del proceso de Pruebas Técnicas a vehículos del CID del Transporte de las FAR** y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste se firma la presente a los ____ días del mes de _____ del año _____.

Ernesto Fajardo González

Yeiciel Gómez Acosta

Tutor: Tte. Ing. Enier Maceo Cisnero

Co.Tutor: Tte. Ing. Henry Ernesto Bermúdez Pérez.

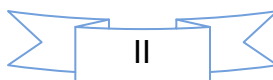


Datos de contacto

Datos de contacto

Tutor: Ing. Tte.: Enier Maceo Cisnero. Actualmente es el jefe del proyecto Sistemas de Dirección de las Transportaciones Militares de la Unidad de Compatibilización, Integración y Desarrollo de Productos Informáticos para la Defensa del país.

Co Tutor: Ing. Tte.: Henry Ernesto Bermúdez Pérez. Actualmente se encuentra trabajando en la Unidad de Compatibilización, Integración y Desarrollo de Productos Informáticos para la Defensa del país.



Dedicatoria

Dedicatoria

Ernesto Fajardo González

Este trabajo de diploma se lo dedico a mis padres y mi padrastro por ser ellos la fuente principal de mi inspiración, de mi quehacer de cada día, a mi hermano que siempre me ha visto como su ejemplo más cercano, a mi familia en sentido general y a todas mis amistades.

Yeiciel Gómez Acosta

Dedico este trabajo a mis padres y a mi familia en general, a todos mis amigos y personas que me quieren y me aprecian, y a todas las personas que me ayudaron en mi formación como estudiante y en el desarrollo de este trabajo diploma.



Agradecimientos

Agradecimientos

Ernesto Fajardo González

En primer lugar a mis padres porque a ellos le debo lo que hoy soy conjuntamente con mi padrastro que siempre me apoyo y se comporto como mi segundo papa, a mi hermano que es más pequeño y siempre ve en mi un ejemplo a seguir, a mis tíos Nela y Eduardo que me han acogido como un hijo mas durante estos cinco años en la UCI, a mis primos que siempre estuvieron a mi lado apoyándome en todo lo posible, a Yacel por ser más que un profesor un hermano.

También le quiero agradecer a mi tía María Teresa, a mi prima Yoania y su esposo Yohanis porque me guiaron cada día desde mi ingreso en la Universidad, a mis amistades los cuales me han acompañado desde el primer momento. Además de ellos también agradecer a mi compañero de tesis que siempre estuvo pendiente a nuestro trabajo de diploma dando lo mejor de él en cada jornada, al tutor que siempre nos ayudo a encontrar una solución a cada una de las problemáticas encontradas en el trabajo y al tribunal que en cada corte nos hicieron las recomendaciones necesarias para perfilar lo mejor posible esta solución.

A todas aquellas personas lindas que la vida me permitió conocer, que me aprecian y me quieren.

Agradecimientos

Yeiciel Gómez Acosta

Primero que todo quiero agradecer a mis padres Bárbara y Cristóbal y a mi hermana Adriana a los cuales dedico éste trabajo diploma ya que siempre me apoyaron a lo largo de toda mi carrera y me han sabido educar correctamente y por eso les agradezco lo que hoy soy y llegare a ser.

A mi sobrinito Daykel que lo quiero con la vida.

A todo el resto de mi familia que también me han ayudado y apoyado en todo momento como mi tía Arlines, mi prima Yeni, Yuni y demás a las cuales quiero mucho y siempre las llevo conmigo.

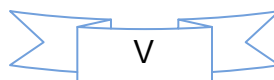
A mi novia que me ayudo y aconsejo mucho y siempre me dio fuerzas para terminar mi tesis.

A dos personas que nunca olvidaré ya que los considero como un padre y una madre más para mí: Ramón Y Elsita a los cuales también dedico este título.

A todos mis amigos y personas que me quieren y que de una forma u otra me han demostrado que puedo contar con ellos en cualquier momento: Danima, Dayanni, Yani, Daylin, Anier, Orlanyer y otros.

Al tutor y al co-tutor que me ayudaron mucho en la realización de este trabajo.

A todos los profesores que han sido protagonistas de mi enseñanza y en especial al profesor Frank y al profesor Yasser.



Resumen

Resumen

El transporte de vehículos es un factor principal en el buen desenvolvimiento de la logística de todos los medios que existen en las Fuerzas Armadas Revolucionarias (FAR). La Unión de la Industria Militar Cubana es la empresa encargada de la producción e importación de vehículos para las FAR. El Centro de Investigación y Desarrollo del Transporte de las FAR (CID DCM TRANS) es el encargado de comprobar que los medios cumplan con los requisitos establecido por los fabricantes y de cierta manera puedan ser explotados eficientemente bajo las condiciones medioambientales existentes en Cuba con el empleo de nuevas tecnologías que disminuyan el tiempo de duración de las pruebas, que permitan una mayor comunicación e intercambio de información entre los especialistas que participan en los experimentos y reduzcan la utilización del papel, posibilitará una mejor comprensión y análisis de los resultados, reduciendo así los costos económicos.

El desarrollo de este trabajo va dirigido a mejorar la gestión de la información de las pruebas en el CID del Transporte, por lo que se da paso a la implementación del Módulo Pruebas Técnicas a los Vehículos del Sistema de Dirección de las Transportaciones Militares para su posterior despliegue.

Para el desarrollo de dicho software se realizó un estudio detallado del proceso de pruebas, empleando la observación, la entrevista y la modelación como métodos fundamentales para la comprensión del problema. También se analizó el estado del arte de los principales sistemas y empresas dedicadas a las pruebas de vehículos, así como se explican brevemente las herramientas que se emplearon en el desarrollo del trabajo.

Con este trabajo se pretende dar un primer paso al desarrollo informático del proceso de pruebas del CID del Transporte de las FAR.

Palabras Claves: Centro de Investigación y Desarrollo del Transporte de las FAR, Unión de la Industria Militar Cubana.

Tabla de contenido

Introducción.....	1
Capítulo 1: Fundamentación Teórica. Estado del Arte.....	4
1.1 Introducción del capítulo.....	4
1.2 Necesidad del sistema.....	4
1.3 Conceptos asociados al dominio del problema.....	5
1.4 Análisis de soluciones existentes.....	6
1.4.1 En el ámbito internacional.....	6
1.4.2 En el ámbito nacional.....	7
1.5 Tecnologías y herramientas utilizadas.....	8
1.5.1 Sistema gestor de Base de Datos propuesto.....	8
1.5.2 Marco de Trabajo SAUXE.....	10
1.5.3 Lenguaje de programación utilizado PHP 5.0.....	14
1.5.4 Java Script.....	15
1.5.5 Servidor web Apache 2.0.....	17
1.5.6 Visual Paradigm.....	18
1.5.7 Easy Eclipse.....	19
1.6 Conclusiones parciales.	20
Capítulo 2: Características del sistema.....	21
2.1 Introducción.....	21
2.2 Objeto de estudio.....	21
2.2.1 Descripción general.....	21
2.3 Problema y situación problemática.....	27
2.4 Flujo actual del proceso.....	29
2.5 Objetos de automatización.	30
2.6 Información que se maneja.	30
2.7 Propuesta del sistema.	30
2.8 Especificación de los requisitos del software.	31

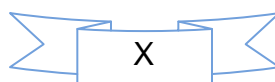
2.8.1 Requisitos funcionales.....	32
2.8.2 Requisitos no funcionales.	34
2.9 Conclusiones parciales.	36
Capítulo 3: Implementación y Prueba.	37
3.1 Introducción.....	37
3.2 Arquitectura utilizada.....	37
3.2.1 Representación gráfica de la arquitectura en 3 capas.	38
3.2.2 Ventajas de esta Arquitectura.	39
3.3 Patrones utilizados.	39
3.3.1. Modelo – vista –controlador.	39
3.4 Diagrama de componentes.	42
3.4.2 Descripción del diagrama de componente de los nomencladores.	44
3.5 Matriz de integración de componentes.	44
3.5.1 Tabla de integración de componentes internos.	45
3.5.2 Tabla de integración de componentes externos.	46
3.6 Pruebas.	46
3.6.1 Diseño de Casos de prueba.	47
3.7 Resumen de las pruebas.	53
3.8 Conclusiones parciales.....	53
Conclusiones del trabajo.....	54
Recomendaciones.....	55
Bibliografía.....	56
Referencias bibliográficas.....	57
Abreviaturas.	58
Glosario de términos.	59
Anexos.....	60

Indice de Tablas

3.5.1	Tabla	de	integración	de	componentes	
	internos.....					45
3.5.2	Tabla	de	integración	de	componentes	
	externos.....					46
3.6.2	Descripción del Caso de prueba Gestionar nomencladores.....					47
3.6.3	Descripción del caso de prueba “Gestionar proyecto”.....					50

Indice de Figuras

3.2.1 Representación gráfica de la arquitectura en 3 capas.....	38
3.3.1. Modelo – vista –controlador.	39
3.4 Diagrama de componentes.	42
Fig. 5. Diagrama de componente de los nomencladores.....	43



Introducción

Introducción

El constante desarrollo y modernización de la técnica militar constituye un proceso en el cual la Industria Militar Cubana (UIM) se ve inmersa en los momentos actuales. Las limitaciones a las que se enfrenta la economía del país impiden la adquisición de todo el armamento que se desearía.

Los medios desarrollados y/o modernizados por la UIM, deben ser expuestos a pruebas con la finalidad de verificar si los prototipos cumplen con las especificaciones del diseño así como su adaptabilidad a las condiciones y regímenes de explotación a los que serán expuestos en el **TOM** (Teatro de Operaciones Militares).

La técnica de transporte no está exenta a estos procesos de cambios, mejoras y pruebas, dado que entre los objetivos del desarrollo de los medios de combate, está la meta de alcanzar la mayor movilidad posible. Ello exige que los vehículos en especial, sean sometidos a rigurosos regímenes de evaluación, actividades de las que se encarga el Centro de Investigación y Desarrollo del Transporte de las FAR (**CID DCM TRANS**).

Los especialistas del CID DCM TRANS trabajan sobre la base de que **probar**:

“no significa realizar una serie de exámenes durante el proceso de adquisición y asimilación, sino una continua recolección y análisis de la información a lo largo del ciclo de vida de las pruebas. La principal función del proceso de prueba es crear una base de conocimientos que sea capaz de recolectar los datos del prototipo, de los instrumentos de medición y analice toda esta información y pueda ser capaz de comprender toda la información y brinde la información necesaria para llegar a una mejor conclusión del sistema que se prueba”.¹

Lograr las metas que se plantean en este concepto, requiere un constante monitoreo de las pruebas e intercambio de información de forma continua y oportuna, por parte de los especialistas del CID DCM TRANS, sin embargo, las condiciones actuales en las que se desarrollan las pruebas impiden que se alcancen de forma rápida y eficiente estos objetivos. En la actualidad el personal a cargo de las pruebas, gestiona la

¹ Programa de Pruebas. Manuel Ollet Nerey. La Habana. 2008. Pág. 3

Introducción

información del proceso con métodos y medios ineficientes que limitan las potencialidades del CID DCM TRANS.

Sobre la base de estas dificultades, se desarrolló una investigación cuya finalidad consistió en realizar una propuesta arquitectónica para un sistema de gestión de información que soporte el proceso de pruebas del CID DCM TRANS. **La presente investigación está encaminada a** mejorar la gestión de información del Proceso de Pruebas Técnicas del CID del Transporte sobre las bases planteadas en la propuesta arquitectónica realizada y la mejora de esta en caso de ser necesario. **Por lo antes expuesto la idea a defender será** desarrollar un sistema de gestión de información para el Proceso de Pruebas Técnicas del CID del Transporte.

Durante el desarrollo del trabajo se **realizará un estudio** del Proceso de Pruebas Técnicas del CID del Transporte aunque los mayores esfuerzos estarán movilizados en función de la gestión de la información del mencionado proceso. Considerando que una solución informática mejorará la gestión de información de Proceso de Pruebas Técnicas a Vehículos del CID del Transporte, **el objetivo del trabajo** consistirá en:

Desarrollar un sistema de gestión de información para el Proceso de Pruebas Técnicas del CID del Transporte contemplando los elementos planteados en la propuesta arquitectónica realizada.

Para ello se proponen las siguientes **tareas** para cumplir los **objetivos**:

- Estudiar el proceso de Prueba Técnicas.
- Modelar la base de datos.
- Implementar el sistema.
- Probar el sistema.

Durante el desarrollo de esta investigación se emplearon diferentes procedimientos que permitieron dar cumplimiento al objetivo propuesto, siendo la **entrevista informal y formal** a los principales especialistas y la **observación** del proceso, los que permitieron razonar y analizar las dificultades que presenta el proceso de pruebas. También se utilizaron métodos como el **Histórico Lógico** que permitió realizar el estudio a nivel nacional e internacional de sistemas similares al que se quiere implementar, así como los procesos de gestión de información de las actividades de las pruebas en el UCID de Transporte de las FAR, el **Hipotético**

Introducción

Deductivo que permitió establecer conclusiones y obtener una idea del funcionamiento del sistema partiendo de la información acumulada como conocimiento y la teoría existente sobre el tema y el de **Análisis de Documentos** que permitió realizar un estudio de los modelos manejados por los trabajadores encargados de la realización de las pruebas.

Para una mejor organización del presente trabajo se ha dividido el contenido del mismo en 3 capítulos estructurados como sigue a continuación:

Capítulo 1: Fundamentación teórica: En este capítulo se recogen los conceptos necesarios para comprender el desarrollo del sistema. Se realizó un estudio acerca de las aplicaciones que existen actualmente y que guardan algún tipo de relación con la que se quiere desarrollar. También se reflejan cuáles fueron las herramientas, los métodos y las tecnologías usadas en la solución del problema.

Capítulo 2: Características del sistema: En este capítulo se realiza todo lo referente a las características que tiene el software a desarrollar basándose principalmente en los requisitos tanto funcionales como no funcionales para satisfacer las necesidades del cliente.

Capítulo 3: Implementación y pruebas al sistema: En este capítulo se realizó todo lo referente a la fase de implementación, donde se implementaron las funcionalidades del software y los diferentes CU necesarios para la solución y la realización de pruebas al sistema con el objetivo de verificar que la misma cumpla con los requisitos del usuario.

Capítulo 1: Fundamentación Teórica

Capítulo 1: Fundamentación Teórica. Estado del Arte.

1.1 Introducción del capítulo.

En este capítulo se describen detalladamente los problemas que dieron origen al desarrollo de esta aplicación informática, explicando detalladamente los conceptos principales que intervienen en el tema de estudio. Se refleja el estado del arte de las pruebas a los vehículos del transporte en las FAR evaluándose la manera en la que se llevan a cabo en el ámbito internacional y nacional analizando los sistemas creados que ayudan e intervienen de una manera u otra en el ciclo de prueba. Además se definen las tecnologías y herramientas que serán empleadas para la construcción del software.

1.2 Necesidad del sistema.

Debido a la alta demanda de pruebas que realiza el Centro de Investigación y Desarrollo del Transporte de las FAR a los diferentes vehículos, necesita un sistema que sea capaz de ofrecer tanto agilidad y rapidez en recepción como seguridad, control y fiabilidad; debido a que se necesita los resultados de las mismas en el menor tiempo posible.

Estos procesos en la actualidad se desarrollan de forma manual lo que trae como consecuencia que los resultados de los experimentos desarrollados no puedan ser observados hasta finalizar los mismos. Además, la recopilación de estos es a través de la herramienta informática Excel, la cual no permite que varios investigadores trabajen en los datos guardados en ella simultáneamente, alcanzando como resultado que para trabajar con ellos los trabajadores deben hacerlo en diferentes copias de estos archivos guardando así una copia del mismo individualmente, lo antes expuesto deriva grandes problemas con la actualización de los datos y un gran volumen de datos.

Los informes que se crean con los resultados finales de las pruebas se guardan en papel, que es un material muy costoso, los mismos no tienen uniformidad y se guardan en cajas desde que se iniciaron estos experimentos en el Centro de Investigación y Desarrollo del Transporte de las FAR provocando que se haga engorroso el trabajo con los mismos y el empleo de tiempo innecesario en la búsqueda de algún documento en caso de ser necesario.

Capítulo 1: Fundamentación Teórica

Es por ello que los sistemas y programas utilizados para el desarrollo de las pruebas no se correspondan con lo que se necesita para procesar la información necesaria de las pruebas.

1.3 Conceptos asociados al dominio del problema.

Desde épocas muy remotas el hombre ha jugado un papel importante en el proceso de transformación, el cual le ha permitido mejorar sus condiciones de vida con el empleo del término probar y conjuntamente con ello tener éxito en las diferentes esferas de la vida, pero a su vez lograr un desarrollo progresivo en la humanidad los cuales lo han convertido en el ser protagónico dentro de esta evolución. El término **probar** es bastante antiguo, teniendo diferentes significados según el lugar y el momento en el que se emplee.

Según la editorial española de Europa, se define **prueba** como:

- “Examen o experimentación para comprobar el buen funcionamiento de alguna cosa o su adecuación a un determinado fin. Ensayo o experimento que se hace de algo, para saber cómo resultará en su forma definitiva.
- Acción y efecto de **probar**.²

Existen varias acepciones del término probar, que raramente se encuentran relacionadas a los vehículos, según la editorial española se define el término **probar** como:

“Hacer examen y experimento de las cualidades de alguien o algo”.

“Hacer prueba, experimentar o intentar algo.”³

Según el programa de pruebas confeccionado en el Centro de Investigación y Desarrollo del Transporte en las FAR probar es:

2 Unidad Editorial, S.A. Diccionarios del mundo. <<http://www.elmundo.es/diccionarios/>>. [consultado: 15 de febrero del 2009].

3 Programa de Pruebas. Manuel Ollet Nerey. La Habana. 2008. Pág. 3

Capítulo 1: Fundamentación Teórica

“Probar, no significa realizar una serie de exámenes durante el proceso de adquisición y asimilación, sino una continua recolección y análisis de la información desde su inicio y a lo largo del ciclo de vida del vehículo. La principal función del proceso de pruebas es comenzar a confeccionar el banco de datos del ciclo de vida, verificando el funcionamiento apropiado de la instrumentación, recolectando y analizando los datos del vehículo y desarrollando una comprensión completa sobre el mismo.”⁴

De las definiciones antes mencionadas, se considera la idea del CID del Transporte de las FAR como la más completa, debido a que el proceso de pruebas a vehículos que se realiza en el país está basada en el análisis de toda la información obtenida del proceso, lo cual permite una mejor comprensión de los resultados obtenidos y la realización de posibles estudios que ayuden a mejorar el desarrollo de los procesos de producción e importación de vehículos.

Las pruebas en las Fuerzas Armadas es una parte vital del desarrollo y de las mejoras continuas del ciclo de vida. Los adelantos tecnológicos aplicados por un equipo de prueba experimentado serán cruciales en la reducción de los costos de la de la prueba, enfocando las mismas hacia las áreas críticas y comprendiendo mejor los resultados en un futuro.

1.4 Análisis de soluciones existentes.

1.4.1 En el ámbito internacional.

En el mundo actual existen varios centros de investigaciones a nivel internacional que se dedican exclusivamente a las pruebas al parque automotor como son:

I. Centro de Pruebas en Lommel Europa.

II. Centro de Seguridad Volvo.

III. Centro de Investigaciones del polígono de Aberdeen de Estados Unidos.

Existen potencias que dedican grandes sumas de dinero a la carrera armamentista, lo cual les ha permitido que se encuentren a la vanguardia en la producción de armas y pertrechos militares. La realización de las

4 Programa de Pruebas. Manuel Ollet Nerey. La Habana. 2008. Pág. 3

Capítulo 1: Fundamentación Teórica

pruebas a cada armamento es fundamental en el éxito de la producción del mismo, en 1940 Estados Unidos crea el Centro de Pruebas del Polígono de ABERDEN con el objetivo de realizar pruebas a sus armas. Para el desarrollo de las pruebas de manera eficiente se auxilian de dos programas informáticos que le ayudan a tener una mejor visión de su posible resultado como son el **SARD (ADACS)** y el **SIOPFA (ATIRS)**.

El Sistema de Informes de las Ocurrencias de Pruebas (**SIOPFA**) es un sistema de base de datos distribuido, que conecta a todo el Comando de Pruebas y Evaluación a las computadoras del Centro de Pruebas del SARD, propiciando que los datos de las ocurrencias de las Pruebas de Desarrollo y las Pruebas Operacionales estén preparados, detenidos, o ejecutándose.

El **SARD** es un sistema de información integrado creado con el objetivo de conducir, grabar y analizar los datos de la mejor manera posible de las pruebas de fiabilidad, disponibilidad, mantenimiento, las prestaciones, el registro de fuego, la vulnerabilidad del blindaje entre otras. Sus datos se manejan de forma electrónica, se distribuyen a través del correo electrónico o Fax, se organizan por proyectos y se guardan en una base de datos permitiendo su disponibilidad para un futuro y generando como resultado final un informe de ocurrencias del proceso de pruebas concluido. Es capaz de gestionar a los usuarios permitiéndoles acceso a diferentes interfaces según el nivel de autorización que tengan, también administra de manera eficiente el sistema de base de datos (**SIOPFA**), realizando salvadas diarias para minimizar los daños que puedan ocurrir en un futuro.

1.4.2 En el ámbito nacional.

En el ámbito nacional no existe hasta estos momentos un sistema que se encargue del registro de las pruebas, solo existe el Instituto Nacional del Transporte que viabiliza un poco este proceso y el Centro de Investigación y Desarrollo del transporte en las FAR que es quien se encarga de realizar pruebas a vehículos del MITRANS y a los carros blindados de las FAR y al cual este software va dirigido.

Capítulo 1: Fundamentación Teórica

1.5 Tecnologías y herramientas utilizadas.

1.5.1 Sistema gestor de Base de Datos propuesto.

PostgreSQL es uno de los motores de base de datos relacionales más potentes que existen actualmente. Permite modificar su código añadiéndole el soporte para el lenguaje SQL surgiendo así el mismo. Ofrece la posibilidad de ejecutar y trabajar varios procesos al mismo tiempo sobre la misma tabla sin ser dañada, donde cada usuario obtiene una versión de lo último que ha hecho evitando la pérdida de información. Provee soporte para números de precisión arbitraria, texto largo ilimitado, figuras geométricas, direcciones de IPv4 y IPv6, direcciones MAC, arreglos, permite a los usuarios crear sus propios tipos de datos. Tiene su propio lenguaje PL/PgSQL, pero también se pueden usar lenguajes como C, C++, Gambas, Java PL, Java Web, Perl, Php, Python.

Este gestor ofrece muchas ventajas respecto a otros sistemas de bases de datos como son:

Instalación ilimitada:

Es frecuente que las bases de datos comerciales sean instaladas en más servidores de lo que permite la licencia. Algunos proveedores comerciales consideran a esto la principal fuente de incumplimiento de licencia. Con PostgreSQL, nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del software.

Esto tiene varias **ventajas** adicionales como son:

- Modelos de negocios más rentables con instalaciones a gran escala.
- No existe la posibilidad de ser auditado para verificar cumplimiento de licencia en ningún momento.
- Flexibilidad para hacer investigación y desarrollo sin necesidad de incurrir en costos adicionales de licenciamiento.

Mejor soporte que los proveedores comerciales:

Además de las ofertas de soporte, tenemos una importante comunidad de profesionales y entusiastas de PostgreSQL de los que su compañía puede obtener beneficios y contribuir.

Capítulo 1: Fundamentación Teórica

Ahorros considerables en costos de operación:

El software ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que los productos de los proveedores comerciales, conservando todas las características, estabilidad y rendimiento. Además de esto, nuestros programas de entrenamiento son reconocidamente mucho más costo-efectivos, manejables y prácticos en el mundo real que aquellos de los principales proveedores comerciales.

Estabilidad y confiabilidad legendarias:

En contraste a muchos sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad.

Extensible:

El código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.

Multiplataforma:

PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas.

Diseñado para ambientes de alto volumen:

PostgreSQL usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes. Los principales proveedores de sistemas de bases de datos comerciales usan también esta tecnología, por las mismas razones.

Herramientas gráficas de diseño y administración de bases de datos:

Existen varias herramientas gráficas de alta calidad para administrar las bases de datos (pgAdmin, pgAccess) y para hacer diseño de bases de datos (Tora, Data Architect).

Capítulo 1: Fundamentación Teórica

1.5.2 Marco de Trabajo SAUXE.

La Unidad de Compatibilización, Integración y Desarrollo de Software para la Defensa (UCID) tiene definido en su marco de trabajo una serie de tecnologías para el desarrollo de aplicaciones de Gestión de Información como son Zend Framework , Doctrine, Ext JS las cuales vamos describir a continuación. Primeramente se describirá lo que es un framework.

La palabra inglesa framework define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

En el desarrollo de software, un framework es una estructura conceptual y tecnológica de soporte definida, normalmente con artefactos de software concretos, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros programas para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

Los mejores frameworks en JavaScript son: MooTools, Dojo, ExtJS, CLEAN, RIALTO, QooxDoo, script.aculo y MochiKit.

1.5.2.1 Zend Framework.

Es un framework de código abierto para desarrollar aplicaciones web y servicios web con PHP 5. ZF es una implementación que usa código cien por ciento orientado a objetos. Es a menudo llamado "biblioteca de componentes", porque tiene muchos componentes débilmente acoplados que se puede utilizar más o menos independiente. Ofrece un gran rendimiento y una robusta implementación MVC, una abstracción de base de datos fácil de usar, y un componente de formularios que implementa la prestación de formularios HTML, validación y filtrado para que los desarrolladores puedan consolidar todas las operaciones usando de una manera sencilla la interfaz orientada a objetos. Brinda soluciones para construir sitios web modernos,

Capítulo 1: Fundamentación Teórica

robustos y seguros. Además es Open Source y trabaja con PHP 5 a diferencia de CakePHP que trabaja con PHP 4 y PHP 5.

Zend ofrece la depuración integral y soporte de pruebas que le ayuda a analizar y resolver problemas rápidamente. Mediante el apoyo a nivel local y remoto (en el servidor) de depuración, de perfiles, la inspección del código y una solución rápida, la generación de prueba y elaboración de informes, usted tiene las herramientas que necesita para diagnosticar y resolver problemas rápidamente, mientras que garantizar la aplicación de su código se inicia correctamente. Estrecha integración con Zend velocidades de servidor análisis de causa raíz de los problemas detectados en las pruebas, puesta en escena o el entorno de producción.

El mismo presenta las siguientes características:

- Todos los componentes son completamente orientado a objetos de PHP 5, y son compatibles con E_STRICT.
- Uso a voluntad la arquitectura con componentes débilmente acoplados y un mínimo de interdependencias.
- Apoyo a los sistemas de bases de datos múltiples y proveedores, incluyendo MySQL, Oracle, IBM DB2, Microsoft SQL Server, PostgreSQL, SQLite, y de Informix Dynamic Server.
- Redacción de correo electrónico y la entrega, la recuperación a través de mbox, Maildir, POP3 e IMAP4.
- Sub-sistema de almacenamiento en caché flexible con soporte para muchos tipos de backends, tales como la memoria o de un sistema de archivos.

Después del estudio realizado encontramos las siguientes ventajas que brinda su uso:

- Reduce el "time to market" de las aplicaciones, permitiendo ofrecer presupuestos más ajustados.
- Estandariza los procesos más frecuentes, dotándolos de gran robustez.

Capítulo 1: Fundamentación Teórica

- Facilita el mantenimiento de las aplicaciones.
- Ofrece muchas facilidades para el acceso a recursos avanzados (Web services securizados, por ejemplo) que de otro modo resultan bastante más costosos de desarrollar.
- A diferencia de otros frameworks, es posible utilizarlo en modo "desacoplado", es decir, aquellas clases o componentes que sean necesarios en cada proyecto, sin arrastrar todo el framework detrás para cualquier pequeña necesidad.
- Tiene el respaldo de la propia ZEND, creadora de PHP, lo que asegura su continuidad futura tanto como la del propio lenguaje PHP.

1.5.2.2 Doctrine.

Doctrine es un mapeador de objetos relacionales (ORM) para PHP 5.2.3 + que se sienta encima de una potente capa de abstracción de bases de datos (dbal). Una de sus principales características es la opción de escribir las consultas de base de datos en un objeto de propiedad orientada dialecto SQL llamada Doctrine Query Language (DQL), inspirado en Hiberna HQL. Esto proporciona a los desarrolladores una poderosa alternativa a SQL que mantiene la flexibilidad, sin necesidad de la duplicación de código innecesaria.

Un ORM o (Object Relation Mapper) es una técnica de programación que nos permite convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, es decir, las tablas de nuestra base de datos pasan a ser clases y los registros objetos que podemos manejar con facilidad.

Utilizar un ORM tiene una serie de ventajas que nos facilitan enormemente tareas comunes y de mantenimiento:

- Reutilización: La principal ventaja que aporta un ORM es la reutilización permitiendo llamar a los métodos de un objeto de datos desde distintas partes de la aplicación e incluso desde diferentes aplicaciones.
- Encapsulación: La capa ORM encapsula la lógica de los datos pudiendo hacer cambios que afectan a toda la aplicación únicamente modificando una función.

Capítulo 1: Fundamentación Teórica

- **Portabilidad:** Utilizar una capa de abstracción nos permite cambiar en mitad de un proyecto de una base de datos MySQL a una Oracle sin ningún tipo de complicación. Esto es debido a que no utilizamos una sintaxis MySQL, Oracle o SQLite para acceder a nuestro modelo, sino una sintaxis propia del ORM utilizado que es capaz de traducir a diferentes tipos de bases de datos.
- **Seguridad:** Los ORM suelen implementar mecanismos de seguridad que protegen nuestra aplicación de los ataques más comunes como SQL Injections.
- **Mantenimiento del código:** Gracias a la correcta ordenación de la capa de datos, modificar y mantener nuestro código es una tarea sencilla.

Cuando se trabaja con Doctrine, se necesita informar a su motor interno de cuál es el modelo de nuestra aplicación, para ello podemos hacer **ingeniería inversa** de nuestra base de datos existente, o si empezamos la aplicación desde 0, crear el modelo en la sintaxis específica que nos propone Doctrine y luego generar toda la base de datos.

1.5.2.3 Ext JS.

En la actualidad existen numerosas alternativas para disminuir la carga laboral de los programadores, sobre todo en la generación automática de código que permite este propósito. Ext. es un framework de java script permite hacer un modelo de componentes, modelo de contenedores, capas, tablas, plantillas, vistas de datos, sirve de puente entre las librerías JS más usadas (Prototype, YUI). Debido a que se inicio como una extensión de la herramienta YUI esta presenta una cierta ventaja de compatibilidad respecto a las demás. Presenta interfaces visuales agradables a la vista del usuario y es capaz de acoplarse fácilmente a los lenguajes de comunicación con la base de datos tales como PHP, Python, MySQL, entre otros. En estos momentos está dentro de los estándares de programación del UCID como lenguaje java script, desarrollándose con ambiente desktop.

Una de las grandes ventajas de utilizar ExtJS es que nos permite crear aplicaciones complejas utilizando componentes predefinidos así como un manejador de layouts similar al que provee Java Swing, gracias a esto provee una experiencia consistente sobre cualquier navegador, evitando el tedioso problema de validar que el código escrito funcione bien en cada uno (Firefox, IE, Safari, etc.). Además la ventana flotante que provee

Capítulo 1: Fundamentación Teórica

ExtJS es excelente por la forma en la que funciona. Al moverla o redimensionarla solo se dibujan los bordes haciendo que el movimiento sea fluido lo cual le da una ventaja inmensa frente a otros.

Usar un motor de render como ExtJS nos permite tener además estos **beneficios** entre los que se encuentran:

- Existe un balance entre Cliente – Servidor. La carga de procesamiento se distribuye, permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo.
- Comunicación asíncrona. En este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente se dé cuenta.
- Eficiencia de la red. El tráfico de red puede disminuir al permitir que la aplicación elija que información desea transmitir al servidor y viceversa, sin embargo la aplicación que haga uso de la pre-carga de datos puede que revierta este beneficio por el incremento del tráfico.

1.5.3 Lenguaje de programación utilizado PHP 5.0.

Los lenguajes de programación son herramientas que nos permiten crear programas y software. Están constituidos por un grupo de reglas gramaticales y símbolos utilizables. A continuación se describe el lenguaje de programación seleccionado para la confección de la herramienta de prueba y se fundamenta el porqué de su elección.

PHP es un acrónimo recursivo que significa PHP Hypertext Pre-processor. Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es un lenguaje de programación del lado del servidor, gratuito e independiente de plataforma, rápido, con una gran librería de funciones, mucha documentación y capaz de ejecutarse en el servidor web. Es multiplataforma y multiparadigma, permite la realización de aplicaciones web por la cantidad de librerías que cuenta. Posee una amplia documentación y permite la técnica de programación orientada a objetos. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red.

Capítulo 1: Fundamentación Teórica

Es compatible con numerosas bases de datos de MySQL, PostgreSQL, Oracle e Informix. También existen Gestores de Contenido (CMS) desarrollados en php que ayudan al programador tales como Drupal, Joomla, y otros. Además en la Unidad de Compatibilización e Integración y Desarrollo para la Defensa ha desarrollado un marco de trabajo en este lenguaje el cual nos permite tener una amplia comunidad de desarrolladores que contribuyen con nuestro progreso.

Entre sus **ventajas** se pueden encontrar:

- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Biblioteca nativa de funciones sumamente amplia e incluida.
- Tiene manejo de excepciones (desde PHP5).
- Permite aplicar técnicas de programación orientada a objetos.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).

1.5.4 Java Script.

Java Script es un lenguaje de scripting basado en objetos, utilizado para acceder a objetos en aplicaciones. Se caracteriza por ser un lenguaje basado en prototipos, con entrada dinámica y con funciones de primera clase, es interpretado por todos los navegadores modernos y puede incluirse en cualquier documento, es compatible con HTML en el navegador del cliente, ya sea PHP, Active Server Pages, ASP, JSP y SVG. Se usa fundamentalmente para el lado del servidor, implementado como un componente del navegador web, permitiendo el desarrollo de interfaces de usuario mejorada y sitios web dinámicos.

Es un lenguaje que puede ser utilizado por profesionales y para quienes se inician en el desarrollo y diseño de sitios web. No requiere de compilación ya que el lenguaje funciona del lado del cliente, los navegadores son los encargados de interpretar estos códigos. Java script es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con Java script podemos crear diferentes efectos e interactuar con nuestros usuarios. Este lenguaje posee varias características, entre ellas podemos mencionar que es un lenguaje basado en acciones que posee menos restricciones. Además, es un lenguaje que utiliza Windows y sistemas X-Windows, gran parte de la programación en este lenguaje está centrada en describir

Capítulo 1: Fundamentación Teórica

objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros.

Como todos los lenguajes de programación tiene ventajas y desventajas que las abordamos a continuación:

- **El lenguaje de scripting es seguro y fiable** porque está en claro y hay que interpretarlo, por lo que puede ser filtrado; para el mismo Java script, la seguridad es casi total y sólo en su primera versión el CIAC (Computer Incident Advisory Committee) señaló problemas de leve entidad, entre ellos la lectura de la caché y de los sitios visitados, de la dirección e-mail y de los archivos presentes en el disco. Sin embargo, estos fallos se corrigieron ya en las versiones de Netscape sucesivas a la 2.0.
- **Los script tienen capacidades limitadas, por razones de seguridad**, por lo cual no es posible hacer todo con Java script, sino que es necesario usarlo conjuntamente con otros lenguajes evolucionados, posiblemente más seguros, como Java. Dicha limitación es aún más evidente si queremos operar en el hardware del ordenador, como, por ejemplo, la fijación en automático de la resolución vídeo o la impresión de un documento.
- Un problema importante es que **el código es visible y puede ser leído por cualquiera**, incluso si está protegido con las leyes del copyright. Esto, que en mi opinión es una ventaja, representa el precio que tiene que pagar quien quiere utilizar el web: la cuestión de los derechos de autor ha asistido a una verdadera revolución con la llegada de Internet (citamos, como ejemplo más representativo, el MP3). La tutela que proporcionan las leyes actuales resulta débil e inadecuada, por lo que la única solución es tomarse las cosas con filosofía.
- El código Java script se ejecuta en el cliente por lo que el servidor no es solicitado más de lo debido; un script ejecutado en el servidor, sin embargo, sometería a éste a dura prueba y los servidores de capacidades más limitadas podrían resentir de una continua solicitud por un mayor número de usuarios.
- El código del script debe descargarse completamente antes de poderse ejecutar y ésta es la otra cara de la moneda de lo que hemos dicho anteriormente: si los datos que un script utiliza son muchos (por ejemplo, una recopilación de citas que se mostrara de manera casual), el tiempo que tardará en descargarse será muy largo, mientras que la interrogación de la misma base de datos en el servidor sería más rápida.

Capítulo 1: Fundamentación Teórica

1.5.5 Servidor web Apache 2.0.

Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras que implementan el protocolo HTTP/1.1 y la noción de sitio virtual. Presenta entre otras características altamente configurables bases de datos de autenticación y negociado de contenido; a continuación se mencionan una serie de características que lo convierten en el servidor web más utilizado en la actualidad.

- Corre en una multitud de Sistemas Operativos lo que lo hace prácticamente universal.
- Es una tecnología gratuita de código fuente abierta. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Esto le da una transparencia a este software de manera que si se quiere ver que es lo que se instalando como servidor, lo podemos saber, sin ningún secreto, sin ninguna puerta trasera.
- Es un servidor altamente configurable de diseño modular y es sencillo ampliar las capacidades del servidor Web Apache. Actualmente existen muchos módulos para Apache que son adaptables a este, otro punto importante es que cualquiera que posea una experiencia decente en la programación de C o Perl puede escribir un modulo para realizar una función determinada.
- Trabaja con gran cantidad de Perl, PHP y otros lenguajes de script. Perl se destaca en el mundo del script y Apache utiliza su parte del pastel de Perl tanto con soporte CGI como con soporte Perl.
- También trabaja con Java y páginas jsp, teniendo todo el soporte que se necesita para tener páginas dinámicas.
- Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto. Tiene una alta configurabilidad en la creación y gestión de log. Apache permite la creación de ficheros de log a medida del administrador, de este modo puedes tener un mayor control sobre lo que sucede en tu servidor.

Capítulo 1: Fundamentación Teórica

También tiene como **ventajas** los siguientes planteamientos:

- El paquete del servidor es más flexible en tiempo de ejecución porque el proceso actual del servidor puede ser ensamblado en tiempo de ejecución por medio de LoadModule en httpd.conf en lugar de hacerlo por medio de la configuración en tiempo de compilación. De este modo se pueden arrancar diferentes instancias del servidor (estándar, versión SSL, mínima, versión potenciada [PHP, etc.], etc.) con una única instalación de Apache.
- El paquete del servidor puede ser fácilmente ampliado con módulos de terceros incluso después de la instalación. Esto representa un gran beneficio para los que mantienen paquetes, ya que les permite crear el paquete del núcleo de Apache y adicionalmente paquetes que contengan extensiones como PHP, mod_perl, mod_fastcgi.
- Mayor facilidad en los prototipos de módulos Apache porque con DSO y apxs se puede trabajar fuera del árbol fuente de Apache y necesitar un único comando apxs -i seguido de un apachectl restart para cargar una nueva versión del módulo desarrollado en el servidor Apache que esté funcionando actualmente.

1.5.6 Visual Paradigm.

Visual Paradigm es una potente herramienta Case muy fácil de utilizar y que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML.

Presenta un diseño centrado en casos de uso y proporciona a los desarrolladores de software una interfaz simple y amigable, con muchas opciones tales como: diversidad de idiomas, generación de código para varios lenguajes de programación posee facilidad para la instalación y actualización, así como compatibilidad entre sus ediciones. También facilita la interoperabilidad con otras herramientas Case y la mayoría de los

Capítulo 1: Fundamentación Teórica

principales entornos de desarrollo integrados (IDE). Presenta licencia gratuita cuando es usada para el sistema operativo Linux.

1.5.7 Easy Eclipse.

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido". Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados. Es también una comunidad de usuarios, extendiendo constantemente las áreas de aplicación cubiertas. Los siguientes componentes constituyen la plataforma de cliente enriquecido:

- Plataforma principal - inicio de Eclipse, ejecución de plugins.
- OSGi - una plataforma para bundling estándar.
- El Standard Widget Toolkit (SWT) - Un widget toolkit portable.
- JFace - manejo de archivos, manejo de texto, editores de texto.
- El Workbench de Eclipse - vistas, editores, perspectivas, asistentes.

El entorno de desarrollo integrado (IDE) de Eclipse emplea módulos (en inglés plug-in) para proporcionar toda su funcionalidad al frente de la plataforma de cliente rico, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Adicionalmente a permitirle a Eclipse extenderse usando otros lenguajes de programación como son C/C++ y Python, permite a Eclipse trabajar con lenguajes para procesado de texto como LaTeX, aplicaciones en red como Telnet y Sistema de gestión de base de datos. El SDK de Eclipse incluye las herramientas de desarrollo de Java, ofreciendo un IDE con un compilador de Java interno y un modelo completo de los archivos fuente de Java. Esto permite técnicas avanzadas de refactorización y análisis de código.

Eclipse dispone de un Editor de texto con resaltado de sintaxis. La compilación es en tiempo real. Tiene pruebas unitarias con JUnit, control de versiones con CVS, integración con Ant, asistentes (wizards) para creación de proyectos, clases, tests, etc., y refactorización. Eclipse es un lenguaje multi-entorno de software de desarrollo que comprende un entorno de desarrollo integrado (IDE) y un enchufe extensible en el sistema.

Capítulo 1: Fundamentación Teórica

1.6 Conclusiones parciales.

En este capítulo, se abordaron los objetivos que se quieren lograr con el desarrollo de este sistema de gestión de información de Pruebas Técnicas del CID del Transporte para ayudar a elevar el nivel de informatización de dicho centro. Además se explicó detalladamente en qué consistía el objeto de estudio para comprender los dilemas a lo que se enfrenta este proceso en la actualidad. También se realizó un estudio profundo acerca de las tecnologías y herramientas que se van a utilizar en la creación del producto.

Se llega a la conclusión por el estudio realizado, de que no existe ningún sistema integral o de gestión en nuestro país, que sea capaz de satisfacer las necesidades que se plantean en la problemática vista anteriormente, ya que las herramientas que se emplean no son las mejores que se pueden usar en estos momentos, teniendo en cuenta el desarrollo en la rama de la informática que existe en Cuba por lo que se hace necesario el desarrollo de un SGI para el Proceso de Pruebas Técnicas del CID del Transporte.

Capítulo 2: Características del sistema

Capítulo 2: Características del sistema.

2.1 Introducción.

El presente capítulo tiene como objetivo dar a conocer las características principales del sistema a desarrollar, siguiendo de cerca el problema por el cual fue concebido. Además se especifican las funcionalidades que se automatizarán en la búsqueda de satisfacer las necesidades de los usuarios, así como la lista de requisitos que la presente solución debe de tener para cumplir con lo que el cliente o usuario final necesita. También se describe los diferentes procesos que componen el problema a resolver, es decir, el flujo actual de los mismos, se detalla la información que se maneja en el centro y se da a conocer la propuesta de solución que se realiza.

2.2 Objeto de estudio.

2.2.1 Descripción general.

La actual infraestructura de transporte del país posibilita la conexión entre todos los puntos del territorio nacional en un relativo corto plazo de tiempo pero muchos de los prototipos que se utilizan son importados y se desconoce si el medio cumple con las condiciones para realizar determinada transportación.

Las Fuerzas Armadas Revolucionarias, al igual que en otras esferas, han jugado un papel protagónico en la producción e importación de los vehículos para el país. Con el objetivo de comprobar si cada uno de estos recursos se adaptan a las condiciones de nuestro país surge el Centro de Investigación y Desarrollo del Transporte en las FAR (CID DCM TRANS), el cual aplica rigurosas pruebas tratando de descubrir las insuficiencias durante el proceso de prueba.

También se tiene en cuenta los diferentes tipos de análisis que se le quieran realizar al vehículo, las cuales se diferencian tanto por el objetivo de ensayo como por la designación de estas, los métodos de realización, entre otros. Estas están divididas en cinco grandes los grupos los cuales se exponen a continuación:

Por el objeto de prueba, dentro del conjunto de trabajo para crear un nuevo modelo o modernizado de vehículo, las pruebas se dividen en:

Capítulo 2: Características del sistema

- a) De modelos de vehículos en experimentación (o de maquetas), para nuevos o modernizados y sus modificaciones.
- b) De modelos de series que se establecen, (primeras partidas de fábricas) de los vehículos de nuevos modelos.
- c) De modelos de series, o sea de producciones normales (modelos bases o sus modificaciones).
- d) De modelos de reparación general, para los vehículos que han pasado la reparación general.

Por el lugar donde se realizan las pruebas, estas se dividen en:

- De banco (en condiciones de laboratorio).
- De polígono (en rutas experimentales y construcciones de polígonos).
- De camino (por caminos de integración general).
- De explotación (en empresas o unidades de transporte; en las empresas productoras experimentales).
- En condiciones especiales (en regiones montañosas, de alta humedad, de temperaturas extremas, entre otros.).

Según las peculiaridades de cada prueba, se clasifican de la siguiente forma atendiendo al volumen y los objetivos:

De terminación.

Esta prueba tiene como objetivo comprobar si la construcción de vehículo se corresponde con los parámetros de explotación, la documentación del mismo y las normas empleadas. Chequea la capacidad de trabajo del modelo, el volumen de los trabajos de mantenimiento, las posibilidades de sustitución de determinados agregados, conjuntos y piezas. Además permite determinar los datos del pesaje, las insuficiencias constructivas, tecnológicas y productivas del vehículo; por último verifica que las medidas tomadas para erradicar estos defectos se cumplan correctamente.

Capítulo 2: Características del sistema

Previas (de fábricas).

Determina los valores finales obtenidos como resultado de las pruebas y su correspondencia con la documentación técnica. Comprueba el nivel de confiabilidad del vehículo, la correspondencia del mismo con las normas internacionales para la explotación, los requisitos establecidos por el cliente y establece la posibilidad de presentar el medio pasando las pruebas de aceptación satisfactoriamente. Estas dos pruebas iniciales se realizan durante el proceso de elaboración del nuevo vehículo y se pueden ir desarrollando simultáneamente.

De aceptación.

Examina la correspondencia de la tarea técnica, normas y documentación del proyecto con el nuevo modelo, así como la correspondencia del mismo con los estándares internacionales de seguridad y toxicidad y los requisitos establecidos para la explotación. Evalúa el nivel técnico del medio, sus cualidades constructivas y de explotación. Detecta el volumen de trabajos realizados para la construcción del vehículo. El programa de pruebas de aceptación se realiza por una comisión de acuerdo con la entidad solicitante y es aprobado por la dirección de Tanques y Transportaciones del MINFAR o por el ministerio encargado de su producción.

De series estables.

Estudia la efectividad de las medidas para eliminar las insuficiencias detectadas en el proceso de las pruebas de aceptación. Aprueba las tecnologías de fabricación del vehículo sobre la base de la evolución de la calidad del mismo, sus agregados y conjuntos. Controla la calidad del modelo de vehículo de las producciones seriadas.

De explotación (en condiciones concretas).

Evalúa que las condiciones de vehículo se correspondan con las exigencias de explotación. Recolecta los datos para determinar la confiabilidad en las condiciones de explotación dadas. Puntualiza los índices de explotación y la tecnología de mantenimiento y reparación. Determina los niveles de seguridad, toxicidad y el cumplimiento de las exigencias establecidas para la explotación en esta materia. Determina la confiabilidad de la prueba y detecta el volumen acceso del acabado constructivo del modelo.

Capítulo 2: Características del sistema

De control.

Su objetivo es comprobar la correspondencia del vehículo con los requisitos técnicos de control de la calidad de producción, del arme, de las regulaciones, modificaciones y acabado del vehículo, así como la confiabilidad del mismo, sus agregados conjuntos y piezas en los límites establecidos (con frecuencia “de garantía”), de recorrido (durante las pruebas prolongadas de control). Evalúa la efectividad de las medidas constructivas y tecnológicas introducidas en la producción, en el periodo transcurrido. Después de la fabricación del vehículo, sometido a pruebas de control anteriores (prolongadas).

Determinativas.

Determina las cualidades de explotación del vehículo con una exactitud y probabilidad de confianza dada.

De recurso.

Las pruebas de recursos se establecen en dependencia de las condiciones del terreno, climáticas y de explotación del recurso hasta su primera reparación general del ciclo de vida, al igual que de los recorridos inter-reparacionales, sistemas y agregados hasta su estado límite o baja. Su objetivo principal es establecer o confirmar la duración del modelo. Las pruebas de recurso se realizan en los polígonos de vehículos, en las rutas de empleo general con la carga establecida mediante el lastrado del vehículo o en las unidades y entidades de transporte durante la transportación de la carga real.

De entrega-recepción.

Comprueba si el medio se corresponde con las exigencias de las condiciones técnicas y de otra documentación técnica.

De certificación.

Evalúa el nivel técnico y la calidad de la producción del vehículo.

Científicas e investigativas.

Esta se encarga de estudiar los procesos de trabajo que ocurren en los mecanismos de los vehículos, así como especificar los parámetros que determinan la evaluación y caracterización del prototipo utilizado con el medio exterior (por ejemplo los coeficientes de resistencia del aire, resistencia al movimiento, de adherencia de los neumáticos con el terreno, entre otros).

Capítulo 2: Características del sistema

Especiales.

Verifica la correspondencia del vehículo con las exigencias especiales y de utilidad para cumplir determinadas tareas, por ejemplo; para transportar cargos especiales, así como para la explotación en condiciones especiales. Cada una de estas pruebas está controlada por diferentes grupos que son los encargados de comprobar todos los parámetros físicos de la prueba.

Para una mejor organización dentro del proceso de las pruebas que se realizan, los especialistas del centro han dividido las mismas en subsistemas como se refleja a continuación:

- Evaluación mediante el diagnóstico general y biométrico.
- Evaluación de la seguridad del vehículo.
- Evaluación de las características dinámicas.
- Evaluación del consumo de combustible.

Evaluación de la superación de pendientes (por zonas montañosas).

- Evaluación de opacidad de los gases de escape.
- Evaluación del sistema eléctrico.
- Evaluación de la movilidad nocturna.
- Evaluación de las modificaciones a los sistemas y mecanismos.

El ciclo de vida de una prueba está constituido por diferentes fases, donde todas ellas están estrechamente relacionadas y dependen del éxito de la anterior, estas son:

- **Recepción del vehículo.**
- **Preparación del vehículo para la prueba.**
- **Ejecución de las pruebas.**

Para garantizar una alta calidad de las pruebas, una mejor seguridad de las mismas y una mejor comparación entre ellas, se escogen los polígonos como la vía más efectiva de prueba, a continuación se mencionan sus diferentes clasificaciones:

Capítulo 2: Características del sistema

- Vías dinamométricas para medir los parámetros meteorológicos.
- Vías con ondas rectas.
- Vías de cantos con trozos de piedra.
- Vías de cantos grandes.
- Vías de ondas inclinadas.
- Vías de colinas de cuñas, rectas e inclinadas.
- Terraplén enfangado, en buen y mal estado.
- Campo traviesa.
- Terreno pantanoso.
- Estanque de aguas profundas y poco profundas.

Los vehículos y medios técnicos que se someten a prueba pueden ser:

- Vehículos de carga.
- Vehículos y auto cisternas especializados.
- Auto tren.
- Remolcadores de vehículos.
- Remolques y semirremolques de vehículos.
- Vehículos ligeros.
- Ómnibus.
- Cargadores y autocargadores.
- Vehículos contra incendio.

Capítulo 2: Características del sistema

- Vehículos para los servicios comunales y de uso general.
- Motores de vehículos.
- Medios especializados de transporte para la transportación de diferentes tipos de equipamientos.
- Fuentes de energía eléctrica y componentes del sistema de encendido.
- Equipamiento de garaje.
- Vehículos de esteras.
- Vehículos de esteras para los complejos coheteriles.

Después del estudio del proceso de prueba podemos decir que existen herramientas ofimáticas que agilizan el tiempo durante todo el ciclo de vida de la prueba como el Excel pero no satisfacen las necesidades actuales del CID, por lo que una posible solución sería aplicar inteligentemente la tecnología para que el equipo de prueba pueda tener una mejor visión de los posibles resultados de las pruebas.

2.3 Problema y situación problemática.

La constante actividad que existe en el Centro de Investigación y Desarrollo del transporte en las FAR debido a las demandas que existen con respecto a las pruebas de diferentes vehículos no permiten que algunas pruebas se realicen en el menor tiempo posible, por lo que exige contar con información precisa en un momento determinado que permita tener una visión mejor de todos los objetivos que se persiguen con la realización del experimento.

El sistema de pruebas debe de tener dos factores que conlleven al éxito de la prueba en el menor tiempo posible:

- El primero de estos factores está vigente en los instrumentos que se emplean para la realización de la prueba, debido a que estos deben de estar lo más ajustados posible para poder lograr una mejor medición.
- El segundo está presente en toda la información que se va obteniendo durante el desarrollo del ciclo de vida de una prueba, donde es almacenada, procesada y analizada instantáneamente para poder sacar

Capítulo 2: Características del sistema

conclusiones de forma rápida contribuyendo así a una mejor toma de decisiones por parte del jefe del equipo de pruebas.

El análisis y el procesamiento de la información se cumplen satisfactoriamente, sin embargo la recopilación de información no se realiza de la mejor manera. Intervienen diversos factores que obstaculizan e impiden de manera gradual que el proceso de pruebas se lleve a cabo en el menor tiempo posible, como son los que a continuación se plantean: El Excel, herramienta informática utilizada para la recopilación de la información y otros sistemas de cálculos, permiten la realización de operaciones de cómputo, lo cual brinda apoyo a los expertos en cuanto al almacenamiento de la información al finalizar cada prueba.

Esta herramienta no permite que varias personas trabajen al mismo tiempo sobre los datos guardados, ello trae consigo que cada integrante del equipo de prueba trabaje de manera independiente conservando una copia del archivo, lo cual origina serios problemas en la actualización de estos, provocando incertidumbre acerca de los datos que se poseen, además resulta muy engorroso trabajar con ella cuando se acumula un gran volumen de los datos. Estos programas vistos anteriormente no satisfacen las necesidades de información del personal a cargo de las pruebas, pues los datos demoran mucho en ser procesados y no permiten hacer un análisis más profundo sobre los mismos.

Los dirigentes principales de las pruebas, no conocen lo que está sucediendo en el ensayo hasta que este no ha concluido y hayan sido registrados los datos, así como tampoco es de su conocimiento las insuficiencias y defectos aledaños que hayan sido detectados durante el proceso de prueba, por lo que afecta a la toma de decisiones de los jefes que están al frente de la tarea, impidiéndoles tener una visión más amplia de los resultados obtenidos.

Los resultados de las pruebas se son guardados el papel y son almacenados en cajas y almacenes debido a la gran cantidad de pruebas que se han realizado desde el año 1996 hasta la fecha, lo cual hace difícil el acceso a ellos. No existe uniformidad en los documentos generados por el proceso de pruebas. Continuamente los técnicos, diseñadores e ingenieros de prueba necesitan estudiar la información contenida en los resúmenes de prueba con el objetivo de realizar investigaciones más profundas o realizar comparaciones con datos históricos para emitir mejores criterios acerca de la técnica probada, pero no se

Capítulo 2: Características del sistema

hace posible porque estos tienen una gran extensión y el acceso a los mismos se torna difícil, por lo que se pierde tiempo realizando búsquedas de información en montañas de papeles.

Los integrantes del equipo de prueba requieren intercambiar información de manera constante facilitando la comunicación entre ellos, actividad que se ve limitada al no disponer de un programa que facilite la colaboración entre los especialistas. El proceso de prueba se ve interrumpido frecuentemente por la necesidad de intercambio de información del personal lo cual no permite que las pruebas se realicen con la mejor fluidez.

2.4 Flujo actual del proceso.

La **recepción del vehículo** se lleva a cabo teniendo en cuenta las pruebas a realizar, primero que todo se escoge el método de recepción del medio. En este proceso se comprueba fundamentalmente la correspondencia del prototipo seleccionado con las condiciones técnicas, la documentación de los planos de construcción y la calidad de su fabricación. En otros casos, como la empresa productora tiene derecho a realizar la preparación necesaria para la eliminación de los defectos en la recepción se comprueban las correspondencias de los modelos de vehículos con las condiciones técnicas y con los planos de la documentación constructiva y la calidad de su fabricación.

En la **preparación del vehículo** para la prueba se eliminan los defectos que impiden el trabajo normal del equipo y sus agregados que pudieran reflejarse en la seguridad de la prueba. Se comprueban el estado técnico de las piezas, se le agregan los materiales de combustible y lubricación que tienen que corresponder con los indicados por la fábrica. Se pesa el vehículo con todas las condiciones de carga, así como se escoge el personal que va a intervenir en la realización de los ensayos. Antes de ejecutar se comprueban las condiciones de la vía y que las condiciones meteorológicas sean las adecuadas.

En el último caso está el proceso de **ejecución de las pruebas**, este depende totalmente de la preparación del vehículo ya que muchos de estos nunca llegan a probarse debido a que no están totalmente preparados para las pruebas por sus cualidades físicas.

Al final de cada ensayo del proceso de prueba, se elabora una documentación se registran los datos de la ejecución, entre ellos se encuentran: protocolo (plan escrito y detallado de un experimento científico),

Capítulo 2: Características del sistema

bitácora, informes, actas y otras que se ilustran con esquemas, gráficas, fotografías y videos. Toda esta documentación se organiza en forma de **Expediente** y los datos principales se archivan en papeles y cajas de informes.

2.5 Objetos de automatización.

Para un mejor desarrollo de las actividades antes expuestas se hace necesario la automatización del proceso de prueba, ya que este es muy incómodo para los que en ellos trabajan, además existe cierto nivel de desorganización en los archivos que se manejan con los resultados de las pruebas a los vehículos, falta de comunicación en algunas ocasiones y demoras en la presentación de los resultados.

El proceso a automatizar es el de pruebas técnicas a vehículos, dentro de él, todo lo referente al registro de los medios que serán probados, la gestión de los proyectos en el centro permitiendo la modificación o eliminación de algunos de ellos en cualquier momento y la incorporación de personal a los mismos.

2.6 Información que se maneja.

La información que se maneja en el centro está centrada principalmente en el estado y realización de las pruebas de los vehículos, los cuales desde que son aceptados, comienzan a ser un nuevo proyecto, es decir se les crea el expediente y al final se genera un informe de cada prueba con los resultados obtenidos.

La otra información que se maneja en el centro son los datos necesarios de las empresas que hacen el pedido para realizar las pruebas a alguno de sus medios, la cantidad y tipos de pruebas a efectuarse, así como los datos del personal involucrado en dicho proceso.

2.7 Propuesta del sistema.

Según las investigaciones realizadas hasta el momento en el CID del Transporte de las FAR, y el estudio realizado sobre la bibliografía seleccionada, se concluye que existe un bajo nivel de informatización en el centro debido a que las herramientas que se utilizan para gestionar la información en el proceso de pruebas no permiten que el intercambio de información entre los especialistas sea rápido, fluido y preciso en el momento que se desea. Constantemente los ensayos se ven interrumpidos por la falta de comunicación entre

Indice de Tablas

los integrantes que trae como resultado que se demore el proceso impidiendo que la actividad en ocasiones no se cumpla en el plazo establecido y que el análisis de los resultados no sea tan profundo.

Teniendo en cuenta las insuficiencias identificadas se propone un sistema de gestión de información, “**que no es más que un conjunto de componentes y elementos asociados que interactúan entre sí para lograr un objetivo**”⁵, siendo capaz de transmitir la información de forma fluida entre todos los especialistas que intervienen en los experimentos, así como registrar y controlar el gran volumen de información existente en el centro desde el año 1996. El sistema debe de ser una aplicación web que permita la comunicación desde cualquier punto del país brindándole diferentes funcionalidades que permitan trabajar de forma rápida y agradable sobre la información y erradique las deficiencias antes mencionadas.

La aplicación a desarrollar tiene como objetivo mejorar el proceso de prueba a vehículos del CID del Transporte, para esto cuenta con una serie de funcionalidades entre las que se encuentran: crear proyectos para la realización de las pruebas, modificarlos y eliminarlos, así como cambiar el estado de cada uno de ellos. Otra funcionalidad será adicionar un vehículo, entrar la información referente al mismo entre otras y asignar el personal calificado a dicho proyectos de prueba como la futura realización de las mismas.

2.8 Especificación de los requisitos del software.

Un proyecto no puede ser exitoso sin una descripción detallada, correcta y exhaustiva de los requerimientos pues estos definen lo que debe hacer un sistema y la forma en que debe hacerlo, permiten una comprensión completa de todo el sistema, fundamentalmente cuando se describen cada uno de ellos. Estos se dividen en dos grandes grupos, aquellos que permiten plantear las principales funcionalidades que puede llegar a tener el sistema propuesto y los que permiten que la aplicación funcione correctamente y plantean una estrategia acerca de cómo debe de hacerse la misma. Los requisitos funcionales y no funcionales permitirán satisfacer las necesidades de los usuarios finales guiando así el proceso de desarrollo de software.

⁵ O’Brien J.A, **Sistemas de información gerencial**, McGRAW – HILL INTERAMERICANA, S. A, Bogotá, Colombia, 2001, Pág. 42.

Capítulo 2: Características del sistema

2.8.1 Requisitos funcionales.

Los requisitos funcionales especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física. Por lo general se describen mejor a través del Modelo de Casos de uso y los casos de uso como tal. Por lo tanto los requisitos funcionales especifican el comportamiento de entrada y salida del sistema y surgen de la razón fundamental de la existencia del producto.

Los Requisitos Funcionales para este sistema se presentan a continuación:

1. Gestionar Personal.

1.1 Mostrar Personas.

1.2 Adicionar Persona.

1.3 Modificar Persona.

1.4 Eliminar Persona.

2. Gestionar Nomencladores.

2.1 Países.

2.1.1 Mostrar Países.

2.1.2 Adicionar País.

2.1.3 Modificar País.

2.1.4 Eliminar País.

2.2 Tipo de Vehículo.

2.2.1 Mostrar Vehículos.

2.2.2 Adicionar Vehículos.

2.2.3 Modificar Vehículos.

Capítulo 2: Características del sistema

2.2.4 Eliminar Vehículos.

2.3 Marca.

2.3.1 Mostrar Marcas.

2.3.2 Adicionar Marcas.

2.3.3 Modificar Marcas.

2.3.4 Eliminar Marcas.

2.4 Modelo.

2.4.1 Mostrar Modelos.

2.4.2 Adicionar Modelo.

2.4.3 Modificar Modelo.

2.4.4 Eliminar Modelo.

3. Gestionar Proyectos.

3.1 Mostrar Proyectos.

3.2 Adicionar Proyecto.

3.3 Modificar Proyecto.

4. Asignar Personal a Proyecto.

4.1 Mostrar Personal Asignado por Proyecto.

4.2 Asignar Persona.

4.3 Eliminar Persona.

5. Autenticar Usuario.

Capítulo 2: Características del sistema

2.8.2 Requisitos no funcionales.

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener, forman una parte significativa de la especificación. Tienen la característica de marcar la diferencia, pues una vez comprobado que el producto cumple con lo requerido por el cliente, las propiedades no funcionales; como cuán usable, seguro, conveniente y agradable, distinguen a un producto bien aceptado de uno con poca aceptación.

Para un correcto funcionamiento el sistema debe de tener como características principales para su funcionamiento los siguientes requisitos no funcionales:

➤ Usabilidad.

El usuario entra la menor cantidad de información ya que casi todo es a través de clic.

La interfaz debe ser agradable a la vista del usuario.

➤ Portabilidad.

La aplicación es compatible con los navegadores Mozilla Firefox e Internet Explorer.

Sistema multiplataforma debido a que se puede usar en Windows y todos los sistemas operativos Linux.

➤ Hardware.

Servidor:

- ❖ Procesador Pentium 4, de 4 a 1.5GHz.
- ❖ 1GB de memoria RAM o superior.
- ❖ 80GB de disco duro o superior.
- ❖ Tarjeta de Red.

Capítulo 2: Características del sistema

- ❖ Tarjeta de Video.

Ciente:

- ❖ Procesador Pentium III a 200 MHz o superior.
- ❖ 256MB de memoria RAM o superior.
- ❖ 6 GB de disco duro o superior.
- ❖ Tarjeta de Red.
- ❖ Tarjeta de Video.

➤ **Seguridad.**

Las políticas de seguridad son soportadas por el UCID. Las políticas de seguridad, control y acceso de usuarios por roles es asegurado por el servicio desarrollado para aplicaciones web en el UCID. La seguridad del sistema de pruebas a vehículos es implementada por el módulo de administración que es el encargado de gestionar los aspectos fundamentales para el funcionamiento del mismo.

➤ **Integridad de los Datos.**

El sistema debe de realizar salvadas constantes de la información manteniendo así la integridad de los datos.

La información entrada por el usuario es validada por el sistema antes de ser guardada.

Capítulo 2: Características del sistema

2.9 Conclusiones parciales.

En este capítulo se analizaron las tendencias tecnológicas actuales que se emplearon para el desarrollo de la investigación que se encuentran dentro del marco de trabajo de la Unidad de Compatibilización, Integración y Desarrollo para la Defensa (UCID) dándose una breve descripción de cada una de ellas así como de las principales herramientas que sirvieron de apoyo para poder lograr los objetivos que se persiguen con esta investigación.

También se abordó las principales características del sistema propuesto, el flujo principal de los procesos de prueba así como el que se debe de automatizar y demás detalles como el tipo de información que se maneja en la entidad. Además de ello se listan todos los requisitos del software ya sean funcionales o no funcionales.

Capítulo 3: Implementación y pruebas

Capítulo 3: Implementación y Prueba.

3.1 Introducción.

El flujo de trabajo de Implementación tiene como objetivo definir la organización del código teniendo en cuenta los subsistemas de implementación, la implementación de los elementos de diseño en términos de ficheros fuentes, binarios y ejecutables, para poder integrar los diferentes componentes de desarrolladores o equipos y generar un ejecutable entregable o producto final. En este capítulo se presenta el artefacto diagrama de componentes, el cual muestra las dependencias entre dichos componentes. Además de la etapa de prueba, que se le realiza al software con el objetivo de asegurar completitud, correctitud, calidad, entre otros factores de gran importancia.

Además de que se lleva a cabo el proceso de pruebas siguiendo el método de caja negra. También se describe la arquitectura seleccionada y el papel que juegan cada una de sus partes en el desarrollo de esta solución.

3.2 Arquitectura utilizada.

La arquitectura utilizada para darle solución a nuestro problema es la definida en el marco de trabajo del centro, es decir la arquitectura en capa o por niveles, la misma implementa el patrón modelo – vista – controlador y el patrón ORM para la conexión a la base de datos.

La arquitectura 3 capas o programación 3 capas consiste literalmente en separar un proyecto en Capa de Presentación, Capa de Negocio y Capa de Datos. Esto permite distribuir el trabajo de creación de una aplicación por niveles; de este modo, cada grupo de trabajo está totalmente abstraído del resto de niveles, de forma que basta con conocer la API que existe entre niveles.

Capítulo 3: Implementación y pruebas

3.2.1 Representación gráfica de la arquitectura en 3 capas.

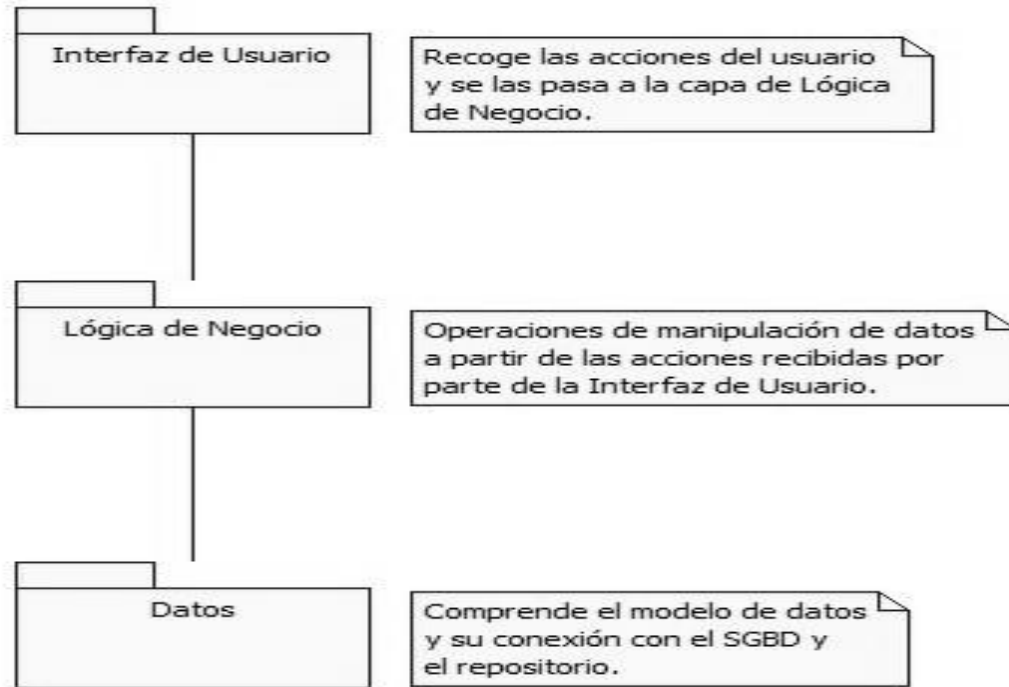


Fig. 1 Representación gráfica de la Arquitectura 3 Capas.

- ❖ **Capa de presentación:** Esta capa es la que visualiza el usuario, le presenta el sistema, le comunica la información y captura la información del mismo en un mínimo de proceso. Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica y debe tener la característica de ser "amigable" o entendible para el usuario generalmente, se presentan como formularios.
- ❖ **Capa de negocio:** En esta capa es donde, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio (e incluso de lógica del negocio) porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de

Capítulo 3: Implementación y pruebas

presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él.

❖ **Capa de datos:** Es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

3.2.2 Ventajas de esta Arquitectura.

- ✓ El desarrollo se puede llevar a cabo en varios niveles.
- ✓ Desarrollos paralelos (en cada capa).
- ✓ Aplicaciones más robustas debido al encapsulamiento.
- ✓ En caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar el código mezclado.
- ✓ Mantenimiento y soporte más sencillo (es más sencillo cambiar un componente que modificar una aplicación monolítica).
- ✓ Mayor flexibilidad (se pueden añadir nuevos módulos para dotar al sistema de nueva funcionalidad).
- ✓ Alta escalabilidad. La principal ventaja de una aplicación distribuida bien diseñada es su buen escalador, es decir, que puede manejar muchas peticiones con el mismo rendimiento simplemente añadiendo más hardware. El crecimiento es casi lineal y no es necesario añadir más código para conseguir esta escalabilidad.

3.3 Patrones utilizados.

3.3.1. Modelo – vista –controlador.

Elementos del patrón:

- ✓ **Modelo:** datos y reglas del negocio.

Indice de Tablas

- ✓ **Vista:** muestra la información del modelo al usuario.
- ✓ **Controlador:** gestiona las entradas del usuario.

Un modelo puede tener diversas vistas, cada una con su correspondiente controlador. Un ejemplo es la información de una BD, que se puede presentar de diversas formas: diagrama de tarta, de barras, tabular, etc. y otros.

El **modelo** es responsable de:

- ✓ Acceder a la capa de almacenamiento de datos.
- ✓ Define las reglas del negocio (la funcionalidad del sistema).
- ✓ Llevar un registro de las vistas y controladores del sistema.

El **controlador** es responsable de:

- ✓ Recibir los eventos de entrada.
- ✓ Contiene reglas de gestión de eventos. Estas acciones pueden suponer peticiones al modelo o a las vistas.
- ✓ Carga el modelo, las librerías, helpers, plugins, y todos los demás recursos necesarios para satisfacer nuestra petición.

Las **vistas** son responsables de:

- ✓ Recibir los datos del controlador y mostrárselos al usuario.

Capítulo 3: Implementación y pruebas

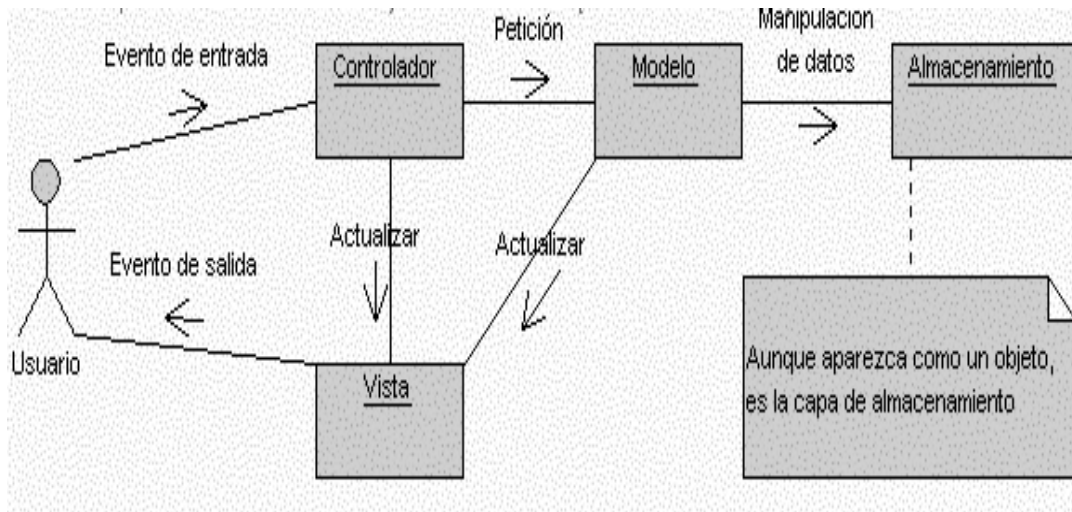


Fig. 2 Patrón modelo – vista – controlador⁶.

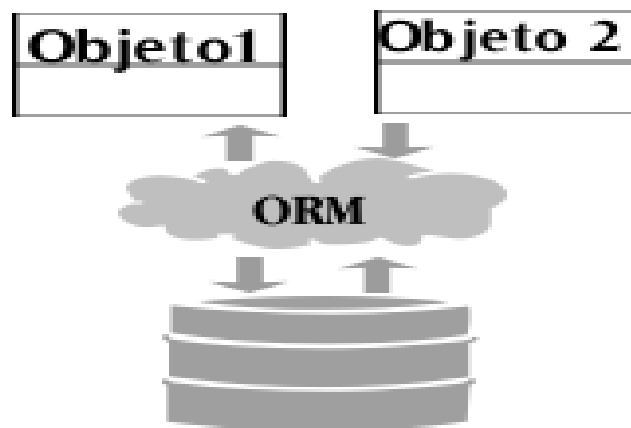


Fig. 3 Patrón ORM.

⁶ Proactiva-Calidad. Tecnología orientada a procesos de negocio.2002.

<<http://www.proactiva-calidad.com/java/patrones/index.html>> [consultado: 9 de mayo del 2010]

Capítulo 3: Implementación y pruebas

El ORM es un componente de software que permite trabajar con los datos persistidos como si ellos fueran parte de una base de datos orientada a objetos (en este caso virtual). Debido a que lo estándar es trabajar con BD relacionales, se deben realizar operaciones que permitan transformar un registro en objeto y viceversa. A esta funcionalidad se la llama Mapeo objeto-relacional (ORM).

Object Space: es el objeto principal que nos permite interactuar con los datos. Nos brinda facilidades para instanciar objetos, salvar objetos, eliminar objetos, crear transacciones, cargar listas de objetos, etc de una forma muy intuitiva.

Utilizando herramientas ORM se accederá el proceso de desarrollo de software. También se obtiene una independencia del motor de base de datos ya que el desarrollador no escribe SQL directamente sino un pseudo lenguaje que es traducido al SQL específico de la base de datos configurada.

El código queda mucho más limpio y menos código es necesario lo cual minimiza la probabilidad de errores.

El paradigma relacional se basa en principios matemáticos con tablas y relaciones entre estas y el paradigma de OO se basa en principios de desarrollo de software como clases, objetos y herencia, estos dos paradigmas no conviven fácilmente. Esto se ve claramente en la manera de acceder a los datos. Mientras en el mundo OO los objetos se relacionan a través de propiedades, en el mundo relacional las tablas se unen a través de llaves primarias. Este problema es conocido como "Impedance Mismatch"(una traducción cercana sería "desajuste de relaciones") y es conocido hace ya un buen tiempo. Las herramientas ORM tratan de solucionar este problema.

3.4 Diagrama de componentes.

Un Diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean éstos componentes de código fuente, binarios o ejecutables. Se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo. Los elementos de modelado dentro de un Diagrama de componentes serán componentes y paquetes.

Capítulo 3: Implementación y pruebas

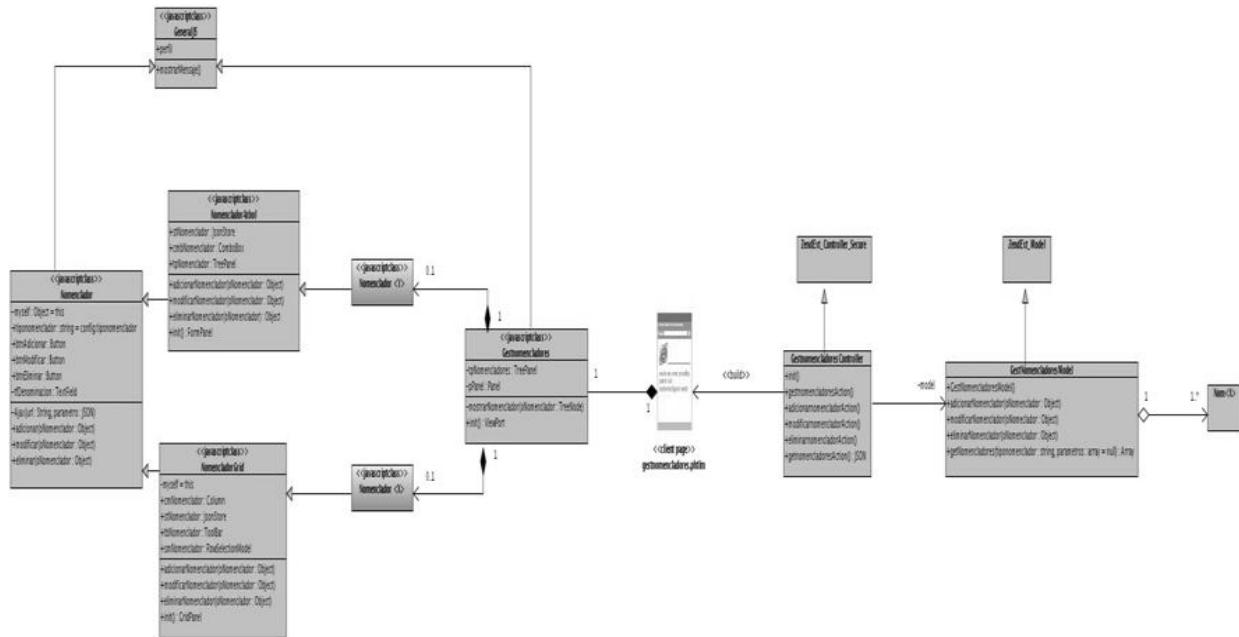


Fig. 4 Diagrama de componentes.

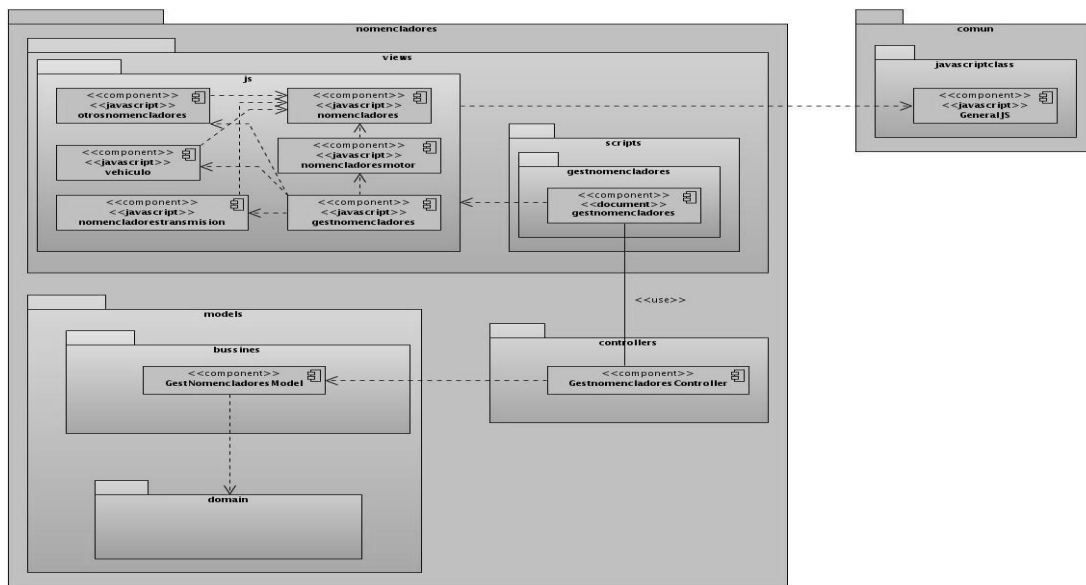


Fig. 5. Diagrama de componente de los nomencladores.

Capítulo 3: Implementación y pruebas

3.4.2 Descripción del diagrama de componente de los nomencladores.

El componente nomencladores está integrado por 3 paquetes de componentes principales que a continuación se mencionan y se explica la función de cada uno de ellos:

1 - paquete views: agrupa todos los componentes cuya responsabilidad reside en construir las interfaces para representar los datos de los diferentes nomencladores definidos en el sistema. Este paquete está compuesto por 5 componentes javascript que agrupan las clases interfaces de los diferentes nomencladores de sistema, además de un componente gestnomencladores que se encarga de gestionar la interfaz que se debe mostrar en cada momento.

2 - paquete controllers: En este paquete se encuentra el componente GestnomencladoresController cuya responsabilidad está en servir de puente entre los componentes del paquete views y los componentes que gestionan las reglas del negocio que implementa la solución.

3 - paquete models: Agrupa los componentes que implementan las reglas del negocio y el dominio el componente GestNomencladoresModel implementa las reglas definidas para la gestión de los nomencladores. En paquete domain agrupa los componentes del dominio de los nomencladores. Cada componente implementa una abstracción de una tabla de la base de datos.

3.5 Matriz de integración de componentes.

La matriz de integración de componentes contiene todos los componentes definidos en el subsistema, de forma matricial, y en las intercepciones se especificarán los servicios que consume el componente en la vertical del horizontal. Existen dos matrices de integración de componentes, Interna y Externa.

Interna: se especifica la integración entre los componentes internos del sistema.

Externa: se especifica la integración entre los componentes externos del sistema.

Capítulo 3: Implementación y pruebas

3.5.1 Tabla de integración de componentes internos.

Componentes internos	Componente que brinda	Componente que brinda	Componente que brinda
	Nomencladores	Motor de Pruebas	Proyecto
Proyecto	Adicionar, modificar, eliminar.	Modelo y ejecución de pruebas.	Ninguno
Nomencladores	Ninguno	Ninguno	Adicionar, modificar, eliminar.
Motor de prueba	Ninguno	Ninguno	Modelo y ejecución de pruebas.

Tabla 1. Tabla de integración de componentes internos.

Capítulo 3: Implementación y pruebas

3.5.2 Tabla de integración de componentes externos.

Componentes internos	Componente que brinda	Componente que brinda
	Registrar Persona	Proyecto
Proyecto	Adicionar, modificar, eliminar.	Ninguno
Registrar persona	Ninguno	Adicionar, modificar, eliminar.

Tabla 2. Tabla de integración de componentes externos.

3.6 Pruebas.

La prueba del software 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. El objetivo de la etapa de pruebas es garantizar la calidad del producto desarrollado. Es una actividad en la cual un sistema o componente es ejecutado bajo condiciones o requerimientos específicos, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente.

Después de un análisis de los diferentes métodos de pruebas que existen y la función de cada una de ellas hemos llegado a la conclusión de que es más factible aplicar las pruebas de Caja Negra por lo que fue el método que se escogió en el proyecto para comprobar la validez en las respuestas del programa ante las acciones del usuario y la calidad de las salidas en dependencia de las entradas. Para probar la validez de la solución propuesta en la implementación del sistema se decidió probar los elementos fundamentales de la interfaz de usuario mediante casos de prueba de Caja Negra así como la validez de las salidas del sistema en dependencia de las entradas del usuario.

Capítulo 3: Implementación y pruebas

3.6.1 Diseño de Casos de prueba.

El objetivo fundamental del diseño de casos de prueba es conseguir un conjunto de pruebas que tengan la mayor probabilidad de encontrar los defectos del software.

Un caso de prueba es un conjunto de entradas de pruebas, condiciones de ejecución y resultados esperados desarrollados para verificar una función esperada.

3.6.2 Descripción del Caso de prueba Gestionar nomencladores.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
Adicionar nomenclador.	Se inserta un nuevo nomenclador al sistema.	EP 1.1: Adicionar nomenclador correctamente.	<ul style="list-style-type: none">• Se selecciona en la interfaz el botón Adicionar nomenclador.• El sistema muestra los una interfaz donde permite se muestran los datos a llenar del nuevo nomenclador a adicionar.• El usuario llena los datos correctamente y presiona el botón Aceptar.• El sistema muestra un mensaje indicando que la operación se realizó de manera correcta.

Tabla 4. Descripción del caso de prueba “Gestionar nomenclador.”

Capítulo 3: Implementación y pruebas

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
Adicionar nomenclador.	Se inserta un nuevo nomenclador al sistema.	EP 1.2: Adicionar nomenclador incorrectamente	<ul style="list-style-type: none">• Se selecciona en la interfaz el botón Adicionar nomenclador.• El sistema muestra una interfaz donde se muestran los datos a llenar del nuevo nomenclador a adicionar.• El usuario llena los campos con datos incorrectos y/o deja campos obligatorios vacíos.• El sistema marca los datos incorrectos y/o vacíos.• El usuario presiona el botón Aceptar.• El sistema no permite insertar.
		EP 1.3: Cancelar operación	<ul style="list-style-type: none">• Se selecciona el botón Adicionar nomenclador.• El sistema muestra los campos a llenar.• El usuario llena o no los campos y presiona el botón Cancelar.

Capítulo 3: Implementación y pruebas

Id del escenario	Escenario	Descripción	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar nomenclador correctamente.	Nombre del nomenclador.	Se espera a que el sistema muestre un mensaje confirmando que el nomenclador se insertó correctamente.	El sistema muestra un mensaje indicando que el nomenclador se insertó correctamente.
EP 1.2	Adicionar nomenclador incorrectamente.	Nombre del nomenclador.	Se espera a que el sistema marque a que existen errores.	El sistema marca la denominación demostrando que existen errores.
EP 1.3	Cancelar operación.	nulo	El debe mostrar "Operación cancelada".	El sistema marca que existen errores.

Capítulo 3: Implementación y pruebas

3.6.3 Descripción del caso de prueba “Gestionar proyecto”.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
Adicionar nuevo proyecto de prueba.	Se inserta un nuevo proyecto de prueba al sistema.	EP 1.1: Adicionar nuevo proyecto de prueba correctamente.	<ul style="list-style-type: none">• Se selecciona en la interfaz el botón Adicionar nuevo proyecto de prueba.• El sistema muestra una interfaz donde se muestran los datos a llenar del nuevo proyecto a adicionar.• El usuario llena los datos correctamente y presiona el botón Aceptar.• El sistema muestra un mensaje indicando que la operación se realizó de manera correcta.

Tabla 3. Descripción del caso de prueba “Gestionar proyecto”.

Capítulo 3: Implementación y pruebas

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
Adicionar nuevo proyecto de prueba.	Se inserta un nuevo proyecto de prueba al sistema.	EP 1.2: Adicionar proyecto de prueba incorrectamente.	<ul style="list-style-type: none">• Se selecciona en la interfaz el botón Adicionar nuevo proyecto de prueba.• El sistema muestra los una interfaz se muestran los datos a llenar del nuevo proyecto de prueba a adicionar.• El usuario llena los campos con datos incorrectos y/o deja campos obligatorios vacíos.• El sistema marca los datos incorrectos y/o vacíos.• El usuario presiona el botón Aceptar.
		EP 1.3: Cancelar operación	<ul style="list-style-type: none">• Se selecciona en la interfaz el botón Adicionar nuevo proyecto de prueba.• El sistema muestra los campos a llenar.• El usuario llena o no los campos y presiona el botón Cancelar.

Capítulo 3: Implementación y pruebas

Id del escenario	Escenario	Descripción	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar nuevo proyecto de prueba	Descripción del nuevo proyecto.	Se espera a que el sistema muestre un mensaje confirmando que el nuevo proyecto de prueba se insertó correctamente.	El sistema muestra un mensaje indicando que el nuevo proyecto de prueba se insertó correctamente.
EP 1.2	Adicionar nuevo proyecto de prueba	Descripción del nuevo proyecto.	Se espera a que el sistema marque que existen errores.	El sistema marca la descripción demostrando que existen errores.
EP 1.3	Cancelar nuevo proyecto de prueba	nulo	El debe mostrar "Operación cancelada".	El sistema marca que existen errores".

Capítulo 3: Implementación y pruebas

3.7 Resumen de las pruebas.

Se realizaron un total de 2 casos de prueba en los que los resultados en cada uno de los escenarios que los comprende fue el esperado por el desarrollador, es decir no se tuvo no conformidades, por lo que se demuestra el cuidado de los programadores a la hora del desarrollo del sistema.

3.8 Conclusiones parciales.

De este modo se da por concluido el presente capítulo donde se describió la arquitectura utilizada para el desarrollo de esta solución, las ventajas de la misma y por último se lleva a cabo el proceso de pruebas para verificar si el sistema no presenta deficiencias utilizando el método de caja negra.

Conclusiones

Conclusiones del trabajo

Con el desarrollo de la investigación se lograron cumplir los objetivos propuestos al principio de la misma, se desarrolló la implementación del Sistema de Gestión de Información de Pruebas Técnicas a Vehículos del CID del Transporte, logrando así dotar al mismo de flexibilidad ante los diferentes medios a probar. A continuación se mencionan los resultados obtenidos durante la investigación:

- Se realizó el estudio del arte el cual nos permitió conocer acerca de los diferentes softwares existentes en el mundo que permiten la realización exitosa de los procesos de prueba.
- Se investigaron los principales centros y posibles soluciones que podían darle solución al problema planteado, aunque se llegó a la conclusión de que ninguno cumplía las condiciones necesarias.
- Se justificaron las herramientas, metodologías y tecnologías que se utilizaron en el desarrollo del trabajo las cuales permitieron la implementación de un sistema que cumpla con los requisitos necesarios para un eficiente proceso de pruebas.
- Se describieron los requisitos funcionales y no funcionales del sistema.
- Se realizó la implementación del sistema atendiendo a la propuesta arquitectónica y se diseñaron los modelos de pruebas, los cuales permitieron verificar la calidad de la solución.

Recomendaciones

Recomendaciones

Teniendo en cuenta los resultados obtenidos en esta investigación y basado en la experiencia adquirida y sin dejar de tener en cuenta que este trabajo solo es la primera versión de este sistema y que pueden surgir nuevas necesidades del cliente se proponen las siguientes recomendaciones:

- Continuar la investigación de nuevas funcionalidades y capacidades para agregar al sistema permitiendo un proceso más amplio.
- Trabajar en mejoras para el diseño del sistema de forma tal que se logren interfaces menos cargadas de datos.
- Que se desarrollen nuevas versiones del software y se le realicen cambios con el cursar del tiempo.
- Implementar el módulo de gestionar reporte para tener una versión digital de los resultados de las pruebas.

Bibliografía

Bibliografía

1. Reinier Sayu Belon. Sistema de Dirección de las Transportaciones Militares. Modulo Pruebas Técnicas a los vehículos .Universidad de las Ciencias Informáticas. La Habana.2009.122 pág.
2. Manual de Prueba de Carros Militares. Fundamentos de las Pruebas de Vehículos. Pág. 13.
3. Manual de Prueba de Carros Militares. Fundamentos de las Pruebas de Vehículos. Pág. 19.
4. Ollet Nerey, M. (2008). Manual de Pruebas de Carros Militares. Fundamentos de las Pruebas de Vehículos. Ciudad de la Habana.
5. Ollet Nerey, M. (2008). Programa de Pruebas. Ciudad de la Habana.
6. Programa de Pruebas. Manuel Ollet Nerey. La Habana. 2008. Pág. 3.
7. Ollet Nerey, M. (2008). Protocolo de Pruebas. Ciudad de la Habana.
8. DCM-TRANS, C. d. (2008). Expediente de Pruebas del Vehículo. Ciudad de la Habana, Cuba.
9. Ingeniero Miguel Cárdenas Garnier. Evaluación y Prueba de los Automóviles. Pág. 8.
10. Resumen de las Informaciones sobre el Polígono de Aberdeen. Centro de Pruebas de los Estados Unidos. Pág. 5.

Referencia bibliográfica

Referencias bibliográficas.

1. Centro Principal de Automatización MINFAR. Defensa Nacional. 2003. Disponible en: <http://www.cubagob.cu/otras_info/minfar/industria/industria_militar.htm>. [consultada: 10 de enero del 2010].
2. Unidad Editorial, S.A. Diccionarios del mundo. Disponible en: <<http://www.elmundo.es/diccionarios/>>. [consultado: 15 de enero del 2010].
3. Ventajas de PosgretSQL. Disponible en: <http://soporte.tiendalinux.com/portal/Portfolio/postgresql_ventajas.html>.
4. Catalani, Exequiel. "Arquitectura Modelo Vista Controlador". Disponible en: <<http://exequielc.wordpress.com/2007/08/20/arquitectura-modelovistacontrolador/>>.
5. Bienvenido a la doctrina del proyecto. Disponible en: <<http://translate.google.com.cu/translate?hl=es&sl=en&u=http://www.doctrine-project.org/&ei=ksKAS92FONKW8AbT6qX1BA&sa=X&oi=translate&ct=result&resnum=1&ved=0CBYQ7gEwAA&prev=/search%3Fq%3Ddoctrine%26hl%3Des>>.
6. ¿Qué es Doctrine ORM? Disponible en: <<http://www.tecnoretas.com/programacion/que-es-doctrine-orm/>>.
7. Extjs, lo bueno, lo malo y lo feo. Disponible en: <<http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>>.
8. Ext JS Frameworks. Disponible en: <<http://www.quizspot.com/2009/01/ext-js-framework/>>.

Abreviaturas

Abreviaturas.

CID: Centro de Investigación y Desarrollo de las Fuerzas Armadas Revolucionarias.

UIM: Unión de Industrias Militares.

DQL: Doctrine Query Language.

ORM: Object Relation Mapper.

IDE: Entornos de desarrollo integrados.

SWT: Standard Widget Toolkit.

SARD o **ADACS:** Sistema de información integrado.

SIOFPA o **ATIRS:** Sistema de informes de las ocurrencias de Pruebas.

Glosario de términos

Glosario de términos.

Artefacto: Pieza de información tangible que es creada, modificada y usada por los trabajadores al realizar actividades; representa un área de responsabilidad, y es candidata a ser tenida en cuenta para el control de la configuración. Un artefacto puede ser un modelo, un elemento de un modelo, o un documento.

Software: Es la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo y que “un producto de software es un producto diseñado para un usuario”.

Gestión de información: es un conjunto de componentes y elementos asociados con la información que interaccionan entre sí para lograr un objetivo: facilitar o recuperar información.

Estado del arte: Hace referencia al nivel más alto de desarrollo conseguido en un momento determinado sobre cualquier temática.

Proceso de prueba: es la realización de ensayos a los diferentes vehículos, además incluye el análisis de toda la información obtenida del proceso.

Propuesta arquitectónica: es un diseño propuesto en base al cual se va a trabajar. Conjunto de ideas para un diseño a realizar.

Prototipos: se refiere a cualquier tipo de máquina en pruebas, o un objeto diseñado para una demostración de cualquier tipo.

Anexos

Interfaz gráfica del Adicionar proyecto de prueba.

The screenshot shows a software application window titled "Gestionar proyecto". At the top, there is a navigation bar with "Inicio" and "Gestionar...". Below this, a toolbar contains "Adicionar", "Modificar", "Eliminar", and "Registrar datos del motor". A table displays project data:

No.	Nombre proyecto	Entidad cliente	Estado del proyecto
1	Este	UNIVERSIDAD CIENCIAS INFORMATICAS	PROPUESTO

A modal dialog titled "Adicionar proyecto de prueba" is open, containing the following fields:

- Nombre proyecto:
- Entidad cliente:
- Estado proyecto:
- Descripción:
- Datos iniciales del vehículo:
 - Chasis:
 - Número serie:
 - Marca-Modelo vehículo:
 - País fabricante:
 - Peso(Ton):

Buttons for "Cancelar" and "Aceptar" are located at the bottom right of the dialog. The main window footer shows "Página 1 de 1".

Figura Interfaz gráfica del Adicionar Proyecto de Prueba.

Anexos

Interfaz gráfica del Modificar proyecto de prueba.

The screenshot displays a web-based application interface for managing test projects. At the top, there is a navigation bar with 'Inicio' and 'Gestionar...' buttons. Below this, the main window is titled 'Gestionar proyecto' and features a toolbar with 'Adicionar', 'Modificar', 'Eliminar', and 'Registrar datos del motor' options. A table lists the current project: 'Este' (No. 1), 'UNIVERSIDAD CIENCIAS INFORMATICAS' (Entidad cliente), and 'PROPUESTO' (Estado del proyecto). A modal dialog titled 'Modificar proyecto de prueba' is open, allowing for editing. It includes fields for 'Nombre proyecto' (set to 'Este'), 'Entidad cliente' (set to 'UCI'), and 'Estado proyecto' (set to 'PROPUESTO'). A text area for 'Descripción' contains 'lalaalaalaa'. A section titled 'Datos iniciales del vehículo' contains fields for 'Chasis' (mmm), 'Número serie' (123456), 'Marca-Modelo vehículo' (LADA--2107), 'País fabricante' (COREA), and 'Peso(Ton)' (50). 'Cancelar' and 'Aceptar' buttons are located at the bottom of the dialog. The bottom of the main window shows a pagination control: 'Página 1 de 1'.

Figura Interfaz gráfica del Modificar Proyecto de Prueba.

Anexos

Interfaz gráfica del Eliminar proyecto de prueba.

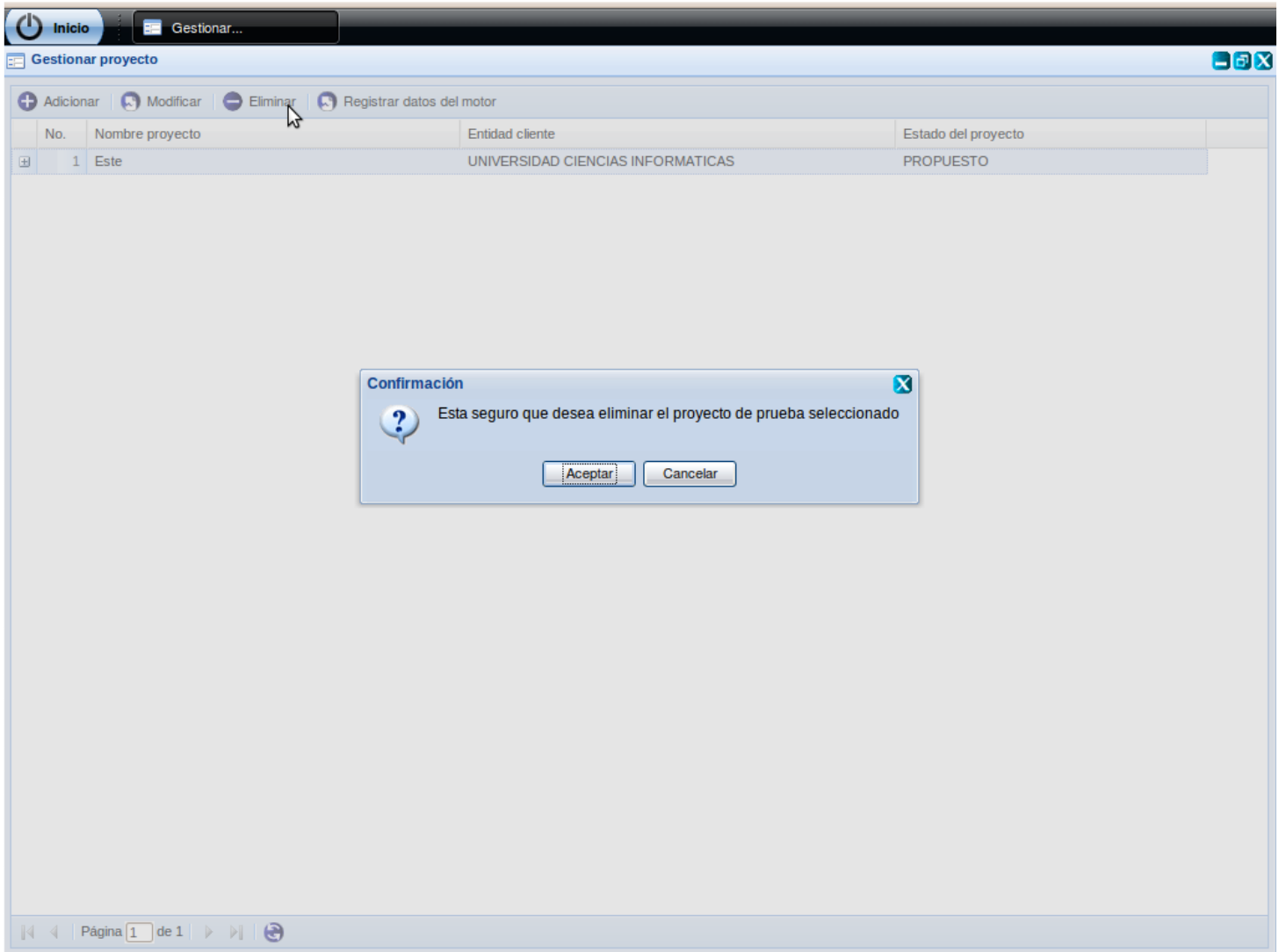


Figura Interfaz gráfica del Eliminar Proyecto de Prueba.

Anexos

Interfaz gráfica del gestionar proyecto de prueba.

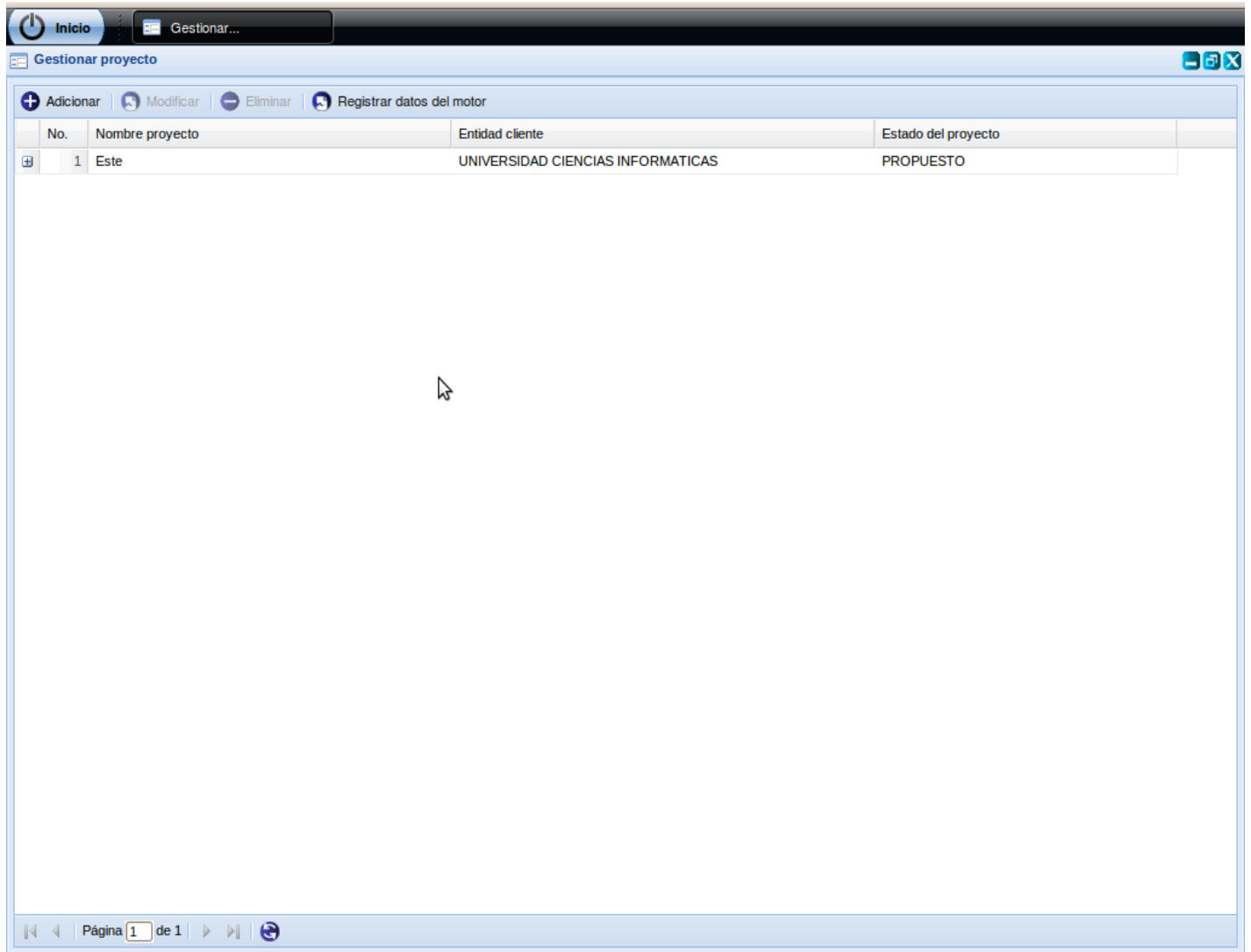


Figura Interfaz gráfica del Gestionar Proyecto de Prueba.

Anexos

Interfaz gráfica del Gestionar Nomencladores.

The screenshot displays the 'Gestionar Nomencladores' application interface. On the left, a tree view shows the following categories and sub-items:

- Motor**
 - Marca del motor
 - Modelo del motor
 - Tipo de aspiración
 - Tipo de lubricación
 - Disposición de cilindros
- Sistema de distribución**
 - Tipo de accionamiento
 - Tipo de árbol de leva
 - Tipo de mando
 - Tipo de balancines
- Sistema de refrigeración**
 - Tipo de sistema de refrigeración
 - Tipo de radiador
 - Tipo de ventilación
 - Tipo de ventilador
 - Tipo de termostato
 - Tipo de circuito de refrigeración
- Sistema de alimentación**
 - Tipo de bomba
 - Tipo de filtro
 - Tipo de inyección
- Sistema de transmisión**
- Vehículo**
 - Marca de vehículo
 - Modelo de vehículo
- Otros nomencladores**
 - Pais fabricante
 - Estado
 - Entidad
 - Rol

The main window, titled 'Modelo del motor', features a search bar with 'ZIL' selected and 'MM23' entered. Below the search bar are buttons for '+ Adicionar', 'Modificar', and '- Eliminar'. The table below contains the following data:

No.	Marca del Motor	Modelo del Motor
1	VOLVO	131
2	VOLGA	JJJ
3	ZIL	MM23
4	ZIL	2107
5	MERCEDES BENZ	23565

Figura Interfaz gráfica del Gestionar Nomencladores.