

Universidad de las Ciencias Informáticas

Facultad 3



“Desarrollo del módulo Control de combustible del Sistema de Control de Flota y Mantenimiento de la Dirección de Transporte de la Universidad de las Ciencias Informáticas.”

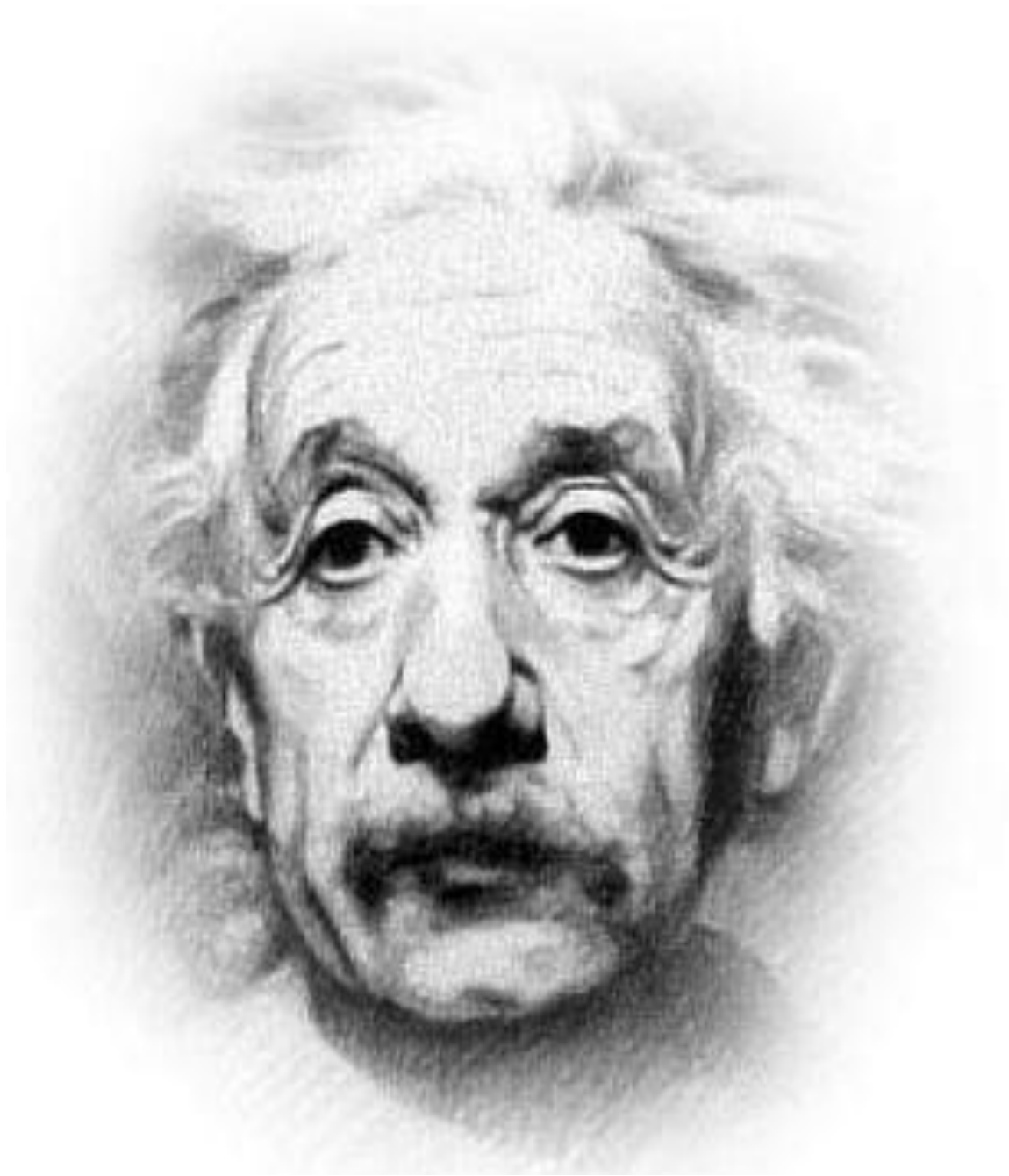
Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor(es): Arellys Saavedra Ayrado

Rolando Rodríguez Lastra

Tutora: Ing. Mailyn Hernández Gómez

Junio de 2013



*"Nunca consideres el estudio como una obligación sino como una
oportunidad para penetrar en el bello mundo del saber"*

Albert Einstein

Declaramos que somos los autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Arelys Saavedra Ayrado

Firma del Autor

Rolando Rodríguez Lastra

Firma del Autor

Ing. Mailyn Hernández Gómez

Firma del Tutor

Tutor: Ing. Mailyn Hernández Gómez.

Correo electrónico: mhgomez@uci.cu

Título de graduado: Ingeniero en Ciencias Informáticas.

Ingeniero en Ciencias Informáticas de la UCI desde Junio de 2009. Instructor. Actualmente se desempeña como Analista Principal del proyecto Mantenimiento vehicular.

Arelys:

Agradecer a mi mamá por todos los sacrificios que ha hecho por mí, por haber dedicado toda su vida a mi felicidad y por estar siempre presente cuando la he necesitado. A mi papá por quererme, apoyarme y enseñarme que no hay sueños imposibles, por siempre darme esas fuerzas de seguir adelante sin temor a nada. A mis hermanos por siempre ayudarme en lo que he necesitado y confiar en mí en todo momento. Agradecer sin lugar a duda a mi novio Gregorio por su dedicación, por haberme hecho ver la vida de otro modo, por siempre creer en mí, por darme su cariño y amor, por acompañarme en los malos y buenos tiempos, por ser más que novio, un gran amigo. A los profesores que contribuyeron en mi formación como profesional, agradezco de forma especial a Alain, por ayudarme incondicionalmente cada vez que lo necesitaba y por ser mi más grande guía, no solo como profesor sino como un excelente amigo. A la tutora Mailyn mil gracias, que supo guiarnos por el camino correcto y brindarnos su apoyo total en la realización de este trabajo de diploma, y a mi compañero de tesis Rolando, que sin su ayuda no se hubiera logrado el desarrollo de este trabajo.

En general a todos los que de una forma u otra me han ayudado a superarme durante estos 5 años.

Rolando:

Durante nuestra vida nos encontramos con situaciones tan difíciles que pensamos que se nos viene el mundo encima y que no podemos lograr nada, lo más importante en ocasiones es mirar para el lado y darse cuenta de que existen personas con las que se puede contar y que estarían encantados ayudándote, es por ello que hoy quisiera agradecer a todos los profesores y personas que han influido en mi formación durante estos 5 años de carrera, a todos los integrantes del tribunal y al oponente que han sabido guiarnos en la realización de este trabajo, a nuestra tutora, nada que hablar, no creo que podamos encontrar una mejor, admiro de usted su disposición ante cualquier situación que se nos presentó, agradezco además a

profesores del proyecto como Erich, Dasiel y Pedro, de ellos he aprendido mucho, fueron un gran apoyo para aclarar nuestras dudas, doy mi más sincero agradecimiento para mi compañera de tesis Arelys y al "gran team" que formamos juntos, de no ser por ella hoy no estaríamos aquí, que llegue desde aquí mi felicitación para su familia. Además quisiera agradecer a mis compañeros de aula por estos fenomenales 5 años y todos los momentos ricos que pasamos, en especial al dream team de fútbol, a mis amigos Pavel, Luisi, Tony, Acebal, el Loren, el fa, Rafiki y a mis compañeras de aula Kiri, Evelin y Lianet, quisiera agradecer a "la yuca" que demostró ser para mí un excelente amigo, una persona con la cual se puede contar para cualquier cosa y admiro de él su deseo de aprender cada día algo nuevo, desde mi corazón brota un inmenso "Gracias por existir para la niña más linda de este mundo", mi cosita linda, Meylin, gracias por darme los mejores 29 meses de mi vida, te quiero mucho, y esto también es para ti, gracias además a su familia por acogerme como un miembro más, quisiera hacer un especial agradecimiento a mi familia.

Arelys:

Dedico el fruto de mis horas de estudio e investigación a mis padres Alina y Alberto por ser las personas más importantes de mi vida, por ayudarme, guiarme y demostrarme que no hay nada imposible en la vida, a mis hermanos y a toda mi familia en general.

Rolando:

Quiero dedicar este trabajo a mi familia en general, a mis abuelos Gina, Daniel, Rosalina y Rolando, en especial a mi abuelo Daniel que es un ejemplo para mí en todo lo que hago, sé que hoy estarás muy orgulloso de mí, a mi mamita linda que es la mejor madre del mundo, eres un ejemplo de sacrificio y deseo de "echar pa adelante", yo te dedico este trabajo con todo el amor que siento por ti, te quiero mucho, mami, a mi papa que siempre ha estado ahí cuando lo he necesitado y me enseñó a ser el hombre que soy hoy, te quiero mucho también papi, quiero dedicar este trabajo a mis hermanos que aunque no pudieron estar aquí hoy, los llevo en mi corazón a donde quiera que voy, desde aquí un los quiero mucho mis mellis, y que les sirva de ejemplo la realización de esta tesis para su futuro, tanto a ellos como a mis primas Baby y Adry, a quienes junto a mi tía Ginita y mi tío Juanito también dedico mi tesis, ustedes que supieron estar en estos 5 años que me encontraba lejos de mi casa y supieron ser mi familia chiquita, que me brindaron su apoyo ante cualquier problema que tuve de forma incondicional, los quiero mucho, quiero dedicar este trabajo a mi tía Yaque, a mi primos Dany, Wicho, Yaimi y Daina y a mis sobrinitos lindos Cristobal y Diana, que los quiero con todas mis fuerzas a todos y espero que estén todos orgullosos de mí como yo siempre lo estoy y estaré de ustedes, los quiero a todos.

RESUMEN

El Departamento de Soluciones empresariales (SOLEM) del Centro de Informatización para la Gestión de Entidades (CEIGE) de la Universidad de las Ciencias Informáticas (UCI), tiene dentro de sus líneas de desarrollo una que está enfocada en el desarrollo de sistemas de gestión de combustible. Actualmente se le dio la tarea de desarrollar un sistema de gestión de flota de vehículos para la Dirección de transporte de la UCI.

En el presente trabajo se exponen los artefactos generados durante el desarrollo del módulo Control de combustible del Sistema de Control de Flota y Mantenimiento, con el objetivo de mejorar el proceso Control y análisis de índices de consumo de combustible de la dirección de transporte de la UCI, debido a que actualmente la información relacionada con este proceso se realiza de forma manual en formato duro y a través de documentos Excel lo cual trae como consecuencia que se puedan cometer errores, afectando esto la toma de decisiones del personal que trabaja en el área de Portadores energéticos de esta área.

Con la implementación de este módulo se satisfacen las necesidades plasmadas por el cliente en cuanto al control de combustible y se mejora la gestión de este proceso contribuyendo de esta forma a mejorar la toma de decisiones del personal de la dirección de transporte de la UCI.

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	5
1.1. Introducción	5
1.2. Gestión de combustible	5
1.2.1. Gestión y administración del combustible en la UCI	5
1.3. Sistemas de gestión de flotas de vehículos nacionales y extranjeros	6
1.3.1. Sistemas extranjeros	6
1.3.2. Sistemas nacionales.....	9
1.3.3. Análisis comparativo de los sistemas estudiados	11
1.4. Modelo de desarrollo orientado a componentes (1.1)	13
1.5. Arquitectura de software basada en componentes	13
1.5.2. Patrón arquitectónico Modelo-Vista-Controlador.....	14
1.6. Notación de modelado de procesos de negocio BPMN	14
1.7. Lenguaje unificado de modelado UML	15
1.8. Tecnología Ajax	16
1.9. Lenguaje de programación del lado del servidor	17
1.10. Lenguajes de programación del lado del cliente	17
1.11. Marco de Trabajo Sauxe (2.0)	18
1.12. Biblioteca de clases Highcharts (2.3.3)	20
1.13. Biblioteca de clases TCPDF (2.1)	20
1.14. Herramienta CASE Visual Paradigm (8.0)	21
1.15. Servidor de aplicaciones Apache (2.2)	21
1.16. Sistema gestor de base de datos PostgreSQL (9.1)	22
1.17. Navegador Firefox 4.0 en adelante	22
1.18. Conclusiones parciales	22
CAPÍTULO 2: MODELO DE NEGOCIO Y REQUERIMIENTOS	24
2.1. Introducción	24
2.2. Modelado del negocio	24
2.2.1. Mapa de procesos	24
2.2.2. Descripción de procesos de negocio	26
2.2.3. Patrones de control de flujo	28
2.2.4. Modelo conceptual.....	29
2.2.5. Validación del modelado de negocio.....	31
2.3. Requerimientos	31

2.3.1.	Técnicas para la captura de requisitos.....	31
2.3.2.	Especificaciones de los requisitos funcionales.....	33
2.3.3.	Administración de los requisitos.....	35
2.3.4.	Priorización de requisitos.....	36
2.3.5.	Validación de requisitos.....	37
2.4.	Requisitos no funcionales	38
2.5.	Conclusiones parciales	39
CAPÍTULO 3: DISEÑO, IMPLEMENTACIÓN Y PRUEBAS		40
3.1.	Introducción	40
3.2.	Diseño	40
3.2.1.	Mecanismos de diseño.....	40
3.2.2.	Diagramas de clases del diseño.....	42
3.2.3.	Descripción de las clases del diseño.....	44
3.2.4.	Diagramas de secuencia.....	46
3.2.5.	Patrones de diseño.....	46
3.2.6.	Métricas para evaluar el diseño propuesto.....	48
3.2.7.	Modelo de datos.....	50
3.3.	Implementación	50
3.3.7.	Diagrama de componentes.....	50
3.3.8.	Modelo de despliegue.....	51
3.3.9.	Estándares de codificación.....	52
3.3.10.	Publicación de servicios entre componentes.....	54
3.4.	Pruebas de software	55
3.4.7.	Pruebas de Caja Blanca.....	55
3.4.8.	Pruebas de Caja Negra.....	60
3.5.	Conclusiones parciales	62
CONCLUSIONES		63
RECOMENDACIONES		64
BIBLIOGRAFÍA		65

INTRODUCCIÓN

Para las empresas con flotas de vehículos el transporte es una actividad que le aporta beneficios, no solo por los ahorros que genera, sino por la calidad de los servicios que proporciona de producción y distribución.

La gestión de las flotas de vehículos se basa principalmente en controlar aspectos relacionados con el consumo de combustible, los kilómetros recorridos, mantenimientos preventivos planificados (aquellos mantenimientos que se planifican con el objetivo de evitar que ocurra una avería) y correctivos (cuando ocurre una avería). Los sistemas de gestión de flotas de vehículos permiten tener toda la información antes mencionada organizada, centralizada e integrada, por cuanto las empresas con flotas de vehículos están optando por su utilización ya que contribuyen a mejorar la rentabilidad de la inversión en la flota y a la toma de decisiones.

El Departamento de Soluciones Empresariales (SOLEM) del Centro de Informatización para la Gestión de Entidades (CEIGE) de la Universidad de las Ciencias Informáticas (UCI), tiene dentro de sus líneas de desarrollo una que está enfocada en el desarrollo de sistemas de gestión de este tipo. Actualmente se le dio la tarea de desarrollar un sistema de gestión de flota de vehículos para la dirección de transporte de la UCI, su flota de vehículos está compuesta por vehículos ligeros (peso máximo de carga no excede de 1,5 toneladas, tales como ciclos con motor, motocicletas, autos, paneles) y vehículos pesados (peso máximo de carga excede de 1,5 toneladas, tales como ómnibus, camiones, cuñas) que brindan servicios de transporte al personal de la Universidad, de carga y distribución de alimentos.

Uno de los procesos que se gestionan en la dirección de transporte de la UCI es el Control y análisis de los índices de consumo de combustible, el cual maneja toda la información referente a la elaboración de los planes de combustible por tipo (diesel y gasolina) y la asignación de combustible a los diferentes vehículos que forman parte del parque vehicular de la Universidad, así como la realización de los cierres de consumo mensual y anual por tipo de vehículo, tipo de actividad (administrativa, servicios, protocolo, transporte de pasajeros, logística, tecnológico, compensados) y tipo de combustible.

Actualmente la información relacionada con este proceso se gestiona de forma manual en formato duro y a través de documentos Excel, lo cual trae como consecuencia que se puedan cometer errores con el control del combustible, afectando esto la toma de decisiones del personal que trabaja en el área de

Portadores energéticos de la dirección de transporte de la UCI. Además para el especialista general que trabaja en esta área resulta un proceso muy engorroso realizar el cierre de consumo de combustible del mes ya que tiene que hacer muchos cálculos utilizando las fórmulas que brinda el Excel en función de obtener esta información, además de que al solicitarle información relacionada con el combustible tiene que revisar uno por uno los planes de combustibles y los cierres de consumo realizados mensualmente.

Sobre la base de los elementos expuestos anteriormente se formula el siguiente **problema a resolver** ¿Cómo mejorar la gestión del proceso Control y análisis de índices de consumo de combustible de la dirección de transporte de la UCI?

Se plantea como **objeto de estudio** Control del combustible en los Sistemas de gestión de flotas de vehículos. Y el **campo de acción** se enmarca en: el proceso Control y análisis de índices de consumo de combustible del Sistema de Control de Flota y Mantenimiento de la dirección de transporte de la UCI.

Para la solución del problema planteado se define como **objetivo general**: Desarrollar el módulo Control de combustible del Sistema de Control de Flota y Mantenimiento de la dirección de transporte de la Universidad de las Ciencias Informáticas.

Para dar cumplimiento al objetivo general planteado se definen los siguientes **objetivos específicos**:

1. Elaborar el marco teórico de la investigación relativo al control de combustible en los sistemas de gestión de flotas de vehículos.
2. Realizar el análisis del módulo Control de combustible del Sistema de Control de Flota y Mantenimiento de la dirección de transporte de la UCI.
3. Diseñar del módulo Control de combustible del Sistema de Control de Flota y Mantenimiento de la dirección de transporte de la UCI.
4. Realizar la implementación del módulo Control de combustible del Sistema de Control de Flota y Mantenimiento de la dirección de transporte de la UCI.
5. Validar la propuesta de solución mediante la aplicación de técnicas, métricas y pruebas.

Se plantea como **idea a defender**: El desarrollo del módulo Control de combustible del Sistema de Control de Flota y Mantenimiento permitirá mejorar la gestión del proceso Control y análisis de índices de consumo de combustible del área de Transporte de la UCI.

Durante el transcurso del trabajo de diploma y para su desarrollo se emplearon varios **Métodos de Investigación** que se exponen a continuación:

Métodos teóricos.

Modelación: Para la elaboración de algunos artefactos como: diagramas de procesos de negocio, modelo conceptual y diagramas de clases del diseño.

Métodos empíricos.

Entrevista: es un método que se aplica para identificar los procesos de negocio y los requisitos funcionales de la solución propuesta.

JAD¹: es una técnica de definición de requisitos y de diseño de la interfaz de usuario, basada en reuniones participativas entre clientes, directiva y desarrolladores.

Tormenta de ideas: es un método de trabajo grupal que facilita el surgimiento de nuevas ideas sobre un tema o problema determinado.

El presente trabajo de diploma se encuentra estructurado de la siguiente forma: introducción, tres capítulos, conclusiones, recomendaciones, anexos y la bibliografía consultada. A continuación se describe detalladamente el contenido que se abordará en cada uno de los capítulos definidos.

Capítulo 1. Fundamentación Teórica: Se describen los conceptos fundamentales relacionados con el proceso Control y análisis de índices de consumo de combustible del Sistema de Control de Flota y Mantenimiento para la dirección de transporte de la UCI. Se realiza una valoración de los diferentes sistemas de control de flota existentes en el mundo. Se efectúa una breve explicación del modelado de negocio y las técnicas empleadas para la captura de requisitos. Además se justifica el empleo de cada una de las herramientas, metodologías, lenguaje de desarrollo, frameworks, sistema gestor de base de

¹ Desarrollo conjunto de aplicaciones del inglés Joint Application Development.

datos y aplicaciones en la implementación del proceso Control y análisis de índices de consumo de combustible del Sistema de Control de Flota y Mantenimiento de la dirección de transporte de la UCI.

Capítulo 2. Modelado del negocio y Requerimientos: En este capítulo se generan los artefactos relacionados con la modelación del proceso de negocio Control y análisis de índices de consumo de combustible y la identificación de los requisitos funcionales y no funcionales de la propuesta de solución. Además se validan los artefactos generados mediante el uso de técnicas.

Capítulo 3. Diseño, implementación y pruebas de la propuesta de solución: En este capítulo se exponen los artefactos generados durante el diseño y la implementación de la solución, así como las métricas y pruebas utilizadas para su validación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción

El presente capítulo muestra las características principales del proceso Control y análisis de índices de consumo de combustible del Sistema de Control de Flota y Mantenimiento, la importancia que tiene para la dirección de transporte de la UCI, así como los procesos de esta área con los que se relaciona. Se realiza un análisis de la gestión de combustible en diferentes sistemas de gestión de flota vehiculares existente en Cuba y el resto del mundo. Este capítulo finaliza con la justificación de la selección de cada una de las herramientas, metodología, lenguajes de desarrollo, frameworks, sistemas gestores de base de datos y servidores web utilizados para el desarrollo de la propuesta de solución.

1.2. Gestión de combustible

Se entiende por gestión de combustible al diseño y la puesta en práctica de un sistema de control, supervisión y seguimiento del consumo de combustible global e individual de los vehículos de una flota de transporte. La gestión de combustible permite aprovechar de la manera más rentable cada litro de combustible adquirido, contribuyendo con ello no sólo a la economía de la empresa, sino también al ahorro energético y a la conservación del medio ambiente (Energía, 2007). Una adecuada gestión de combustible está asociada a:

- Una adecuada planificación de rutas y vehículos.
- La utilización de técnicas de conducción eficiente.
- Un correcto mantenimiento de los vehículos.
- La calidad del servicio prestado al cliente.

1.2.1. Gestión y administración del combustible en la UCI

La gestión del combustible en la UCI se realiza en diferentes niveles, un nivel externo que incluye la solicitud y entrega del combustible a la institución por parte del Ministerio de Educación Superior y uno interno que tiene el objetivo de distribuir el combustible por las diferentes áreas de la Universidad que tienen vehículos asignados. El proceso que maneja la gestión del combustible en esta área se llama:

Control y análisis de índices de consumo de combustible y comienza con la elaboración de los planes de combustible por el especialista en Portadores energéticos. Este plan se realiza mensual y por tipo de combustible (diesel y gasolina) en función del kilometraje previsto a recorrer de los vehículos y los índices de consumo de cada tipo de vehículo según el fabricante. Este plan de combustible es aprobado por el rector de la Universidad y en función de esto se procede a realizar la asignación de combustible para cada vehículo. Esta asignación comienza con la entrega de los documentos anexo de combustible y certificación de combustible habilitado y kilómetros recorridos, documentos estos con los cuales el chofer se presenta en la caja para que le entreguen la tarjeta de combustible. En función de toda la información mencionada anteriormente el especialista realiza el análisis de eficiencia de cada uno de los vehículos de la flota, teniendo en cuenta los estados (Correcto, Distorsionado mayor que 5 y Distorsionado menor que 5). Luego procede a realizar el cierre de consumo del mes de todo el parque vehicular teniendo en cuenta los indicadores establecidos en la 1era edición de la norma cubana llamada: Transporte automotor. Servicio de Transportación de pasajeros y cargas. Términos, definiciones, símbolos y métodos de cálculos, emitida en febrero del 2011 por la Oficina Nacional de Normalización (NC) (Normalización, 2011). Específicamente de esta norma se van a tener en cuenta los indicadores definidos para la intensidad energética (cantidad de combustible que se requiere para producir una unidad de tráfico) y el rendimiento energético (tráfico que se genera con un litro de combustible en la explotación del vehículo).

1.3. Sistemas de gestión de flotas de vehículos nacionales y extranjeros

En el mundo existen diferentes sistemas de gestión de flota de vehículos que controlan la información relacionada con el consumo de combustible, los kilómetros recorridos, mantenimientos preventivos planificados y correctivos, por lo que se realiza un estudio de los mismos para determinar cuáles pudieran contribuir y servir de apoyo para el planteamiento de la solución a la cual se desea arribar al culminar esta investigación.

1.3.1. Sistemas extranjeros

➤ Gestión de Flota:

El sistema de Gestión de Flotas de Informática EUGCOM² es un completo sistema computacional diseñado para la sólida y correcta administración de flotas de vehículos. Permite administrar de una manera óptima toda la información de los vehículos que la empresa maneja y administra (EUGCOM, 2009). Presenta los siguientes módulos:

Vehículos: Módulo de Mantenimiento (Permite Crear, Modificar, Eliminar e Imprimir) de todos los datos de los vehículos del sistema (EUGCOM, 2009).

Choferes: Módulo de ingreso y mantención de los datos de los choferes existentes en la flota (EUGCOM, 2009).

Combustibles: Módulo de ingreso y mantención de los datos que componen el movimiento de cargas de combustible efectuados a los vehículos (EUGCOM, 2009).

Viajes: Pantalla de ingreso y mantención de los datos ingresados al registrar un movimiento de viaje del vehículo (salida y entrada de vehículos) (EUGCOM, 2009).

Neumáticos: Módulo de ingreso y mantención de los datos que componen el movimiento de cambio de neumáticos de los vehículos (EUGCOM, 2009).

Orden de Trabajo: Módulo de ingreso y mantención de las órdenes de trabajo emitidas a los vehículos para efectuar una cierta cantidad de servicios en él (EUGCOM, 2009).

Cargas de Combustible: Módulo de búsqueda de datos que permite ejecutar consultas y obtener resultados acerca de las cargas de combustible registradas en el sistema. Genera un informe sumamente personalizado según los requerimientos del usuario y permite calcular los datos más importantes por cada vehículo (EUGCOM, 2009).

Viajes Realizados: Módulo de búsqueda de datos de los movimientos de viajes realizados por los vehículos en un lapso de tiempo y según las especificaciones establecidas por el usuario (EUGCOM, 2009).

Reprocesamiento de Cargas de Combustible: Módulo que permite reprocesar los datos automáticos generados en el movimiento de cargas de combustible como distancia y rendimiento, y genera un informe

² Empresa informática dedicada al desarrollo de diferentes software.

de errores de las cargas de combustible registradas en el sistema, y es capaz de reparar automáticamente los errores más usuales. Este módulo evita que los informes de combustible contengan información no válida después de editar o eliminar un movimiento (EUGCOM, 2009).

➤ **KreaFlota:**

KreaFlota, un software diseñado y concebido para controlar los aspectos técnicos de cualquier flota vehicular, dispone de potentes módulos para la correcta gestión de cada uno de los vehículos que dispone la empresa (KreaSoft , 2010).

Lleva un control detallado de los vehículos de la flota, para esto prepara una ficha en la cual se especifican todos los datos de cada uno de los vehículos, controlando alarmas de cambios de aceites, revisión técnica, pólizas de seguro. Dispone de una ficha detalla de choferes para tener una base de datos exacta de cada uno de los choferes que contrate la empresa, para así tener un control de la información y determinar su comportamiento (KreaSoft , 2010).

Cuenta con un módulo que permite identificar y mantener un maestro de servicios y repuestos, indicando datos detallados de precios y costos, para luego poder determinar los valores de los servicios realizados a los vehículos de la flota. Además lleva un control exhaustivo del petróleo, llevando un registro diario de consumo para los vehículos permitiendo obtener rendimientos de los mismos. Los módulos y procesos mencionados anteriormente están acompañados de una serie de informes que permitan gestionar y tomar decisiones correctas, previniendo el desgaste excesivo o anticipado de los vehículos que mantiene la flota (KreaSoft , 2010).

➤ **FlotaWeb:**

FlotaWeb es una plataforma de Administración de Flotas basado en Internet, que permite gestionar todos los procesos relacionados con la flota vehicular. Funciona perfectamente para flotas de cualquier tamaño o sector, no importa si tiene 3 vehículos o 1000, ya que habilita funcionalidades de acuerdo a sus necesidades. Puede controlar todos los costos, evaluar rendimientos y costos por kilómetros u horas, administrar eficientemente las llantas y el combustible, recibir alertas preventivas de mantenimientos y documentación (Software Incremental, 2010).

FlotaWeb está dividido por módulos que desarrollan las siguientes funcionalidades:

Vehículos: Centraliza la información completa de los vehículos y todos los procesos que tienen que ver con su correcta administración (Software Incremental, 2010).

Combustible: Permite controlar y supervisar los costos de combustible en la flota, analizar rendimientos, y obtener informes estadísticos de los detalles de consumo por proveedor, ciudades, centros de costo y operaciones (Software Incremental, 2010).

Novedades de Mantenimiento: Permite centralizar todos los reportes de los conductores y tareas pendientes de mantenimiento en una sola aplicación, crear órdenes de trabajo basadas en Novedades de Mantenimiento y supervisar el estado de las novedades ejecutadas y las pendientes (Software Incremental, 2010).

Órdenes Trabajo: Es el proceso donde se le asocia a un vehículo una serie de actividades que involucran trabajos, insumos, rutinas, costos, tiempos, proveedores, mecánicos, presupuestos y alertas (Software Incremental, 2010).

Programación y Calendario de Mantenimiento: Permite planificar y ver los mantenimientos de la flota de una forma gráfica. Utiliza calendarios para ver qué actividades se han realizado y cuáles están pendientes (Software Incremental, 2010).

Rutas: Este módulo está compuesto por la configuración de las rutas y sus duraciones en kilómetros y días, además de la preconfiguración de los gastos de una ruta (Software Incremental, 2010).

1.3.2. Sistemas nacionales

➤ **Apolo:**

Apolo es un producto nacional perteneciente al Ministerio de la Informática y las Comunicaciones (MIC) y la Empresa Cubana Nacional de Software (Desoft S.A), encargada de su desarrollo. Es un sistema cliente-servidor, donde la base de datos está en un servidor central y los usuarios del sistema independientemente del lugar donde se encuentren accederán en tiempo real. Es una herramienta enfocada a gestionar todo tipo de flotas y ofrecer soluciones de asistencia a la gestión y toma de decisiones, con el objetivo de optimizar la rentabilidad de la inversión mediante la reducción de los costos fijos y variables (Desoft, 2004).

Apolo brinda una base informativa muy potente, donde la mayoría de la información que se brinda es directamente definible por un usuario de nivel administrativo del sistema, para esto se precisan algunos nomencladores como: marca de vehículos, tipos de combustible, tipo de usos de cada uno de los tipos de flotas, de la asignación de combustible, lubricantes y líquidos de freno, la cantidad de Km programados para cada marca o línea e información del personal de la entidad. Apolo puede trabajar directamente vinculado con el sistema de talleres, bases o agencias de renta y departamentos. Mantiene un registro de los autorizados de parqueo emitidos por la entidad, que pueden ser temporales o permanente y un registro de hoja de ruta, el cual representa el documento oficial para la circulación del vehículo. Permite planificar la entrega mensual de combustible a la flota en función del kilometraje a recorrer y los índices de consumo de cada línea según el fabricante. El sistema permite crear nuevas tarjetas de combustible (se utilizan para habilitar combustible a un vehículo) o generar una nueva asignación. Admite despachar una cantidad adicional de combustible, donde el sistema calcula automáticamente la cantidad de litros de combustible a entregar, teniendo en cuenta el índice de consumo del vehículo y el kilometraje planificado. Además establece una relación entre el combustible planificado en un período para una línea y el kilometraje caminado según la hoja de ruta a través de tarjetas virtuales (Desoft, 2004).

➤ **Sistema de Gestión de Mantenimiento Vehicular:**

El Sistema de Mantenimiento Vehicular realizado por CEIGE tiene como propósito gestionar los procesos de mantenimiento que se desarrollan en el área de Transporte de la Policía Nacional Bolivariana. Este sistema fue desarrollado basado en los principios de independencia tecnológica utilizando software libre. Mediante su utilización en la Policía Nacional Bolivariana puede gestionar las asignaciones, recepciones, accidentes, inspecciones técnicas, órdenes de trabajo, informes de baja y de resultado de las unidades policiales de forma centralizada, eficiente y segura. Este sistema de gestión tiene toda la información referente a los procesos de mantenimiento preventivo planificado y correctivo de una flota de vehículos, sin importar el tamaño de esta y garantizando una correcta planificación, organización y control en la gestión del mantenimiento de los vehículos (ALBET, 2011).

Presenta los siguientes módulos:

Estructura y Composición: Este módulo es el encargado de crear, actualizar y eliminar las estructuras. Además le brinda la posibilidad de definir la organización jerárquica que tienen los elementos que las

componen. Puede también establecer las estructuras en cada una de las unidades a través de las áreas (ALBET, 2011).

Clasificadores: En este módulo se insertan, modifican y eliminan tipos de mantenimientos, accesorios, causas de fallas, repuestos, herramientas, tipos de unidades, documentos técnicos y unidades de medida (ALBET, 2011).

Configuración: Es el módulo a través del cual se definen los diferentes grupos de unidades de acuerdo a su marca, modelo, régimen de mantenimiento, entre otras características. También se define el umbral de mantenimiento por el cual se van a registrar todas las unidades policiales para la realización de los mantenimientos preventivos planificados que le corresponden y los clientes (que son aquellas dependencias a las cuales se le van a gestionar los procesos de mantenimiento) (ALBET, 2011).

Persona: Es el módulo que gestiona toda la información referente a los recursos humanos (agregar, modificar, eliminar, imprimir) (ALBET, 2011).

Vehículo: Es el módulo desde el cual se va a gestionar toda la información de las unidades policiales (agregar, modificar, consultar, asignar a dependencia, registrar accidentes, realizar inspecciones técnicas y recepción de unidades) (ALBET, 2011).

Taller: Es el módulo que gestiona los órdenes de trabajo (generar, emitir, modificar, cancelar, registrar recursos utilizados en una orden). También gestiona las inspecciones técnicas (agregar, modificar, cancelar e imprimir) (ALBET, 2011).

Documentos: Es el módulo que gestiona todos los informes generados para las unidades policiales ya sean de baja o de resultado, así como los memorándums (ALBET, 2011).

1.3.3. Análisis comparativo de los sistemas estudiados

Los sistemas nacionales e internacionales anteriormente mencionados se analizaron con el objetivo de conocer cómo gestionaban toda la información relacionada con el control del combustible y determinar si era factible su utilización en el desarrollo de la propuesta de solución. Para este análisis se definen los siguientes indicadores: gestión del control de combustible, tipo de tecnología sobre la cual están desarrollados y tipo de aplicación (web o escritorio). El análisis realizado teniendo en cuenta los indicadores definidos arroja los siguientes resultados:

En cuanto a la gestión del combustible:

- Los sistemas Gestión de Flota, KreaFlota y FlotaWeb cuentan con un módulo para la gestión del combustible que permite tener un registro diario del consumo de los vehículos en función de obtener su rendimiento, el ingreso de datos relacionados a la carga de combustible realizada a los vehículos, generar informes y calcular los datos más importantes por cada vehículo. Sin embargo sus funcionalidades no se pueden reutilizar como parte de la propuesta de solución del módulo Control de combustible del Sistema de Control de Flota y Mantenimiento de la dirección de transporte de la UCI, debido a que no planifican la asignación de combustible y por ende no se pueden calcular los índices de consumos de acuerdo al plan y los reales, información ésta que es relevante en la dirección de transporte de la UCI para realizar los cierres de consumo.
- Apolo presenta grandes ventajas y funcionalidades en cuanto a la gestión del combustible, realiza una planificación de la entrega de combustible a la flota de acuerdo a los kilómetros que debe recorrer y los índices de consumo, y establece un relación entre el combustible planificado y el kilometraje recorrido según la hoja de ruta a través del uso de tarjetas virtuales. El uso de tarjetas virtuales puede constituir una posible solución a tener en cuenta para el desarrollo del sistema deseado, ya que hace más fácil y automática la planificación y asignación mensual de combustible. Sin embargo en esta primera versión del desarrollo del módulo de Control de combustible no se reutiliza nada de este sistema, ya que no se van a informatizar los procesos que se manejan en el área de Economía relacionados con el control de las tarjetas de combustible de los diferentes vehículos del parque vehicular de la UCI.
- El Sistema de Gestión de Mantenimiento Vehicular constituye la base del desarrollo de la solución que se desea, pero con relación a la gestión de combustible no posee un módulo que gestione esta información necesaria en la dirección de transporte de la UCI.

En cuanto al tipo de tecnología sobre el cual están desarrollados:

- Los sistemas Gestión de Flota, KreaFlota y FlotaWeb son software propietario, además de que utilizan y se ejecutan sobre tecnologías propietarias lo cual impide su utilización debido a que el CEIGE tiene como política el uso de tecnologías libres para el desarrollo de sus productos.

En cuanto al tipo de aplicación:

- Gestión de Flota, KreaFlota son aplicaciones de escritorio, por lo que su acceso se limita al ordenador donde están instaladas, para configurar o actualizar cada una de las instalaciones depende de las necesidades de cada usuario, son dependientes del sistema operativo que utilice el ordenador y sus capacidades (video, memoria). El sistema que se quiere realizar para la dirección de transporte de la UCI debe ser una aplicación web, que montada en un servidor central tengan acceso todas las subdirecciones de Transporte de la UCI.

1.4. Modelo de desarrollo orientado a componentes (1.1)

Con el objetivo de garantizar el correcto funcionamiento del desarrollo del centro CEIGE, en cada uno de sus departamentos, se lleva a cabo un modelo de gestión basado en los modelos de líneas de productos de software utilizados internacionalmente y con diversos casos de aplicación exitosa en la propia Universidad. Este modelo de desarrollo orientado a componentes en su versión 1.1 seleccionado por el CEIGE, tiene como principios la orientación a dominios del negocio, la especialización y reutilización de los recursos humanos en varios proyectos y el aumento de la productividad. Describe las fases del ciclo de vida del desarrollo de los proyectos del centro y las actividades y artefactos a desarrollar en cada una (CEIGE, 2012).

1.5. Arquitectura de software basada en componentes

La arquitectura basada en componentes consiste en una rama de la Ingeniería de Software en la cual se trata con énfasis la descomposición del software en componentes funcionales. Es completamente modular y favorece la reutilización de todos sus elementos, debido a que el sistema será dividido completamente en componentes, facilitando un mejor entendimiento del desarrollo y logrando una mejor integración entre ellos (Frank Buschmann, 1996).

Para el desarrollo de sus productos, CEIGE define la utilización de la arquitectura basada en componentes ya que permite alcanzar un mayor nivel de reutilización de software, donde las pruebas podrán ser ejecutadas probando cada uno de los componentes antes de probar el conjunto completo de componentes ensamblados. Además, en caso de existir un débil acoplamiento entre componentes, el desarrollador es libre de actualizar o agregarlos según sea necesario, sin afectar otras partes del sistema.

1.5.2. Patrón arquitectónico Modelo-Vista-Controlador

Para el desarrollo del módulo Control de combustible se empleó el patrón MVC debido a que su utilización fue definida por el CEIGE, además de estar implementado dentro del marco de trabajo Sauxe, sobre el cual se desarrolla el sistema. Este patrón divide la aplicación en tres componentes: el Modelo, la Vista y el Controlador. El Modelo se encarga de administrar el comportamiento y los datos del dominio de la aplicación, responder a los requerimientos de información sobre su estado y las instrucciones de cambiar el mismo. Mantiene el conocimiento del sistema y no depende de ninguna Vista o Controlador. La Vista maneja la visualización de la información, mientras que el Controlador analiza los mensajes de eventos que recibe el sistema, modifica u obtiene datos del Modelo en respuesta a las peticiones del usuario (López, 2012).

1.6. Notación de modelado de procesos de negocio BPMN

La Notación para el Modelado de Procesos de Negocio (BPMN) es un estándar basado en los diagramas de flujo, adaptado para suministrar una notación gráfica que describe la lógica de los pasos de un proceso de negocio a través de flujos de trabajo. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades (Bizagi, 2012).

Por la importancia que tiene este tipo de modelado se utiliza para la modelación del proceso Control y análisis de índices de consumo de combustible del Sistema de Control de Flota y Mantenimiento de la dirección de transporte de la UCI. Es independiente de cualquier metodología de modelado de procesos, crea un puente estandarizado para disminuir la brecha entre los procesos de negocio y la implementación de estos; permite modelar los procesos de una manera unificada y estandarizada permitiendo un entendimiento entre todas las personas de la organización.

Con BPMN se pueden aplicar un conjunto de patrones de control de flujo con el objetivo de modelar actividades organizadas de la mejor forma para resolver un tipo de problema determinado. A continuación se muestra un listado de los patrones que se pueden aplicar al modelar con BPMN:

Elección exclusiva: representa un punto en el proceso donde la decisión se toma por elección de una de las posibles rutas. Está marcada por un punto de decisión exclusivo basado en datos o en eventos.

Secuencia: se requiere cuando hay una dependencia entre dos actividades, de tal forma que una actividad no pueda iniciarse antes de que otra haya terminado.

Unión exclusiva: representa un punto en donde dos o más bifurcaciones son unidas sin sincronización y donde ninguna de estas bifurcaciones ha sido ejecutada en paralelo con alguna otra. Usa un punto de decisión exclusivo basado en datos.

División paralela: marca el punto a partir del cual todas las rutas de salida son continuadas en paralelo. Utiliza un punto de división paralelo en modo división.

Unión paralela: espera a que todas las señales de entrada lleguen al punto de decisión para obtener una respuesta de salida. Utiliza el punto de decisión paralelo en modo unión.

Elección múltiple: marca un punto donde después de considerar una cierta decisión, un número de bifurcaciones puede ser seleccionado. Usa un punto de decisión incluido en modo división.

Unión sincronizada: marca un punto donde múltiples rutas convergen en una única salida bajo la condición de sincronización, esto significa que el proceso espera a que se completen todas las rutas. Se usa un punto de decisión incluido en modo unión.

Ciclos arbitrarios: consiste en puntos de decisión que hacen que se vuelva a una actividad anterior del proceso para que parte de él se repita.

1.7. Lenguaje unificado de modelado UML

UML³ es un lenguaje gráfico utilizado para visualizar, especificar, construir y documentar los artefactos del sistema de software (Jacobson, et al., 2000). Se utiliza este lenguaje por las siguientes ventajas:

- Posibilita la descripción de sistemas, simplificando la complejidad de estos y sin pérdida de información, haciendo posible la comprensión del sistema tanto para usuarios como desarrolladores.
- Ofrece varios diagramas para el modelado de sistemas.

³ Lenguaje Unificado de Modelado del inglés Unified Modeling Language.

- Brinda facilidades para el diseño, documentación, reutilización de código y detecciones de fallas.
- Facilita la comunicación entre desarrolladores, permite ahorrar tiempo en el desarrollo del software y hace más sencillas las modificaciones que se vayan a realizar.

1.8. Tecnología Ajax

El término AJAX es un acrónimo de Asynchronous JavaScript + XML, que se traduce como “JavaScript asíncrono + XML”. Es una técnica de desarrollo web para la creación de aplicaciones interactivas. Éstas se ejecutan en el cliente y mantiene comunicación asíncrona con el servidor sin interferir con la visualización ni el comportamiento de la página (Jesse James Garrett, 2005). En realidad, hace de intermediario entre el servidor y el usuario, anticipando peticiones de datos al servidor, de modo que cuando el usuario efectúa una consulta determinada, AJAX ya tiene listos los datos y los muestra directamente, sin tener que volver a hacer una petición al servidor, con la consecuente espera que ello supone.

AJAX combina otras tecnologías existentes: XHTML⁴ y hojas de estilos en cascada (CSS⁵) para presentar la información basada en estándares; (DOM⁶), para interactuar dinámicamente con los datos; XML⁷ y XSLT⁸, para intercambiar y manipular datos; y XMLHttpRequest⁹ y JavaScript¹⁰, para recuperar los datos de forma desincronizada con un servidor web.

Para la implementación del módulo Control de combustible se utiliza AJAX por las ventajas que ofrece:

- El tiempo de espera para una petición se reduce. El usuario al hacer un pedido al servidor, no se envía toda la página.
- La experiencia de usuario en la navegación es mucho más rica. Ya que no se refresca la página constantemente al interactuar con ella.

⁴ Lenguaje Extensible de Marcado de Hipertexto del inglés Extensible Hypertext Markup Language.

⁵ Hojas de estilo en cascada del inglés Cascading Style Sheets.

⁶ Modelo de Objetos del Documento del inglés Document Object Model.

⁷ Lenguaje de Marcas Extensible del inglés Extensible Markup Language.

⁸ Lenguaje de Hojas Extensibles de Transformación del inglés Extensible Stylesheet Language.

⁹ Lenguaje de Marcas Extensible/Protocolo de Transferencia de Hipertextos del inglés Extensible Markup Language/ Hypertext Transfer Protocol.

¹⁰ Lenguaje de programación orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

- Mayor velocidad, debido a que no hay que refrescar toda la página nuevamente.

1.9. Lenguaje de programación del lado del servidor

“PHP es un lenguaje de script incrustado dentro del HTML. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas” (Heredia, Cristian Van Der y Herminio, 2001).

Su empleo para el desarrollo del módulo Control de combustible del Sistema de Control de Flota y Mantenimiento de la dirección de transporte de la UCI se debe a que el marco de trabajo Sauxe, sobre el cual se implementa el módulo, fue realizado con este lenguaje de programación, debido a sus ventajas:

- Es un lenguaje multiplataforma se interpreta y ejecuta de igual forma un script independientemente del tipo de plataforma (tipo de sistema operativo o tipo de servidor web) donde sea ejecutado (Gutiérrez, 2011).
- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL, facilitando así el desarrollo de aplicaciones web dinámicas que acceden a bases de datos en tiempo real.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Permite aplicar las técnicas de programación orientada a objetos.
- No requiere definición de tipos de variables, se evalúan en tiempo de ejecución, es decir, con darles un valor ya quedan definidas también en su tipo.
- Tiene manejo de excepciones (desde PHP5).

1.10. Lenguajes de programación del lado del cliente

Para el desarrollo del módulo Control de combustible se utiliza JavaScript como un lenguaje de programación web. Este lenguaje se maneja principalmente del lado del cliente permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, en bases de datos locales al navegador. Permite la creación de páginas web dinámicas, menús desplegable, efectos visuales sencillos y manipular datos, utilizando

poca memoria y manteniendo un tiempo de descarga rápido. Los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios (Pérez, 2011).

1.11. Marco de Trabajo Sauxe (2.0)

El marco de trabajo Sauxe desarrollado por el departamento de Tecnología del centro CEIGE, da solución a un sin número de escenarios o aspectos arquitectónicos como: gestión y configuración dinámica de cache, integración de componentes de forma distribuida o no distribuida, acceso a bases de datos a través de una capa de abstracción, gestión de concurrencia de recursos, administración centralizada de transacciones, gestión dinámica de las trazas generadas por los sistemas, implementación de mecanismos de autenticación y autorización, implementación de mecanismos de mensajería y control de excepciones, gestión y configuración dinámica de pre condiciones, pos condiciones y validaciones de variables, gestión y configuración de flujos de trabajo, visualización de las funcionalidades de un sistema, entre otros escenarios de alta complejidad, garantizando los atributos de calidad de los sistemas que se desarrollen con el mismo. Cuenta con una arquitectura en capas que a su vez presenta en su capa superior una implementación Modelo-Vista-Controlador (MVC) (Baryolo, 2010).

Siguiendo el paradigma de independencia tecnológica por el cual apuesta el país, Sauxe reutiliza las siguientes tecnologías libres:

ExtJS (3.4)

ExtJs es una biblioteca de clases JavaScript de alto rendimiento, compatible con la mayoría de los navegadores que permite crear páginas e interfaces web dinámicas. Esta biblioteca de clases incluye: Componentes UI (Interfaz de Usuario) de alto rendimiento y personalizables. Modelo de componentes extensibles. Un API fácil de usar y licencias de códigos abiertos y comerciales. Una de las grandes ventajas de utilizar ExtJs es que permite crear aplicaciones complejas utilizando componentes predefinidos, así como un manejador de layouts similar al que provee Java Swing¹¹, gracias a esto se evita el tedioso problema de validar que el código escrito funcione bien en cada uno de los navegadores (Firefox, Internet Explorer, Safari). Además, la ventana flotante que provee ExtJs es excelente por la forma

¹¹ Biblioteca gráfica de Java

en la que funciona. Al moverla o redimensionarla sólo se dibujan los bordes haciendo que el movimiento sea fluido (Baryolo, et al., 2010).

Zend Framework (1.11)

Se trata de un framework para el desarrollo de aplicaciones web y servicios web con PHP, brinda soluciones para construir sitios web modernos, robustos y seguros. La estructura de los componentes de Zend Framework es algo único; cada componente está construido con una baja dependencia de otros. Esta arquitectura débilmente acoplada permite a los desarrolladores utilizar los componentes por separado. A menudo se refiere a este tipo de diseño como "use-at-will" (uso a voluntad). (Inc, 2005-2011).

Es utilizado por las ventajas que proporciona (Zend Technologies, 2012):

- Trabaja con una arquitectura MVC.
- Cuenta con módulos para manejar archivos PDF, canales RSS, Servicios Web (Amazon, Flickr, Yahoo).
- El Marco de Zend también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultarlas, sin tener que escribir ninguna consulta SQL.
- Incluye soporte AJAX a través de JSON¹².
- Robustas clases para autenticación y filtrado de entrada.
- Clientes para servicios web, incluidos Google Data APIs, ProgrammableWeb y Strikelron.

Doctrine framework (1.2.2)

Doctrine no es más que un ORM¹³ para la persistencia de datos que trabaja con lenguaje de programación PHP, situándose sobre su poderosa capa de abstracción de la base de datos DBAL¹⁴. Su utilización brinda

¹² Notación de Objetos de JavaScript del inglés JavaScript Object Notation.

¹³ Mapeo Objeto-Relacional del inglés Object Relational Mapping

¹⁴ Capa de Abstracción de la Base de Datos del inglés Database Abstraction Layer.

grandes facilidades a los desarrolladores a la hora del trabajo con los datos, automatizando la generación de ficheros que responden a las tablas relacionales de la base de datos (Doctrine, 2012).

Alguna de las ventajas que ofrece este framework:

- La separación total del sistema de base de datos, por lo que si en un futuro se decidiera cambiar el motor de base de datos, esto no afectaría al sistema.
- La utilización de los métodos de un objeto de datos desde distintas partes de la aplicación.
- Posee un sistema para evitar tipos de ataque como pueden ser las inyecciones SQL.
- Ordena de forma correcta la capa de datos, por lo que facilita el mantenimiento del código.
- Cuenta con un lenguaje propio para hacer las consultas.
- Encapsula la lógica de los datos permitiendo hacer cambios que afecten toda la aplicación, únicamente modificando una función.

1.12. Biblioteca de clases Highcharts (2.3.3)

Highcharts es una biblioteca de clases¹⁵ escrita en JavaScript que permite la creación de gráficos, ofrece una forma fácil de añadir gráficos interactivos a un sitio web o aplicación web. Facilitando la lectura de datos y mostrando dinámicamente su representación gráfica, es ideal para quien trabaja con sitios web corporativos preocupados en mostrar resultados y estadísticas. Highcharts es compatible con todos los navegadores modernos. No es comercial, por lo que no necesita el permiso de los autores para su implementación en sitios web personales (Software, 2007).

Se utiliza para el desarrollo del módulo Control de combustible ya que la integración de esta biblioteca de clases con el marco de trabajo SAUXE permite representar los datos mediante diferentes tipos de gráficas, de forma sencilla e intuitiva. Además permite exportar dichas representaciones a archivos PDF.

1.13. Biblioteca de clases TCPDF (2.1)

TCPDF es una biblioteca de clases para el lenguaje de programación PHP la cual permite crear ficheros PDF de forma dinámica y rápida. Dos de las cualidades más apreciadas de esta clase, es su simplicidad a

¹⁵ Clases de programación orientada a objetos suministradas por terceros.

la hora de crear archivos PDF y la capacidad de interpretar código XHTML. Contiene métodos y funciones para justificar el texto del documento, manipular los encabezados y numeración automática de las páginas, numerar, alinear, incluir imágenes, aplicar determinados colores, incluir enlaces web, cifrar el documento e incluir código en JavaScript (Asuni, 2001-2011). Por las características anteriormente expuestas se decide utilizar esta biblioteca de clases para la creación de reportes en PDF en el módulo Control de combustible.

1.14. Herramienta CASE Visual Paradigm (8.0)

Visual Paradigm es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, ingeniería inversa, generar código desde diagramas y generar documentación (Paradigm, 2011). Se utiliza esta herramienta por las ventajas que posee:

- Disponibilidad en múltiples plataformas.
- Uso de un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación.
- Diseño centrado en casos de uso y enfocado al negocio que genera un software de mayor calidad.
- Capacidades de ingeniería directa (versión profesional) e inversa.
- Modelo y código que permanece sincronizado en todo el ciclo de desarrollo.
- Soporta aplicaciones web.
- Fácil de instalar y actualizar.
- Las imágenes y reportes generados son de muy buena calidad.

1.15. Servidor de aplicaciones Apache (2.2)

Apache es un servidor web de software libre, cuyo objetivo es servir o suministrar páginas web (en general, hipertextos) a los clientes web o navegadores que las solicitan, funciona sobre cualquier

plataforma, permite que otros ordenadores vean la web mediante un navegador. Es una solución altamente configurable y extensible a través de módulos, se integra perfectamente con varias tecnologías, lenguajes, plataformas, bases de datos (Profesorado, 2008). Por las características anteriormente expuestas se decide utilizarlo para el desarrollo del módulo Control de combustible.

1.16. Sistema gestor de base de datos PostgreSQL (9.1)

Para el módulo a implementar se decide la utilización de PostgreSQL en su versión 9.1 porque es un sistema gestor de base de datos relacional orientado a objetos de software libre que puede ser utilizado en los principales sistemas operativos: Linux, Unix, BSDs, Mac OS, Beos, Windows.

1.17. Navegador Firefox 4.0 en adelante

Mozilla Firefox es un navegador de Internet libre y de código abierto. Es usado para visualizar páginas web. Incluye corrector ortográfico, búsqueda progresiva, marcadores dinámicos y un sistema de búsqueda integrado que utiliza el motor de búsqueda que desee el usuario. Además se pueden añadir funciones a través de complementos desarrollados por terceros (Practicopedia, 2011).

Se utiliza Firefox debido a que:

- Las contraseñas son más seguras.
- Tiene extensiones útiles que se pueden descargar y ofrecen funcionalidades adicionales.
- Se actualiza de forma automática.
- Firefox bloquea las ventanas emergentes.
- Incorpora la navegación por pestañas, una forma increíblemente eficaz e intuitiva para tener varias páginas disponibles al mismo tiempo, sin saturar la barra de tareas.

1.18. Conclusiones parciales

Los sistemas de gestión de flota existentes en Cuba y el mundo analizados poseen algunos inconvenientes en cuanto a la gestión del combustible, tipo de tecnología y tipo de aplicación, los cuales impiden que sean utilizados para el desarrollo del módulo Control de combustible del Sistema de Control de Flota y Mantenimiento, por lo cual es necesario la creación de un sistema informático capaz de

Capítulo 1: Fundamentación teórica

satisfacer las necesidades actuales que presenta la dirección de transporte de la UCI con relación a la gestión del combustible. Además en este capítulo se seleccionan las herramientas, tecnologías y lenguajes de programación, diseño y modelado a utilizar para el desarrollo del módulo Control de combustible a través del cual se realiza la gestión del combustible del área de Transporte de la UCI.

CAPÍTULO 2: MODELO DE NEGOCIO Y REQUERIMIENTOS

2.1. Introducción

En este capítulo se realiza la descripción del proceso Control y análisis de índices de consumo de combustible de la dirección de transporte de la UCI. Se aplican patrones de control de flujo en la modelación del proceso. Se realizan los mapas de procesos donde se presentan los artefactos relacionados con la gestión del control de combustible. Se elabora el modelo conceptual con los principales conceptos que se manejan en el área de transporte de la UCI relacionados con el control de combustible. Además se identifican los requisitos funcionales a partir del uso de técnicas, se especifican y validan.

2.2. Modelado del negocio

El primer paso hacia la construcción de cualquier aplicación es desarrollar el modelo de negocio, el cual tiene como finalidad describir los procesos de negocio, especificando sus datos, identificando quiénes participan y las actividades que requieren informatización (Sanchidiran, et al., 2011).

2.2.1. Mapa de procesos

Un mapa de proceso contribuye hacer visible los procesos que componen un sistema así como sus relaciones principales (Badillo, 2012).

Para la elaboración de los mapas de procesos del módulo Control de combustible del Sistema de Control de Flota y Mantenimiento fue empleada una plantilla conformada por el CEIGE (consultar documentos entregables CIG-CFM-N-MTTO-i1101.xls y CIG-CFM-N-MTTO-i1104.xls), en la cual se representan los procesos identificados en el área de Transporte de la UCI agrupados por niveles, colocando en un nivel 0 los procesos claves¹⁶ de esta organización y en un nivel 1 sus procesos derivados. Dentro de los procesos del nivel 0 se encuentran: Disponibilidad del parque automotor, Control de reparaciones y mantenimiento y Control y análisis de índices de consumo de combustible, en el documento CIG-CFM-N-MTTO-i1101.xls se especifica información más detalla sobre ellos.

¹⁶ Procesos fundamentales para alcanzar los objetivos estratégicos de la organización.

Capítulo 3: Diseño, Implementación Y Pruebas

El contenido de este trabajo se enmarca en los procesos Elaboración de planes de combustible y Asignación de combustible pertenecientes al nivel 1 del proceso Control y análisis de índices de combustible.

Para los procesos identificados en el nivel 1 se crea un documento que contiene: nombre del proceso, breve descripción, referencia, nivel y proceso padre al que pertenecen (consultar documento entregable CIG-CFM-N-MTTO-i1104.xls).

Se identifican para cada uno de los procesos del nivel 1 los artefactos de entrada y salida involucrados en el proceso y el formato en que se manejan actualmente en el área de Transporte de la UCI.

La matriz de procesos contiene la relación que existe entre los procesos, sistemas o involucrados en el proceso, exponiendo los artefactos de salida de un proceso que constituyen entradas para otros y viceversa (consultar Tabla 1 y documento entregable CIG-CFM-N-MTTO-i1104.doc).

Tabla 1 Matriz de relación entre los procesos de negocio de nivel 1.

Entradas							
		Elaboración de planes de combustible	Asignación de combustible	Rectoría	Chofer	Cajera	Área de portadores energéticos
Salidas	Elaboración de planes de combustible			Plan de combustible			Plan de combustible
	Asignación de combustible				Tarjeta de combustible	Anexo de combustible	
					Certificación de combustible habilitado y kilómetros recorridos	Certificación de combustible habilitado y kilómetros recorridos	

Rectoría	Plan operativo					
Chofer						
Cajera				Acta de responsabilidad material		
Área de portadores energéticos						

2.2.2. Descripción de procesos de negocio

Los procesos de negocio que se describen son los que se encuentran en el último nivel, en este caso Elaboración de planes de combustible y Asignación de combustible.

2.2.2.1. Descripción del proceso Elaboración de planes de combustible

Este proceso comienza con la elaboración de los planes de combustible general (también llamado plan de la base) y particular por tipo de combustible (diesel y gasolina) a partir de la información que contiene el plan operativo (contiene la asignación de combustible para la UCI entregada por el Ministerio de Educación Superior). Luego se procede a realizar la discusión de estos planes de combustible con cada una de las áreas de la Universidad que tienen vehículos asignados y con el Vicerrector Económico. Posteriormente son presentados a la Rectora para su aprobación final. Para obtener mayor información sobre la descripción de este proceso (consultar Anexo 1 y 3 y documento entregable CIG-CFM-N-MTTO-i1507.doc).

Tabla 2 Artefactos de entrada y salida del proceso Elaboración de planes de combustibles.

Proceso Elaboración de plan de combustible	
Artefactos de Entrada	Artefactos de Salida

Plan operativo	Plan de combustible diesel
	Plan de combustible de la base
	Plan de combustible gasolina

2.2.2.2. Descripción del proceso Asignación de combustible

El especialista general del área de Portadores energéticos de la dirección de transporte de la UCI es la persona encargada de realizar la asignación de combustible de los vehículos. Para esta operación él entrega los documentos anexos de combustible y certificación de combustible habilitado y kilómetros recorridos a los choferes mensualmente para que vayan a la Caja en el área de Economía a recoger la tarjeta de combustible. En función de la información registrada en estos documentos al final del mes el especialista realiza el análisis de eficiencia teniendo en cuenta los estados (Correcto, Distorsionado mayor que 5 y Distorsionado menor que 5) que se determinan a partir de la fórmula: $((\text{Distancia recorrida}/\text{índice consumo plan}) - \text{consumo combustible}) / \text{distancia recorrida}/\text{índice consumo plan} * 100$, teniendo presente los siguientes indicadores para determinar el estado:

- Si el valor obtenido está entre 5 y -5 el estado es Correcto.
- Si el valor obtenido es mayor que 5 es Distorsionado mayor que 5.
- Si el valor obtenido es menor que 5 es Distorsionado menor que 5

En función de estos indicadores el especialista decide si es necesario realizar la prueba del litro (medición de la cantidad de kilómetros recorridos por litro de combustible) al vehículo para verificar cuántos kilómetros realmente recorren por unidad de combustible gastado.

Para obtener mayor información sobre la descripción de este proceso (consultar Anexo 2 y 4 y documento entregable CIG-CFM-N-MTTO-i1508.doc).

Tabla 3 Artefactos de entrada y salida del proceso Asignación de combustible.

Proceso Asignación de combustible

Artefactos de Entrada	Artefactos de Salida
	Anexo de combustible
	Certificación de combustible habilitado y kilómetros recorridos
	Acta de responsabilidad material
	Tarjeta de combustible

2.2.3. Patrones de control de flujo

Los patrones de control de flujo que se utilizaron en la modelación de los procesos identificados son:

Secuencia: Es el patrón básico de todo el flujo de trabajo utilizado en todos los diagramas de la descripción de los procesos, para así poder ver la dependencia entre dos actividades. Ejemplo: no se puede realizar la entrega de la certificación de combustible habilitado y kilómetros recorridos sin antes haber realizado la entrega del anexo de combustible.

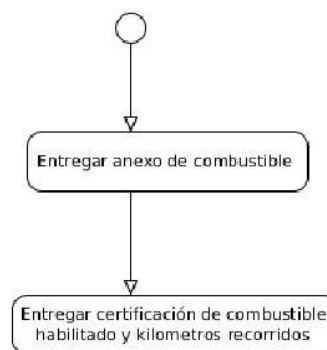


Imagen 1 Ejemplo de aplicación del Patrón de Secuencia.

Elección exclusiva: Se emplea cuando en un punto del trabajo se escoge solo una de varias ramas del proceso, es decir, en un momento del desarrollo del proceso de acuerdo a la condición que cumpla con los requerimientos adecuados toma un camino u otro el proceso tratado en cuestión. Ejemplo: si el

vehículo es alto consumidor o muy eficiente se procede a realizar la prueba del litro, en caso contrario se realiza el cierre de consumo y la entrega del anexo de combustible.

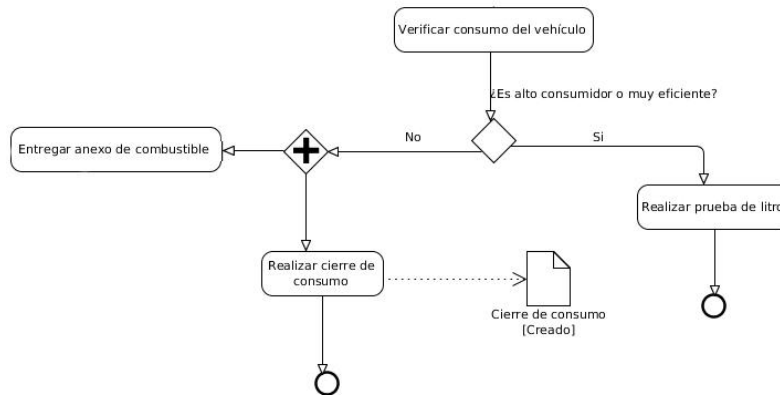


Imagen 2 Ejemplo de aplicación del Patrón de Selección exclusiva.

División paralela: Indica puntos del proceso en el cual las actividades pueden ser llevadas a cabo en forma concurrente (paralela). Ejemplo: en el proceso de Asignación de combustible se realizan en paralelo el cierre de consumo y la entrega del anexo de combustible.

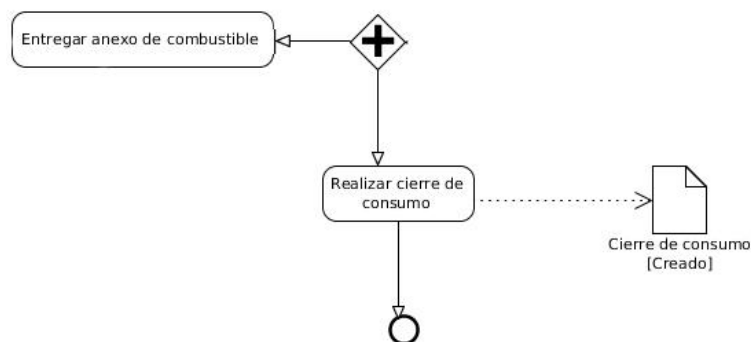


Imagen 3 Ejemplo de aplicación del Patrón de Selección Paralela.

2.2.4. Modelo conceptual

Capítulo 3: Diseño, Implementación Y Pruebas

El modelo conceptual es una representación del sistema mediante definiciones organizadas en forma estructurada, está conformado por conceptos que son relevantes para el área que se analiza, así como la relación entre ellos.

El modelo conceptual del proceso Control y análisis de los índices de combustible fue confeccionado con todas las clases conceptuales identificadas en este proceso junto con los atributos y relaciones existentes entre las mismas, para más información (consultar Imagen 4 y documento entregable CIG-CFM-N-MTTO-002.doc).

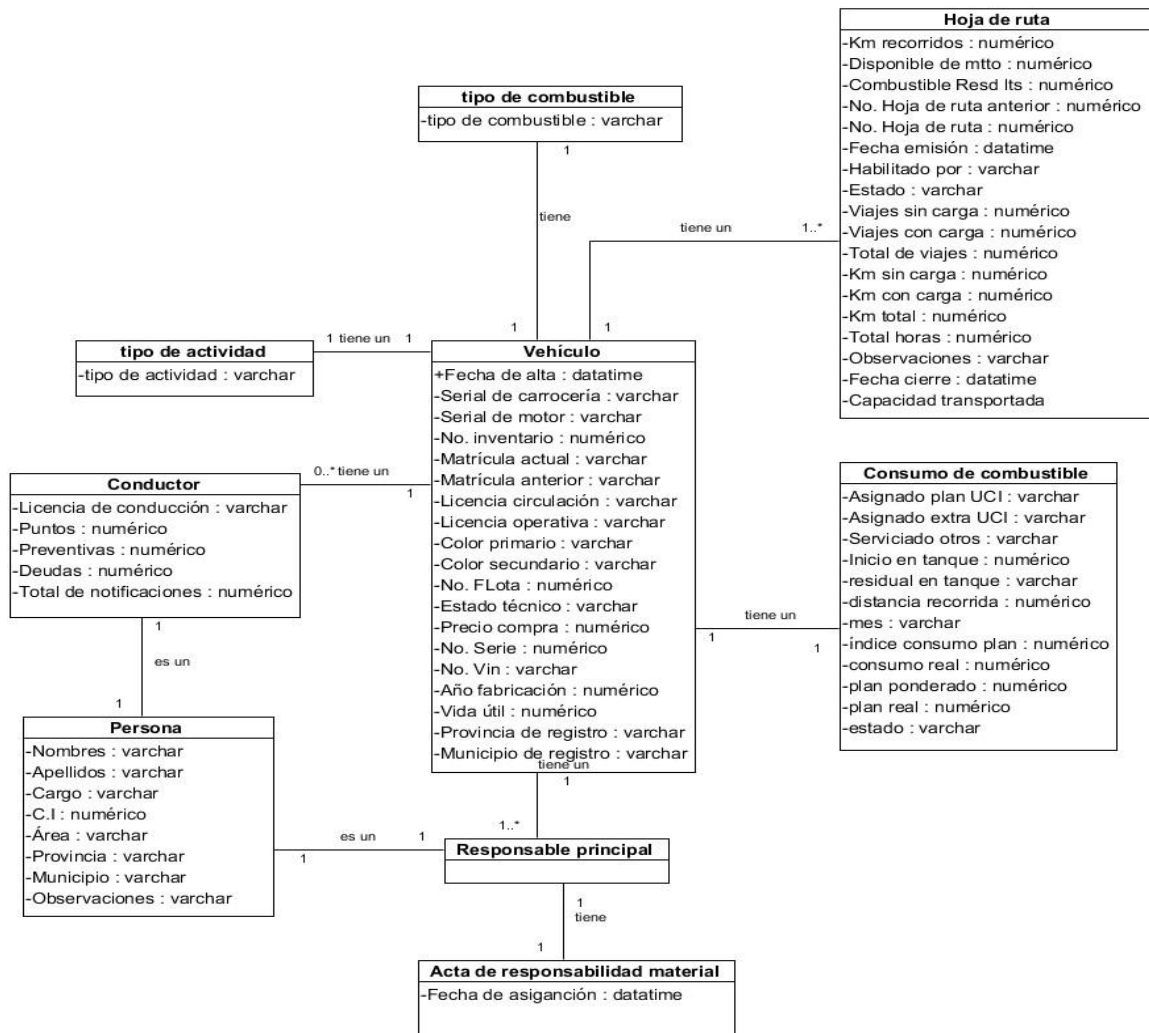


Imagen 4 Modelo conceptual.

2.2.5. Validación del modelado de negocio

La validación de los artefactos generados como parte de la modelación del proceso de negocio se realizó aplicando la técnica revisión técnica formal, donde los clientes revisaron cada documento generado con el objetivo de validar su correcta elaboración y que el flujo de actividades de cada proceso estuviera en correspondencia con la información brindada. Como constancia de esta revisión se emite un Acta de aceptación (Consultar Anexo 5).

2.3. Requerimientos

Un requerimiento puede definirse como un atributo necesario dentro de un sistema, que puede representar una capacidad, una característica o un factor de calidad del sistema de tal manera que le sea útil a los clientes o a los usuarios finales (Sommerville, 2005).

Los requerimientos se clasifican en dos categorías:

Requerimientos Funcionales: definen las acciones o funciones que el sistema debe ser capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Es importante que se describa el ¿Qué? y no el ¿Cómo? se deben hacer esas transformaciones (Sommerville, 2005).

Requerimientos No Funcionales: tiene que ver con características que de una u otra forma pueden limitar el sistema, como por ejemplo rendimiento, fiabilidad y seguridad (Sommerville, 2005).

La captura de requisitos es la actividad mediante la que el equipo de desarrollo de un sistema de software extrae, de cualquier fuente de información disponible, las necesidades que debe cubrir el sistema. El proceso de captura de requisitos puede resultar complejo, principalmente si el entorno de trabajo es desconocido para el equipo de analistas, y depende mucho de las personas que participen en él. Por la complejidad que todo esto puede implicar, la ingeniería de requisitos ha trabajado desde hace años en desarrollar técnicas que permitan hacer este proceso de una forma más eficiente y precisa (Escalona, et al., Diciembre de 2002).

2.3.1. Técnicas para la captura de requisitos

Para realizar la captura de los requisitos se hizo necesaria la utilización de algunas técnicas existentes con este fin. Entre las técnicas utilizadas se encuentran:

Entrevista: A través de encuentros planificados con especialistas del área de Transporte de la UCI se realizaron preguntas relacionadas con el proceso Control y análisis de índices de consumo de combustible en función de obtener las necesidades que tenían de informatización para la mejora de dicho proceso. Esto proporcionó como resultado que fueran identificados los requerimientos por los que se guiaría el desarrollo del módulo, a través del cual se realizaría la gestión de este proceso.

JAD: El equipo de desarrollo junto con especialistas del área de Transporte de la UCI, fueron revisando la documentación de los requerimientos a medida que iban siendo identificados los requisitos, con el objetivo de chequear que las especificaciones realizadas sobre el proceso Control y análisis de índices de consumo de combustible satisficieran las necesidades de informatización de la dirección de transporte de la UCI, dando como resultado que se documentaran los requisitos que dieran cumplimiento a lo descrito por estos.

Tormenta de ideas: En cada encuentro fueron debatidos los requerimientos identificados, fluyendo variadas ideas acerca de los mismos, lo cual arrojó como resultado que se obtuviera una visión más amplia de lo que se quería implementar, favoreciéndose de esta forma el avance de futuras etapas en el desarrollo del módulo, a través del cual se llevaría a cabo la gestión del proceso Control y análisis de índices de consumo de combustible.

2.3.1.1 Requisitos funcionales

Los requisitos funcionales identificados del proceso Control y análisis de índices de consumo de combustible del Sistema de Control de Flota y Mantenimiento de la dirección de transporte de la UCI son:

RF 1. Agrupación de requisitos: Gestionar tipo de combustible.

RF 1.1 Adicionar tipo combustible.

RF 1.2 Modificar tipo combustible.

RF 1.3 Eliminar tipo combustible.

RF 1.4 Listar tipo combustible.

RF 2. Agrupación de requisitos: Gestionar índices de consumo de combustible.

RF 2.1 Listar índices de consumo de combustible.

RF 2.2 Registrar índices de consumo.

RF 2.3 Modificar índices de consumo.

Reportes o salidas del sistema:

RF 3 Análisis de consumo por tipo de combustible.

RF 4 Análisis de consumo por tipo de actividad.

RF 5 Análisis de consumo por tipo de vehículo.

RF 6 Consumo de combustible por tipo de vehículo en un período.

RF 7 Intensidad energética por tipo de vehículo.

RF 8 Rendimiento energético por tipo de vehículo.

RF 9 Comportamiento del consumo de combustible en un año.

RF 10 Comportamiento del consumo de combustible en un mes.

2.3.2. Especificaciones de los requisitos funcionales

La especificación de los requisitos identificados se realizó siguiendo los pasos descritos en la plantilla conformada por el CEIGE.

Tabla 4 Especificación del requisito Registrar índices de consumo.

Precondiciones	El usuario ha sido validado.
Flujo de eventos	
Flujo básico	
1	Se selecciona el vehículo.
2	Se introducen los datos del consumo de combustible: Asignado plan UCI Asignado extra UCI Serviciado otros Inicio en tanque Residual en tanque Distancia recorrida Mes Índice de consumo plan
3	El sistema valida (ver validación 1) los datos introducidos.
4	Si los datos son correctos el sistema los registra.
5	El sistema confirma el registro de los datos.
6	Concluye el requisito.

Capítulo 3: Diseño, Implementación & Pruebas

Pos-condiciones		
1	Se registró en el sistema un nuevo índice de consumo de combustible para un vehículo.	
Flujos alternativos		
Flujo alternativo 4 .a Información errónea.		
1	El sistema señala los datos erróneos y permite corregirlos.	
2	El usuario corrige los datos.	
3	Volver al paso 3 del flujo básico.	
Pos-condiciones		
1	NA	
Flujo alternativo 4.b Información incompleta.		
1	El sistema señala los datos vacíos y permite corregirlos.	
2	El usuario corrige los datos.	
3	Volver al paso 3 del flujo básico.	
Pos-condiciones		
1	NA	
Flujo alternativo *.a El usuario cancela la acción.		
1	Concluye el requisito.	
Pos-condiciones		
1	NA	
Validaciones		
1	Se validan los datos según lo establecido en el Modelo conceptual CIG-CFM-N-MTTO-002.doc.	
Relaciones	Requisitos Incluidos	NA
	Extensiones	NA
Conceptos	Índices de consumo	de Visibles en la interfaz: Asignado plan UCI Asignado extra UCI Serviciado otros Inicio en tanque Residual en tanque Distancia recorrida Mes Utilizados internamente: NA
Requisitos especiales	NA	
Asuntos pendientes	NA	

Prototipo elemental de interfaz gráfica de usuario.

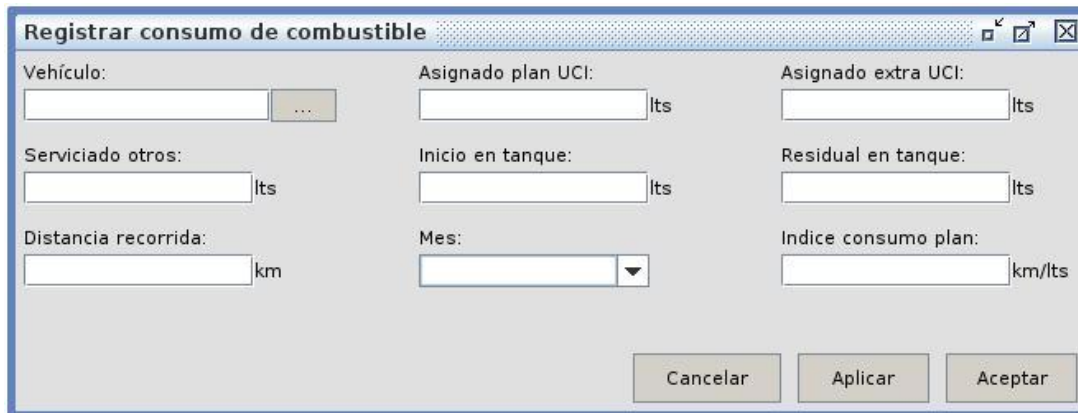


Imagen 5 Registrar consumo de combustible.

El resto de los requerimientos de la agrupación de requisitos Gestionar índices de consumo de combustible se encuentran en el documento entregable CIG-CFM-N-MTTO-i2601.doc y los restantes en los documentos entregables CIG-CFM-N-MTTO-i2604.doc, CIG-CFM-N-MTTO-i2901.doc, CIG-CFM-N-MTTO-i2902.doc, CIG-CFM-N-MTTO-i2903.doc, CIG-CFM-N-MTTO-i2931.doc, CIG-CFM-N-MTTO-i2930.doc, CIG-CFM-N-MTTO-i2934.doc, CIG-CFM-N-MTTO-i2933.doc.

2.3.3. Administración de los requisitos

La administración de requerimientos comprende las actividades relacionadas con la definición, clasificación, asignación, seguimiento y control de los requisitos durante todo el ciclo de vida de desarrollo de software. Para la administración de los requisitos del proceso Control y análisis de índices de consumo de combustible del Sistema de Control de Flota y Mantenimiento de la dirección de transporte de la UCI se utilizó la herramienta Visual Paradigm, para ello se definieron los siguientes elementos de trazabilidad:

- Requisitos.
- Modelo conceptual (entidades del negocio).
- Componentes.
- Diseños de casos de pruebas.

Se realizó el diagrama de requisitos con el objetivo de definir la relación entre los elementos de trazabilidad mencionados anteriormente (Consultar Anexo 7).

Se realizaron las siguientes matrices de trazabilidad:

- Requisitos Modelo Conceptual (entidades del negocio) (Consultar Anexo 8).
- Requisitos Componente (Consultar Anexo 9).
- Requisitos Diseños de casos de pruebas (Consultar Anexo 10).

2.3.4. Priorización de requisitos

Debido a limitaciones de tiempo y personal resulta difícil satisfacer o aplicar todos los requerimientos que se han levantado o exigido para la construcción o mantenimiento de un sistema, por lo que se hace necesario definir cuáles son los requerimientos a implementar de acuerdo a una previa priorización y en mutuo acuerdo con el cliente. Uno de los objetivos de dicha estrategia es determinar cuáles de los requerimientos son los principales a abordar para su desarrollo durante el ciclo de vida del respectivo producto (Estudios del Standish Group, 1997-2000).

La priorización de los requisitos identificados para el proceso Control y análisis de índices de consumo de combustible se realizó mediante la plantilla Evaluación de requisitos (consultar documento entregable CIG-CFM-N-MTTO-i6101.xls) realizada por el CEIGE y teniendo en cuenta los criterios definidos para determinar la complejidad de los requisitos y en función de esto la priorización teniendo mayor prioridad aquellos cuya complejidad sea Alta.

Indicadores definidos:

- **Diferentes comportamientos:** Un mismo requisito se comporta de manera diferente ante determinadas situaciones.
- **Consultas a fuentes de almacenamientos:** Los requisitos pueden presentar diversidad en la cantidad y complejidad de la interacción con la fuente de datos (Base de datos, Ficheros, otros).
- **Restricciones de validación:** Complejidad de todas las validaciones que lleve un requisito, tanto las validaciones en el lado del cliente, como en el servidor.
- **Grado de reutilización:** Complejidad de un requisito, para poder ser reutilizado por otros.
- **Lógica de negocio:** Los requisitos pueden presentar diferentes niveles de complejidad para la implementación de la lógica de negocio que contienen; Ej. Operaciones y métodos matemáticos.

La evaluación de los requisitos teniendo en cuenta los indicadores antes definidos arrojó los siguientes resultados:

Resumen	
Cantidad. de requisitos con complejidad Alta	10
Cantidad. de requisitos con complejidad Media	4
Cantidad. de requisitos con complejidad Baja	0

Para obtener más información (consultar Anexo 11 y documento entregable CIG-CFM-N-MTTO-i6101.doc).

2.3.5. Validación de requisitos

La validación de los requisitos se realiza con la finalidad de comprobar que los requerimientos identificados sean precisos, consistentes, realistas, verificables, definan lo que el usuario desea del producto final, que los errores que hayan sido detectados sean corregidos y el resultado del trabajo cumpla con los estándares establecidos para el proceso (Pressman, 2005).

Para la validación de los requisitos fueron aplicadas las siguientes técnicas:

Revisión técnica formal: La revisión de las especificaciones de los requerimientos del proceso Control y análisis de índices de consumo de combustible es realizada con especialistas del área de Transporte de la UCI. Con la utilización de esta técnica se validó que no existieran errores en el contenido o malas interpretaciones, información incompleta, inconsistencias y que los requisitos no fueran contradictorios, imposibles o inalcanzables, dando como resultado que fueran aprobados los que estaban descritos de forma correcta, clara y consistente. Como constancia de esta revisión se emite un Acta de aceptación (consultar Anexo 6).

Prototipos: A partir de las especificaciones de requisitos funcionales fueron conformados prototipos de interfaz de usuario. Con esto se validó que los requerimientos estaban en concordancia con las necesidades plasmadas por los especialistas del área de Transporte de la UCI. El empleo de esta técnica ofreció como resultados que el cliente tuviera una idea de la estructura de la interfaz de usuario y se

favoreciera la comunicación con el mismo, ya que tenía una visión inicial del módulo a través del cual se gestionaría el proceso Control y análisis de índices de consumo de combustible.

Generación de casos de prueba (test de requisitos): Fueron definidos y diseñados casos de pruebas para cada requerimiento especificado, con el objetivo de verificar el cumplimiento de los mismos. La utilización de esta técnica ofrece los siguientes resultados: son identificados los posibles escenarios de los requisitos, así como los juegos de datos de los campos determinados en estos, se validan y aprueban los que estaban bien enunciados, descritos y consistentes.

Liberación por el grupo de Calidad del CEIGE: Los requisitos identificados fueron validados en 3 iteraciones, revisando elementos como ortografía, formato y aspectos técnicos, emitiendo un Acta de liberación (consultar documento entregable Acta de Liberación de ER Mantenimiento).

2.4. Requisitos no funcionales

Los requisitos no funcionales del Sistema de Control de Flota y Mantenimiento de la dirección de transporte de la UCI se identificaron por el Analista principal del proyecto para todos los módulos del sistema, incluyendo el de Control de combustible.

Algunos de los requisitos no funcionales identificados:

Usabilidad:

1. El idioma de todas las interfaces de la aplicación será el español.
2. El sistema será consistente en el uso de abreviaturas, usará la misma abreviación siempre para la misma palabra y nunca en un elemento de selección o menú.
3. Los flujos de navegación para la gestión de cualquier concepto del negocio no excederán las 3 interfaces.
4. Los campos de texto tendrán un tamaño estándar de acuerdo con el espacio con que se cuente en el área de la página y en la medida que se llene esa área primaria se agregará la barra de desplazamiento vertical.
5. No se utilizarán textos extensos (no más de 9 caracteres) para las etiquetas de la interfaz de usuario, en su lugar se usan íconos para actividades típicas (Adicionar, Modificar, Eliminar). Los

íconos previstos para dichas actividades quedarán plasmados en el Manual de pautas de diseño.

6. El sistema contará con un Manual de usuario.

Fiabilidad:

1. El sistema estará disponible durante 8 horas, los 7 días de la semana, los 365 días del año.
2. El sistema tendrá un respaldo de la información en un servidor aparte del concebido para el almacenamiento regular de los datos, permitiendo la recuperación ante la pérdida parcial o total de la información.

Seguridad:

1. El sistema manejará la seguridad de acceso y administración de usuarios mediante el otorgamiento de privilegios y roles, asignación de perfiles.
2. Se concederá acceso al sistema a partir de un nombre de usuario y una contraseña.
3. El sistema concederá acceso a cada usuario autenticado solo a las funciones que le estén permitidas, de acuerdo a la configuración del sistema. Los menús serán generados a partir del propio perfil del usuario.

Para obtener mayor información sobre el resto de los requisitos no funcionales identificados (consultar documento entregable CIG-CFM-N-MTTO-i2301.doc).

2.5. Conclusiones parciales

La revisión técnica formal realizada permitió comprobar que la elaboración de los artefactos generados como parte de la modelación del proceso de negocio del módulo Control de combustible es correcta y que el flujo de actividades de cada proceso coincide con la información brindada. A partir de la aplicación de técnicas para la validación de los requisitos funcionales se logró probar que los requisitos identificados estaban descritos de forma correcta, clara y consistente, además que corresponden con las necesidades plasmadas por el cliente. Como constancia de las revisiones realizadas fue emitida un Acta de aceptación para cada una de las validaciones efectuadas.

CAPÍTULO 3: DISEÑO, IMPLEMENTACIÓN Y PRUEBAS

3.1. Introducción

En este capítulo se describe el diseño de la propuesta de solución, se explican los mecanismos de diseño utilizados, los diagramas de clases y diagramas de secuencia conformados a partir de los artefactos generados en el análisis y los patrones empleados en la modelación. A través de métricas es validado el diseño propuesto, con el objetivo de comprobar su correcta elaboración. Se muestra el modelo de datos, diagrama de componentes y el modelo de despliegue realizado para el sistema. Se especifican los estándares de codificación empleados para una mayor legibilidad en el código. Para finalizar se describen las pruebas de caja negra y caja blanca realizadas al software y los resultados obtenidos, en aras de validar que el sistema desarrollado posea la calidad requerida.

3.2. Diseño

El diseño es el sitio donde manda la creatividad, donde los requisitos del cliente, las necesidades de negocio y las consideraciones técnicas se unen en la formulación de un producto o sistema. Crea una representación o modelo del software, pero a diferencia del modelo de análisis, el modelo de diseño proporciona detalles acerca de las estructuras de datos, la arquitectura, las interfaces y los componentes del software que son necesarios para implementar el sistema (Pressman, 2005).

3.2.1. Mecanismos de diseño

Los mecanismos de diseño se utilizan con el objetivo de abreviar los diagramas de clases. Cada diseñador establece sus propios mecanismos de diseño, teniendo siempre en cuenta los patrones y estilos seleccionados (Verdecia, 2013).

En el módulo Control de combustible del Sistema de Control de Flota y Mantenimiento de la dirección de transporte de la UCI se definieron los siguientes mecanismos:

Mecanismo de diseño para los nombres de las clases.

Los diagramas de clases del diseño del Sistema de Control de Flota y Mantenimiento se realizaron por agrupaciones de requisitos funcionales. En el caso de las agrupaciones de requisitos conformadas por adicionar, modificar, eliminar, listar y buscar se dividen en dos clases, una que agrupa los requisitos

adicionar, modificar y eliminar; a la cual se le denomina AME_Nombre de la clase; y otra clase que agrupa los requisitos listar y buscar a la cual se le pone el nombre LB_Nombre de la clase.

Mecanismo de diseño para las clases controladoras.



Imagen 8 Mecanismo de diseño para las clases controladoras.

Todas las clases controladoras definidas en el diseño propuesto heredan de la clase `ZendExt_Controller_Secure`, ya que en ella se incluyen numerosas funcionalidades comunes en todas las controladoras.

Mecanismo de diseño para las clases del dominio.

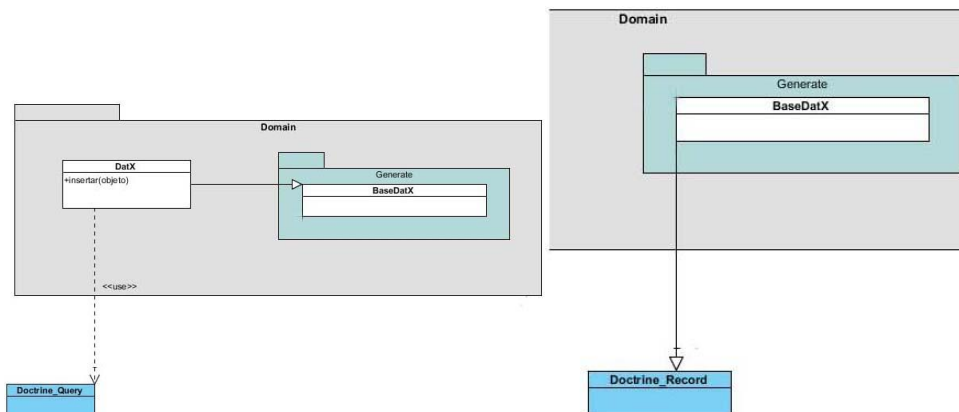


Imagen 10 Mecanismo de diseño para las clases del dominio.

La clase `DatX` representa a las clases del dominio (Domain), que usan a la Clase `Doctrine_Query` del marco de trabajo Doctrine, para el acceso a la base de datos a través del lenguaje DQL¹⁷; estas clases

¹⁷ Lenguaje de Consulta de Datos del inglés Data Query Language o

además heredan de las clases cuyo prefijo es Base con los atributos mapeados de las tablas de la base de datos; en este caso BaseDatX. Todas las clases con el prefijo Base como BaseDatX heredan de Doctrine_Record que permite agrupar en registros u objetos mapeados los datos de las tablas.

El resto de los mecanismos de diseño definidos se encuentran en el documento entregable CEIGE-Modelo de diseño.doc.

3.2.2. Diagramas de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces, contiene información sobre las clases, asociaciones, dependencias, métodos e interfaces (Valencia, 2003).

Para el desarrollo del módulo Control de combustible se realizaron diagramas de clases del diseño para los reportes (salidas del sistema) del módulo Control de combustible y las agrupaciones de requisitos Gestionar índices de consumo de combustible y Gestionar tipo de combustible (consultar Imagen 11, y el resto en el Anexo 12).

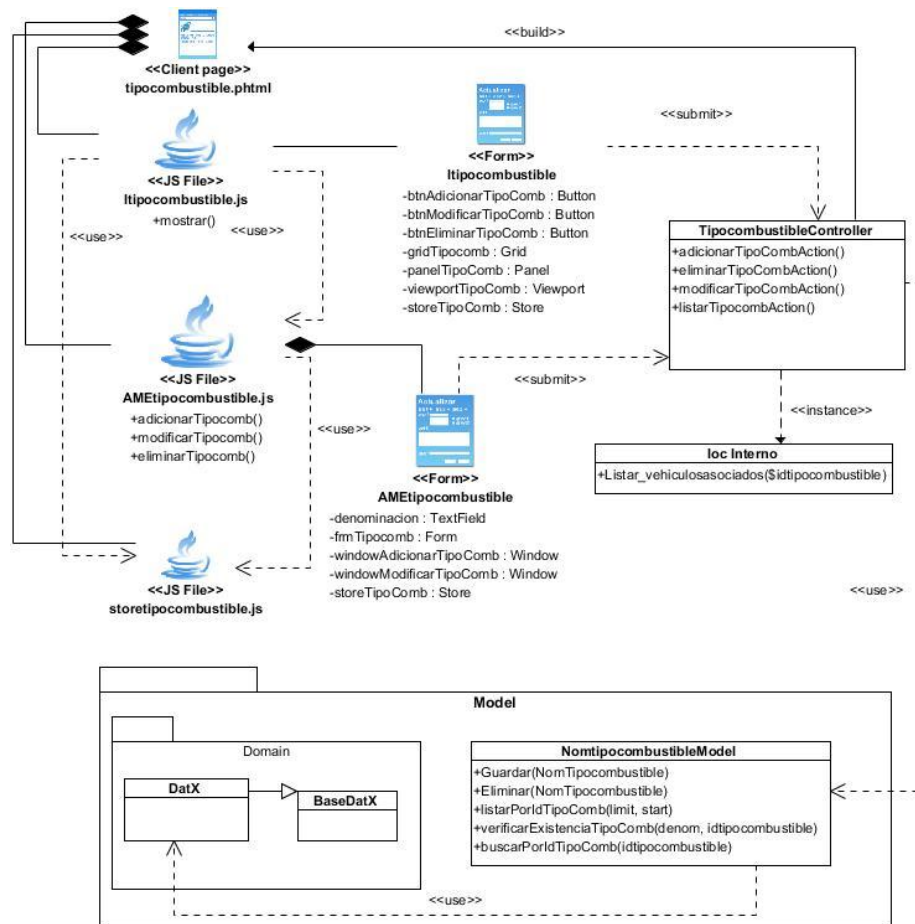


Imagen 11 Diagrama de clases de diseño de la agrupación de requisitos Gestionar tipo de combustible.

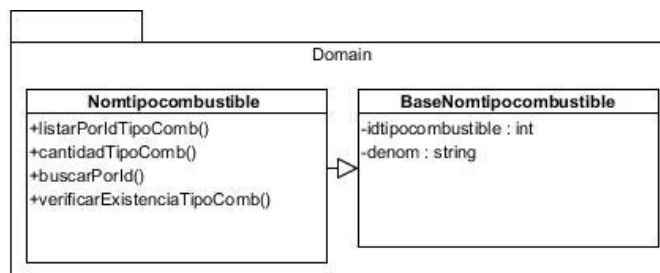


Imagen 12 Diagrama de clases de dominio de la agrupación de requisitos Gestionar tipo de combustible.

3.2.3. Descripción de las clases del diseño

A continuación se presenta la descripción de la clase `DatCombustibleModel`.

Tabla 5 Descripción de la clase `DatCombustibleModel`.

Nombre: <code>DatCombustibleModel</code>	
Tipos de clase: Modelo	
Para cada responsabilidad:	
Nombre	<code>Guardar(\$Combustible)</code>
Descripción	A partir de un arreglo pasado con los datos obtenidos de un combustible, realiza su adición al sistema, enviando estos a la base de datos utilizando la clase <code>DatCombustible</code> .
Nombre	<code>listarPorIdConsumoComb(\$limit,\$start)</code>
Descripción	Permite listar los datos del consumo de combustible que se encuentran en el sistema.
Nombre	<code>calcularIndiceConsComb(\$array)</code>
Descripción	Devuelve un listado de los índices de consumo de los vehículos calculados a partir de un arreglo que contiene la información de los consumos de combustible que se encuentran en la base de datos.
Nombre	<code>buscarPorAreaMatric(\$limit,\$start,\$matricula,\$area)</code>
Descripción	A partir de la matrícula de un vehículo y el área a la que pertenece devuelve un listado de los índices de consumo de combustible del vehículo.
Nombre	<code>verificarExistenciaConsumoComb(\$matricula,\$anno,\$mes,\$idcombustible)</code>

Capítulo 3: Diseño, Implementación & Pruebas

Descripción	Permite verificar a partir del identificador de combustible, el año y el mes en que se realizó la asignación y la matrícula del vehículo, si existe un consumo de combustible con esos mismos parámetros.
Nombre	imprimirAnalisisTipoComb(\$idtipocombustible,\$fecha)
Descripción	Obtiene los datos de los índices de consumo de combustible por tipo de combustible a partir de la fecha de registro y del identificador del tipo de combustible.
Nombre	imprimirAnalisisTipoAct(\$fecha,\$idtipocombustible)
Descripción	Obtiene los datos de los índices de consumo de combustible por tipo de actividad a partir de la fecha de registro y del identificador del tipo de combustible.
Nombre	imprimirAnalisisTipoVehiculo(\$idtipocombustible,\$fecha)
Descripción	Obtiene los datos de los índices de consumo de combustible por tipo de vehículo a partir de la fecha de registro y del identificador del tipo de combustible.
Nombre	imprimirAnalisisCombPorMes(\$idtipovehiculo,\$idtipocombustible,\$fecharegistro)
Descripción	Obtiene los datos del comportamiento del consumo de combustible en un mes a partir de la fecha del registro y del identificador del tipo de combustible y del vehículo.
Nombre	imprimirAnalisisIntensidadEnergetica(\$idtipovehiculo,\$fechainicio,\$fechafin)
Descripción	Obtiene la cantidad de combustible que se requiere para producir una unidad de tráfico.
Nombre	imprimirAnalisisCombPorAnno(\$idtipovehiculo,\$idtipocombustible,\$fecharegistro)

Descripción	Obtiene los datos del comportamiento del consumo de combustible en un año a partir de la fecha del registro y del identificador del tipo de combustible y del vehículo.
Nombre	imprimirAnalisisCombPorMes(\$idtipovehiculo,\$idtipocombustible,\$fecharegistro)
Descripción	Obtiene los datos del comportamiento del consumo de combustible en un mes a partir de la fecha del registro y del identificador del tipo de combustible y del vehículo.

3.2.4. Diagramas de secuencia

Un diagrama de secuencia es un diagrama de interacción que destaca el orden temporal de los mensajes, se utiliza para modelar los aspectos dinámicos de un sistema. Gráficamente, es una tabla que representa objetos, colocados a lo largo del eje X, y mensajes, ordenados según suceden en el tiempo, a lo largo del eje Y (Rojas, 2007).

A continuación se muestran un ejemplo de diagrama de secuencia, para ver el resto de los diagramas consultar Anexo 16.

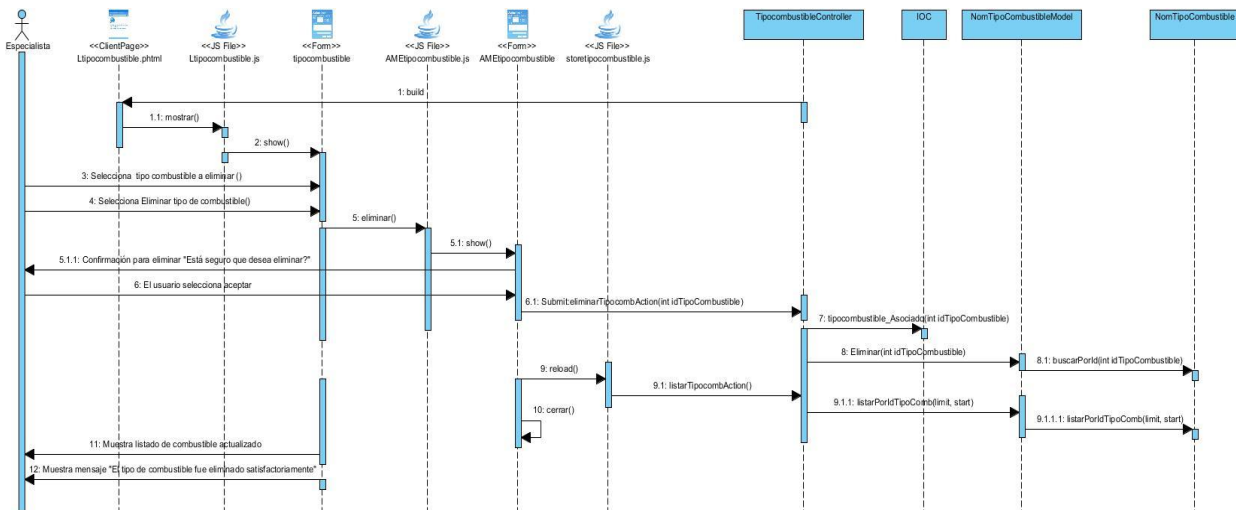


Imagen 12 Diagrama de secuencia del requisitos Adicionar tipo de combustible.

3.2.5. Patrones de diseño

Un patrón es una solución probada que se puede aplicar con éxito a determinados tipos de problemas que aparecen repetidamente en el desarrollo de sistemas software. Puede ser empleado muchas veces, en diferentes contextos, sin tener que duplicar el diseño. Se trata de un elemento de diseño que puede ser reutilizado (Pimentel, 2007).

Fachada:

La utilización de este patrón está reflejada en la creación y empleo de la clase IOC, ya que por la necesidad de integración entre los módulos del Sistema de Control de Flota y Mantenimiento, era necesaria la implementación de una clase fachada donde fueran publicados los servicios necesarios por estos, facilitando su interacción.

Solitario:

La utilización de este patrón se puede observar en la creación de una única instancia de la clase IOC que posee visibilidad en toda la aplicación, a través de la cual se acceden a las funciones que están declaradas en otros componentes. Por ejemplo para obtener los vehículos que poseen un tipo de combustible dado desde el componente Nomencladores, se accede mediante la instancia de la clase IOC a la funcionalidad Tipo_CombustibleAsociado(\$idtipocomb) que da solución a esta búsqueda.

Controlador:

El empleo de este patrón se puede observar en la creación de una clase controladora para la gestión de los tipos de combustible, el control y análisis de índices de consumo de combustible y para mostrar los reportes (salidas del sistema) relacionado con el área de transporte de la UCI.

Experto:

Fue usado en todas las clases definidas ya que estas contaban con la información necesaria para cumplir con la responsabilidad asignada.

Alta cohesión:

En la perspectiva del diseño orientado a objetos, la cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Su empleo se puede observar en el desarrollo del módulo Control de combustible ya que se le asignaron responsabilidades a las clases de forma tal que

cada una se encarga de realizar solamente las funciones que estén en correspondencia con su responsabilidad.

Bajo acoplamiento:

Un ejemplo donde se puede observar la utilización de este patrón es que la clase ReporteCombustibleController, se integra con el componente Nomencladores y no conoce la existencia de la clase Nomtipovehiculo, que se encuentra en dicho componente y contiene los datos que son solicitados, sino que interactúa con la fachada, encargada de pasarle los valores pedidos.

3.2.6. Métricas para evaluar el diseño propuesto

Las métricas del software son una medida cuantitativa de evaluar la calidad de los atributos internos de un sistema. Se emplean con el objetivo de llevar el control de la calidad del producto que se está desarrollando, evaluar la efectividad del proceso y mejorar la calidad del trabajo. Las métricas proporcionan los conocimientos necesarios para crear modelos efectivos de análisis y diseño, un código sólido y pruebas exhaustivas (Pressman, 2005).

Una vez realizado el diseño del módulo Control de combustible, se dio paso a su evaluación, para esto se utilizaron las métricas orientadas a objetos, específicamente a las clases, ya que las medidas y métricas para una clase individual, la jerarquía y las colaboraciones de estas permiten al ingeniero de software evaluar la calidad del diseño propuesto.

3.2.6.1. Métricas orientadas a objetos propuestas por Lorenz y Kidd:

En su libro sobre métricas, Lorenz y Kidd dividen las métricas basadas en clases en cuatro categorías, cada una con un diseño al nivel de componentes: tamaño, herencia, valores internos y valores externos. Las orientadas al tamaño aplicadas a una clase orientada a objetos se centran en el conteo de atributos y operaciones de una clase individual y los valores promedio para el sistema orientado a objetos como un todo (Pressman, 2005). Estas métricas se han introducido para ayudar a un ingeniero del software a usar el análisis cuantitativo para evaluar la calidad en el diseño antes de que un sistema se construya.

La métrica empleada fue **Tamaño de clase (TC)**, para aplicarla se tuvo en cuenta la cantidad de procedimientos que tenían las clases identificadas como parte de la propuesta de solución, luego se

realizó el cálculo del promedio de los procedimientos y mediante un criterio se obtuvo la categoría (baja, media, alta) para la Responsabilidad, Complejidad y Reutilización.

Tabla 6 Umbrales para la TC

Clasificación	Valores de los umbrales
Pequeño	\leq Promedio de operaciones(PO)
Medio	$>$ PO y $\leq 2*PO$
Grande	$>2*PO$

Tabla 7 Tamaño de las clases según sus atributos y operaciones

No	Nombre	Cantidad de atributos	Cantidad de operaciones	Tamaño
1	TipocombustibleController	0	5	Pequeño
2	CombustibleController	0	6	Pequeño
3	ReportecombustibleController	0	10	Medio
4	NomtipocombustibleModel	0	7	Pequeño
5	DatCombustibleModel	0	12	Medio

Tabla 8 Cantidad de clases por clasificación

Clasificación	Cantidad de clases	Responsabilidad de las clases	Complejidad de implementación de las clases	Reutilización
Pequeño	3	Baja	Baja	Alta
Medio	2	Medio	Medio	Medio
Grande	0	Alta	Alta	Baja

Tabla 9 Resultado general de la métrica

Cantidad	Cantidad	Cantidad de	Cantidad de	Promedio	Promedio
----------	----------	-------------	-------------	----------	----------

de clases	de clases pequeñas	clases grandes	clases medias	de atributos	de operaciones
5	3	0	2	0	9

Se hace un análisis de los resultados obtenidos en el empleo de esta técnica, se concluye que el diseño realizado es simple y tiene una calidad aceptable, pues la mayoría de las clases que fueron analizadas se encuentran dentro de la categoría de pequeñas (consultar Anexo 18), demostrándose en los pequeños valores de TC alcanzados que la implementación de forma general es sencilla, se disminuye en gran medida la responsabilidad de las clases lo que significa que las clases existentes dentro del sistema se puedan reutilizar ampliamente.

3.2.7. Modelo de datos

Un modelo de datos es la combinación de una colección de estructuras de datos, reglas, y convenciones que pueden utilizarse para describir un conjunto de datos y las operaciones para manipularlos. Constituye un elemento clave en el diseño de la arquitectura de un manejador de base de datos (Instituto Tecnológico de Colima, 2010).

En el modelo de datos elaborado para el módulo Control de combustible se representan las entidades, sus atributos y las relaciones entre ellas. Son las encargadas de almacenar los datos necesarios para que el módulo ofrezca las funcionalidades que fueron definidas durante el análisis. Para más información sobre el modelo de datos (consultar documento entregable CIG-CFM-N-MTTO-i3201.doc).

3.3. Implementación

3.3.1. Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones, contienen todos los componentes definidos en el subsistema, así como las distintas dependencias existentes entre ellos. Se realizan con el objetivo de obtener una vista general del sistema a partir de las dependencias e integraciones de los componentes.

A continuación se representa el diagrama general de componentes del Sistema de Control de Flota y Mantenimiento. Los elementos señalados con el color rojo en el diagrama son los que están relacionados

con la propuesta de solución, ya que las funcionalidades identificadas están incluidas dentro de los componentes Nomencladores y Vehículo, haciendo uso de los subsistemas Estructura y Composición y Seguridad (maneja la seguridad del sistema) y de las librerías TCPDF y Highcharts para la generación de los reportes o salidas del sistema.

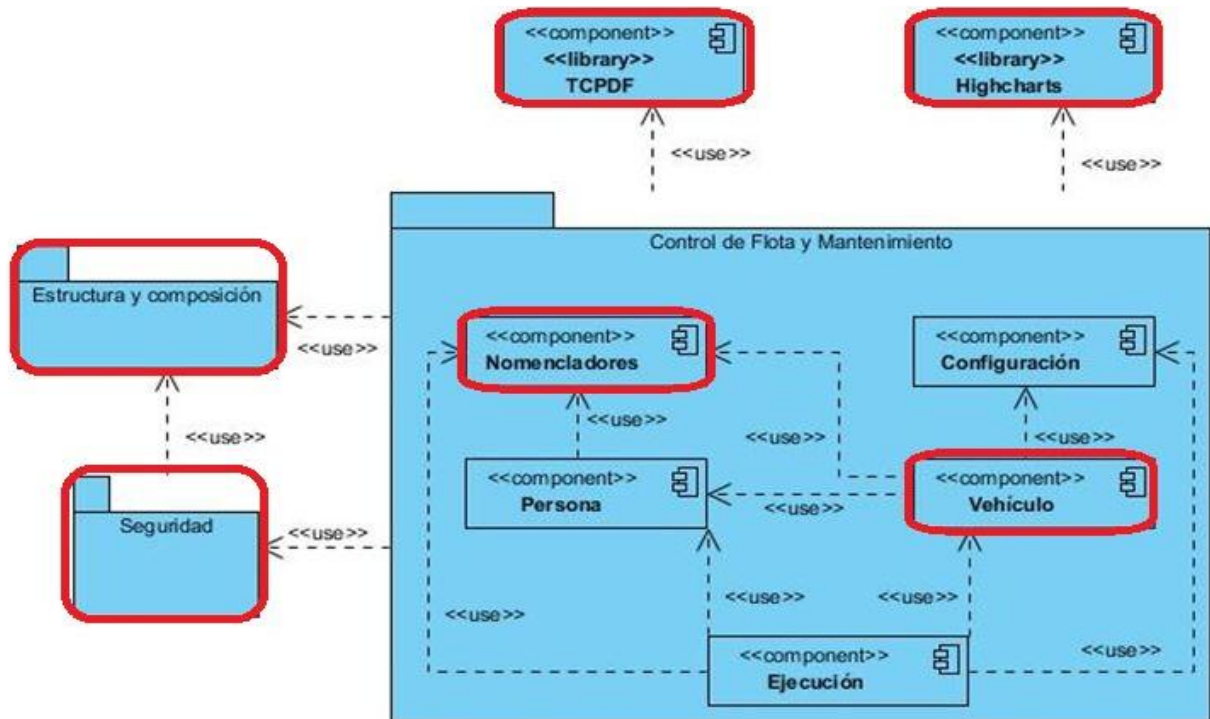


Imagen 18 Diagrama general de componentes.

3.3.2. Modelo de despliegue

Los diagramas de despliegue visualizan la disposición física de los distintos recursos computacionales que componen un sistema. Está compuesto por nodos, dispositivos y conectores. El propósito del modelo de despliegue es capturar la configuración de los elementos de procesamiento y las conexiones entre estos elementos en el sistema. Permite el mapeo de procesos dentro de los nodos, asegurando la distribución del comportamiento a través de aquellos nodos que son representados (Cintra, 2012).

A continuación se presenta el modelo de despliegue elaborado para el Sistema de Control de Flota y Mantenimiento de la dirección de transporte de la UCI:

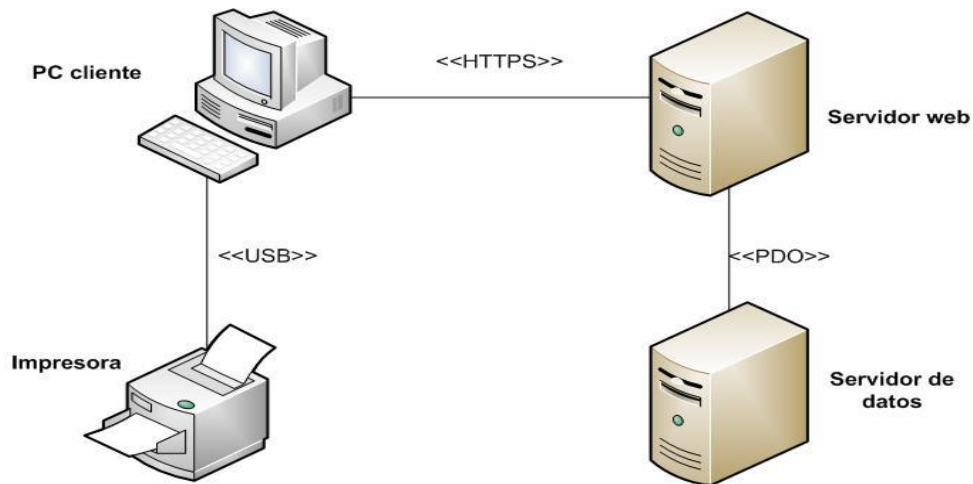


Imagen 19 Modelo de despliegue.

PC Cliente: Accede a la aplicación a través de un computador, donde es ejecutada mediante el navegador Mozilla Firefox versión 4.0 o superior, sobre cualquier sistema operativo.

Servidor Aplicaciones Web: El servidor de aplicaciones empleado donde radica la lógica de negocio de la aplicación es el Servidor Web Apache 2.2 utilizando biblioteca de clases adicionales: PHP 5.

Servidor de Base de datos: Sistema Gestor de Base de Datos: PostgreSQL 9.1 donde se encuentra la base de datos que utiliza el sistema.

Impresora: Utilizada para imprimir los reportes del Sistema de Control de Flota y Mantenimiento de la dirección de transporte de la UCI en caso de ser necesario.

3.3.3. Estándares de codificación

Un estándar de código se basa en la estructura y apariencia física de un programa, con el fin de facilitar la lectura, comprensión, mantenimiento del código y reutilización a lo largo del proceso de desarrollo de un software y no en la lógica del programa.

A continuación se especifican los estándares de codificación que fueron utilizados en el desarrollo del módulo Control de combustible:

Nomenclatura de las clases:

- Para las clases controladoras el nombre comenzará con la primera letra en mayúscula y el resto en minúscula, además estará dado según la función que realiza y se le adicionará al final “Controller”. Ejemplo: CombustibleController.
- El nombre de las clases del modelo que se encuentran dentro de la carpeta “business” comenzará con la primera letra en mayúscula y el resto en minúscula. A este nombre se le agregará al final “Model” y al inicio se le colocará “Nom”, en caso de que sea un nomenclador, de lo contrario será Dat, por ejemplo: DatCombustibleModel, NontipocombustibleModel.
- Las clases del dominio que se encuentran dentro de la carpeta “domain”, son generadas mediante un mapeo a la base de datos realizado por la herramienta Doctrine Generator v3.4, la cual fue desarrollada en el CEIGE y son nombradas igual que las tablas de la base de datos. Ejemplo: DatCombustible.
- Las clases contenidas en la carpeta “generated” son generadas mediante un mapeo a la base de datos realizado por la herramienta Doctrine Generator, la cual fue desarrollada en el CEIGE y delante del nombre se le coloca “Base”.
- El nombre de las clases de la vista comenzará con la primera letra en mayúscula y el resto en minúscula, en caso de que este sea compuesto, se utilizará la notación PascalCasing. A este nombre se le agregaran al principio, las letras en mayúscula de las funciones que se realizarán a través de estas clases. Ejemplo: LBCombustible (listar y buscar), AMCombustible (adicionar o asociar, y modificar).

Nomenclatura de los métodos o funciones

El nombre de los métodos o funciones de una clase comenzará en minúscula, en caso de que sea compuesto se utilizará la notación CamelCasing, o sea, seguido del primer nombre, comenzará el segundo con mayúscula. En caso de que sea una función de una clase controladora se le agregará al final la palabra “Action”. Ejemplo: listarConsumoCombAction, adicionarConsumoCombAction.

Nomenclatura de las variables

Las variables serán nombradas comenzando en minúscula, en caso de que el nombre de estas sea compuesto, se utilizará la notación CamelCasing, o sea, seguido del primer nombre, comenzará el

segundo en mayúscula. En la capa de la Vista, las variables que contengan componentes que formen parte de la interfaz, se les colocará delante un sufijo que indique el tipo de componente que contiene. Ejemplo: btnModificarIndice, storeCombustible, frmCombustible, gridCombustible.

3.3.4. Publicación de servicios entre componentes

Para cada uno de los componentes fueron publicados en sus clases Services, los mismos métodos que son utilizados en estos para la recuperación de los datos en las clases del modelo y del dominio, realizándose de esta manera con el objetivo de reutilizar código ya existente.

A continuación se muestran los servicios que son brindados por el componente Vehículo y que son utilizados en el desarrollo del módulo Combustible:

Tabla 10 Servicios que brinda Vehículo

Servicios	Componentes que lo utilizan	Descripción
listarPorTipoCombustible	Vehículo	Muestra un listado de todos los tipos de combustible que se encuentran registrados en el sistema.
listarTipoVehiculo	Vehículo	Muestra un listado de todos los tipos de vehículos que se encuentran registrados en el sistema.
tipoCombustibleAsociado	Vehículo	A partir del identificador de tipo de combustible verifica si existe al menos un vehículo que tenga asociado un tipo de combustible.
obtenerTipoCombustiblePorId	Vehículo	A partir del identificador de un tipo de combustible obtiene la denominación del mismo.

listarDependenciaPadreServ

Vehículo

A partir de un identificador obtiene las áreas que se encuentran registradas en el sistema.

3.4. Pruebas de software

3.4.1. Pruebas de caja blanca

La prueba de caja blanca también se conoce como prueba de caja transparente o de cristal consiste específicamente en cómo diseñar los casos de prueba atendiendo al comportamiento interno y la estructura del programa, examinándose la lógica interna sin considerar los aspectos de rendimiento (Pressman, 2005)

La prueba de caja blanca empleada en la solución fue la prueba del camino básico, para utilizarla se hace necesario el cálculo de la complejidad ciclomática del algoritmo que vaya a ser analizado, para lo cual se deben enumerar las sentencias de código y a partir de ahí elaborar el grafo de flujo de esta funcionalidad. Las sentencias enumeradas del método `adicionarConsumoCombAction()` se encuentran en el Anexo 17.

A continuación se muestra el grafo de flujo asociado al requisito funcional Registrar índices de consumo de combustible, seleccionado como caso de prueba debido a que tiene una complejidad alta y es una de las funcionalidades principales de la propuesta de solución:

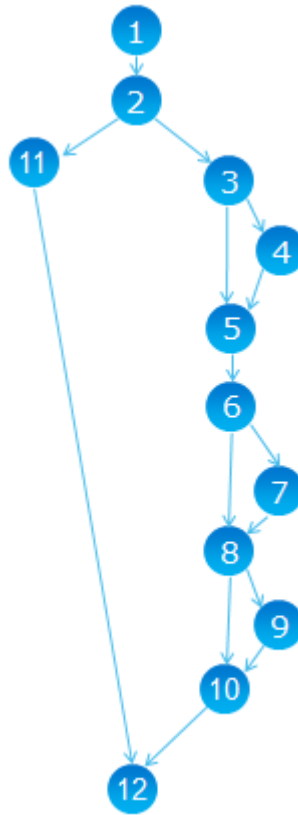


Imagen 20 Grafo de flujo asociado al algoritmo `adicionarConsumoCombAction()`

Cálculo de la complejidad ciclomática a partir de un segmento de código

La complejidad ciclomática es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa. El valor calculado como complejidad ciclomática define el número de caminos independientes del conjunto básico de un programa y ofrece un límite superior para el número de pruebas que se deben realizar para asegurar que sea ejecutada cada sentencia al menos una vez (Pressman, 2005). Al calcular la complejidad del método `adicionarConsumoCombAction()` fueron utilizadas las tres formas posibles con el objetivo de verificar que el cálculo se ha realizado correctamente y los resultados obtenidos en cada una era el mismo. Las fórmulas para realizar dicho cálculo son:

1. $V(G) = (A - N) + 2$, $V(G) = (15 - 12) + 2$, $V(G) = 5$

Siendo A la cantidad total de aristas del grafo y N la cantidad de nodos.

$$2. V(G) = P + 1, V(G) = 4 + 1, V(G) = 5$$

Siendo P la cantidad de nodos predicado (son aquellos de los cuales parten dos o más aristas).

$$3. V(G) = R, V(G) = 5$$

Siendo R la cantidad de regiones que posee el grafo.

Cada una de las fórmulas $V(G)$ representan el valor del cálculo. A partir de los resultados obtenidos en cada uno, se determina que la complejidad ciclomática del código analizado es 5, que a su vez es el número de caminos posibles a circular el flujo y el límite superior de casos de prueba que se le pueden aplicar a dicho código.

A continuación se muestran los caminos básicos por donde puede circular el flujo:

Tabla 111. Caminos básicos del flujo.

Número	Caminos básicos
1	1-2-11-12
2	1-2-3-4-5-6-8-10-12
3	1-2-3-5-6-7-8-10-12
4	1-2-3-4-5-6-8-9-10-12
5	1-2-3-4-5-6-7-8-9-10-12

Para cada uno de los caminos obtenidos se realiza un caso de prueba. Los casos de prueba realizados son los siguientes:

Caso de prueba para el camino básico # 1

Camino: 1-2-11-12

Descripción: Los datos de entrada cumplirán con los siguientes requisitos:

El parámetro \$registro, no está vacío, contiene todos los datos de un registro determinado, la fecha elegida indica que ya existe un registro de consumo de combustible para el vehículo seleccionado.

Entrada: \$registro= [[asigplanuci] => 5464, [asigextrauci] => 7894, [servotro] => 12, [iniciotanque] => 456, [restanque] => 5445, [distrec] => 2112, [indconsplan] => 1.7, [matricula] => HA123, [idvehiculo] => 90000000001]

Resultados esperados: Se espera que no adicione el registro de consumo de combustible y muestre un mensaje informando que no fue adicionado.

Resultados obtenidos: Satisfactorio.

Caso de prueba para el camino básico # 2

Camino: 1-2-3-4-5-6-8-10-12

Descripción: Los datos de entrada cumplirán con los siguientes requisitos:

El parámetro \$registro, no está vacío, contiene todos los datos de un registro determinado, excepto los datos del asignado extra UCI y otros serviciados y el índice de consumo plan es modificado.

Entrada: \$registro= [[asigplanuci] => 2589, [asigextrauci] =>, [servotro] => 23, [iniciotanque] => 215, [restanque] => 122, [distrec] => 2113, [indconsplan] => 1.7, [matricula] => HA123, [idvehiculo] => 90000000001]

Resultados esperados: Se espera que se actualice el índice de consumo plan, adicione el registro de consumo de combustible y muestre un mensaje informando que el registro fue adicionado satisfactoriamente.

Resultados obtenidos: Satisfactorio.

Caso de prueba para el camino básico # 3

Camino: 1-2-3-5-6-7-8-10-12

Descripción: Los datos de entrada cumplirán con los siguientes requisitos:

El parámetro \$registro, no está vacío, contiene todos los datos de un registro determinado, excepto los datos del asignado extra UCI.

\$registro= [[asigplanuci] => 2389, [asigextrauci] =>, [servotro] =>, [iniciotanque] => 231, [restanque] => 82, [distrec] => 1211, [indconsplan] => 4.7, [matricula] => HA123, [idvehiculo] => 90000000001]

Entrada:

Resultados esperados: Se espera que se adicione el registro de consumo de combustible y muestre un mensaje informando que el registro fue adicionado satisfactoriamente.

Resultados obtenidos: Satisfactorio.

Caso de prueba para el camino básico # 4

Camino: 1-2-3-4-5-6-8-9-10-12

Descripción: Los datos de entrada cumplirán con los siguientes requisitos:

El parámetro \$registro, no está vacío, contiene todos los datos de un registro determinado, excepto los datos de otros serviciados y el índice de consumo plan es modificado.

Entrada: \$registro= [[asigplanuci] => 1259, [asigextrauci] =>264, [servotro] =>, [iniciotanque] => 218, [restanque] => 542, [distrec] => 524, [indconsplan] => 4.8, [matricula] => HA123, [idvehiculo] => 90000000001]

Resultados esperados: Se espera que se actualice el índice de consumo plan, adicione el registro de consumo de combustible y muestre un mensaje informando que el registro fue adicionado satisfactoriamente.

Resultados obtenidos: Satisfactorio.

Caso de prueba para el camino básico # 5

Camino: 1-2-3-4-5-6-7-8-9-10-12

Descripción: Los datos de entrada cumplirán con los siguientes requisitos:

El parámetro \$registro, no está vacío, contiene todos los datos de un registro determinado y el índice de consumo plan es modificado.

Entrada: \$registro= [[asigplanuci] => 4364, [asigextrauci] => 894, [servotro] => 22, [iniciotanque] => 156, [restanque] => 445, [distrec] => 112, [indconsplan] => 3.7, [matricula] => HA123, [idvehiculo] => 90000000001]

Resultados esperados: Se espera que se actualice el índice de consumo plan, adicione el registro de consumo de combustible y muestre un mensaje informando que el registro fue adicionado satisfactoriamente.

Resultados obtenidos: Satisfactorio.

Con la aplicación de los casos de prueba expuestos anteriormente se comprobó que el flujo de trabajo de las funcionalidades es correcto, ya que se probó que cada sentencia es ejecutada al menos una vez con los parámetros de entrada y las salidas esperadas. Con las pruebas de caja blanca se comprobó la correcta implementación, se examinó el estado de la aplicación en varios puntos de la misma determinando que el estado real coincidía con el esperado.

3.4.2. Pruebas de caja negra

Las pruebas de caja negra se basan en la especificación del programa o componente a ser probado para elaborar los casos de prueba. El componente es visto como una “caja negra” cuyo comportamiento es desconocido y sólo puede ser evaluado estudiando sus entradas y las salidas obtenidas. También son conocidas como pruebas de comportamiento o pruebas inducidas por los datos.

Para la aplicación de este tipo de prueba se definieron casos de pruebas para cada uno de los requisitos funcionales, los cuales son los siguientes:

Condiciones de ejecución

- Se debe identificar y autenticar ante el sistema, además debe tener los permisos para ejecutar esta acción.
- Se debe seleccionar la opción del menú: **Mantenimiento/Combustible/ Control de combustible/ Registrar índices de consumo de combustible.**

Tabla 12 Descripción del caso de prueba para el requisito Registrar índices de consumo de combustible

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1. Registrar	Debe permitir	EP 1.1: Registrar índice de	– Se introducen los datos del índice de consumo de combustible

Capítulo 3: Diseño, Implementación Y Pruebas

índices de consumo de combustible.	registrar los índices de consumo de combustible en el sistema.	consumo de combustible introduciendo datos válidos.	correctamente. – Se presiona el botón Aceptar . – Se muestra un mensaje de información. – Se presiona el botón Aceptar .
		EP 1.2: Registrar índice de consumo de combustible introduciendo datos válidos presionando el botón Aplicar .	– Se introducen los datos del índice de consumo de combustible correctamente. – Se presiona el botón Aplicar . – Se muestra un mensaje de información. – Se presiona el botón Aceptar .
		EP 1.3: Registrar índice de consumo de combustible introduciendo datos inválidos.	– Se introducen los datos inválidos del índice de consumo de combustible – Se presiona el botón Aceptar . – Se muestra un mensaje de error. – Se presiona el botón Aceptar .
		EP 1.4: Registrar índice de consumo de combustible dejando campos vacíos.	– Se introducen los datos del índice de consumo de combustible dejando algún campo en blanco. – Se presiona el botón Aceptar . – Se muestra un mensaje informando del error. – Se presiona el botón Aceptar .
		EP 1.5: Cancelar.	– Se introducen o no los datos del índice de consumo de combustible. – Se presiona el botón Cancelar .

Para más información sobre el caso de prueba para el requisito Registrar índices de consumo de combustible, consultar el documento entregable CIG-CFM-N-MTTO-i5147.doc. Los casos de prueba elaborados para los restantes requisitos funcionales se encuentran en los documentos entregables: CIG-CFM-N-MTTO-i5146.doc, CIG-CFM-N-MTTO-i5148.doc, CIG-CFM-N-MTTO-i5153.doc, CIG-CFM-N-

MTTO-i5149.doc, CIG-CFM-N-MTTO-i5150.doc, CIG-CFM-N-MTTO-i5151.doc, CIG-CFM-N-MTTO-i5152.doc, CIG-CFM-N-MTTO-i5154.doc.

Con la aplicación de las pruebas de caja negra se comprobó el correcto funcionamiento de las funcionalidades implementadas en el módulo Control de combustible, ya que se obtuvieron los resultados esperados como respuesta del conjunto de condiciones de entradas definidos en cada diseño de caso de prueba y no se encontraron errores de interfaz, funcionalidades incorrectas o ausentes, errores en estructuras de datos o en acceso a la base de datos.

3.5. Conclusiones parciales

El uso de las métricas aplicadas al diseño para la evaluación de la solución propuesta arrojó como resultado que el diseño realizado es simple y presenta una calidad aceptable, debido a que el 60% de las clases resultaron ser pequeñas por lo que la implementación de forma general es sencilla, disminuyendo en gran medida la responsabilidad de las clases, por lo que se pueden reutilizar ampliamente. Las pruebas realizadas demostraron que el flujo de trabajo de las funcionalidades es correcto ya que se examinó el estado de la aplicación en varios puntos de la misma, determinando que el estado real coincidía con el esperado, además no se encontraron errores de interfaz, funcionalidades incorrectas o ausentes, errores en estructuras de datos o en acceso a la base de datos.

CONCLUSIONES

El desarrollo del módulo Control de combustible contribuyó a mejorar la gestión del proceso Control y análisis de índices de consumo de combustible de la dirección de transporte de la UCI ya que:

- Como parte del estudio del estado del arte realizado se analizaron los indicadores definidos en la norma cubana de febrero del 2011, siendo seleccionados la intensidad y el rendimiento energético como indicadores a implementar como parte de la propuesta de solución debido a que estos no estaban implementados en ninguno de los sistemas internacionales y nacionales analizados, representando esto una oportunidad para el sistema a implementar porque a partir de los resultados obtenidos con su aplicación mediante la generación de los reportes o salidas del sistema implementados que ilustran el comportamiento de estos indicadores, se pueden tomar decisiones en cuanto al consumo de combustible de los vehículos de la dirección de transporte de la UCI.
- La utilización de la biblioteca de clases Highcharts en la implementación de la propuesta de solución, permitió generar gráficas dinámicas que muestran el comportamiento del consumo de combustible en un período determinado, información esta que antes era imposible obtener debido a la forma en que se gestionaba el proceso Control y análisis de índices de consumo de combustible, permitiéndole de esta forma al especialista general tomar decisiones en cuanto al consumo de combustible de los vehículos de la dirección de transporte de la UCI.
- Con la implementación del módulo Control de combustible se puede obtener información relacionada con el análisis de índices de consumo de los vehículos de forma rápida mediante los reportes (salidas del sistema) identificados como parte de la propuesta de solución, utilizando la biblioteca de clases TCPDF para su impresión.
- El módulo implementado satisface las necesidades plasmadas por el cliente y presenta una calidad aceptable que se evidencia en los resultados obtenidos en las validaciones realizadas a cada uno de los artefactos generados en las etapas de modelación del negocio, análisis, diseño e implementación de la propuesta de solución, mediante el uso de técnicas, métricas y pruebas.

RECOMENDACIONES

Se recomienda que:

- El sistema incluya la generación de planes de combustible mensuales o anuales y realice las asignaciones de combustible a los vehículos de acuerdo al plan, para que el asignado plan UCI que se maneja en el requisito funcional Registrar índices de consumo de combustible salga de esta información y no se registre al final de cada mes al efectuar el cierre de consumo como se realiza en la propuesta de solución.
- Se implementen en otra versión del sistema los procesos que maneja el área de Economía relacionados con el control de las tarjetas de combustible.
- El presente trabajo de diploma sea utilizado como referencia para el desarrollo de sistemas de control de combustible que se desarrollen para el país.

BIBLIOGRAFÍA

ALBET, S.A. 2011. *Mantenimiento Vehicular, Manual de Usuario.* 2011.

Asuni, Nicola. 2001-2011. TCPDF PHP class for generating PDF documents. [En línea] 2001-2011. [Citado el: 29 de Marzo de 2013.] <http://www.tcpdf.org>.

Badillo, Camilo Esteban Mahecha. 2012. *Mapas De Procesos.* 2012. 298225.

Baryolo, Oiner Gómez. 2010. *Solución informática de autorización en entornos multientidad y multisistema.* . s.l. : Memoria individual presentada para optar por título de máster en informática aplicada, 2010.

Baryolo, Oiner Gómez, Borbón, Yoandry Morejón y Tejo, Darien García. 2010. *ARQUITECTURA TECNOLÓGICA PARA EL DESARROLLO DE SOFTWARE.* Ciudad de la Habana : s.n., 2010.

Bizagi. 2012. Bizagi. [En línea] Diciembre de 2012. [Citado el: 19 de Noviembre de 2012.] <http://www.buenastareas.com/ensayos/Bizagi/4235294.html>.

Blanco, Geidys Padrón y Iglesias, Ailec Rodríguez. 2009. *Procedimiento para la validación de requisitos de software.* La Habana, Cuba : Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas, 2009.

CEIGE. 2012. *MODELO DE DESARROLLO DE SOFTWARE.* 2012.

Cintra, Yordano Yunior Osorio. junio, 2012. *Sistemas para la gestión del banco de problemas de la facultad 3.* s.l. : Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas, junio, 2012.

Diez, A. 2001. *IRqA y el desarrollo de proyectos: Experiencias Prácticas.* s.l. : I Jornadas de Ingeniería de Requisitos, 2001.

Desoft, S.A. 2004. *Apolo Express 2004.* 2004.

Doctrine. 2012. Welcome to the Doctrine Project. [En línea] 18 de Enero de 2012. [Citado el: 20 de Noviembre de 2012.] <http://www.doctrine-project.org/>.

Energía, Agencia Andaluza de la. 2007. *Guía para la gestión del combustible en las flotas de transporte por carretera.* 2007.

Escalona, María José y Koch, Nora. Diciembre de 2002. *Ingeniería de Requisitos.* Sevilla : s.n., Diciembre de 2002.

Estudios del Standish Group. 1997-2000. TRIPOD. [En línea] 1997-2000. [Citado el: 18 de Febrero de 2013.] http://macroos0.tripod.com/n_de_requerimiento3.html.

EUGCOM. 2009. *Descripción del Producto ADMINISTRACIÓN Y CONTROL DE FLOTAS*. Chile : s.n., 2009.

Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad Michael Stal. 1996. *Pattern Oriented Software Architecture: A System of Patterns*. Inglaterra. 1996.

Fuentes, Lidia, Troya, José M. y Vallecillo, Antonio. 2001. *Desarrollo de Software Basado en Componentes*. Dept. Lenguajes y Ciencias de la Computación. Universidad de Málaga : s.n., 2001.

Grande, Laura Henche. 2009. *INTRODUCCIÓN A LA NOTACIÓN BPMN Y SU RELACIÓN CON LAS ESTRATEGIAS DEL LENGUAJE MAUDE*. 2009.

Gutiérrez, David. 2011. Desarrollo web y tecnología. [En línea] 28 de Julio de 2011. [Citado el: 16 de Noviembre de 2012.] http://vitaminaweb.com/que-es-php_863.

— . 2011. Desarrollo web y tecnología. [En línea] 28 de Julio de 2011. [Citado el: 19 de Noviembre de 2012.] http://vitaminaweb.com/que-es-php_863.

Heredia, Cristian Van Der y Herminio. 2001. Introducción al PHP. [En línea] 23 de Mayo de 2001. [Citado el: 23 de Noviembre de 2012.] <http://www.maestrosdelweb.com/editorial/phpintro/>.

Herrera, Érica Uribe, y otros. 2009. *SERVIDOR DE APLICACIONES*. 2009.

Inc, Zend Technologies. 2005-2011. Zend Framework 1.11 Manual. [En línea] 20 de Noviembre de 2005-2011. <http://doczf.mikaelkael.fr/1.11/es/introduction.overview.html>.

Instituto Tecnológico de Colima. 2010. *Tutorial de Fundamentos de bases de datos*. 2010.

Jacobson, Ivar, Booch, Grady y Rumbaugh, James. 2000. *El proceso Unificado ed Desarrollo de Software*. España: Addison Wesley : s.n., 2000.

Jesse James Garrett. 2005. Javalobby. Ajax: a new approach to web application. . [En línea] 18 de Febrero de 2005. [Citado el: 23 de Noviembre de 2012.] <http://www.javalobby.org/articles/ajax/>.

KreaSoft . 2010. KreaSoft Software y Tecnología. [En línea] 2010. <http://www.kreasoft.cl/kreaflota.html>.

Normalización, Oficina Nacional de. 2011. *Transporte automotor. Servicio de Transportación de pasajeros y cargas. Términos, definiciones, símbolos y métodos de cálculos*. 2011.

Paradigm, Visual. 2011. [En línea] 26 de Noviembre de 2011. [Citado el: 2 de Noviembre de 2012.] <http://www.visual-paradigm.com>.

Pérez, Javier Eguíluz. 2011. Introducción a JavaScript. [En línea] 10 de Noviembre de 2011. [Citado el: 23 de Noviembre de 2012.] <http://www.librosweb.es/javascript/>.

Pimentel, Ernesto. 2007. *Patrones de Diseño*. 2007. IDEAS. Isla de Margarita, Venezuela.

PostgreSQL, Grupo Global de Desarrollo de. 1996-2012. PostgreSQL 9.1 Press Kit. [En línea] 1996-2012. [Citado el: 25 de Noviembre de 2012.] <http://www.postgresql.org/about/press/presskit91.html.es>.

Practicopedia. 2011. Cómo son las mejoras y novedades en Firefox. [En línea] 29 de Marzo de 2011. [Citado el: 28 de Noviembre de 2012.] <http://internet-y-ordenadores.practicopedia.lainformacion.com/navegadores/como-son-las-mejoras-y-novedades-en-firefox-12866>.

Pressman, Roger S. 2005. *Ingeniería del Software Un enfoque práctico*. . s.l. : Sexta edición, 2005.

Profesorado, Instituto Nacional de Tecnologías Educativas y Formación del. 2008. Observatorio Tecnológico Ministerio de Educación, Cultura y Deporte. [En línea] Abril de 2008. [Citado el: 23 de Noviembre de 2012.] <http://recursostic.educacion.es/observatorio/web/eu/software/servidores/580-elvira-mifsud>.

Rill Arencibia, Edgar y Guerrero Mejías, Yensy Yoan. 2009. *Sistema informático para la administración de los Recursos Humanos. Módulo de selección de candidatos*. Ciudad de La Habana : s.n., 2009.

Rojas, Gonzalo. 2007. *UML Diagrama de Secuencia*. 2007.

2006. Scribd. [En línea] 10 de Octubre de 2006. [Citado el: 11 de Noviembre de 2012.] <http://www.scribd.com/doc/270431/Ingenieria-requerimientos>.

Sevilla, Centro Técnico de Informática. 2006. *El Framework de desarrollo del Consejo Superior de Investigaciones Científicas*. 2006.

Software Incremental, S.A. 2010. FlotaWeb. [En línea] 2010. [Citado el: 21 de Octubre de 2012.] <http://flotaweb.com/>.

Software, Puro. 2007. Puro Software. [En línea] 2007. [Citado el: 13 de Marzo de 2013.] <http://www.purosoftware.com/desarrollo-web-scripts-graficos-estadisticos/10-highcharts-charts-interactivos.html>.

Sommerville, Ian. 2005. *Ingeniería de Software*. s.l. : Séptima Edición, 2005. ISBN:84-7829-074-5.

Valencia, Maria Eugenia. 2003. *Diagramas de clases del diseño*. s.l. : Dpto. Ciencias de la Computación, 2003.

Verdecia, Ing. Pedro Manuel Alás. 2013. *CEIGE-Modelo de diseño*. 2013. CEIGE-N-00-i001.

Vera, Luciano. 2012. Diseño Luciano Vera. [En línea] 2012. [Citado el: 25 de Noviembre de 2012.]
<http://www.lucianovera.cl/superficie/?que-es-un-browser-o-navegador-web>.

Zend Technologies. 2012. Zend Framework 1.11. [En línea] 2012. [Citado el: 20 de Noviembre de 2012.]
<http://download.techworld.com/3249074/zend-framework-1113>.

