



Universidad de las Ciencias Informáticas

Facultad 1

SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE DE LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS.

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

Autores: Manuel Alejandro Valdivié Fernández
José Manuel Hernández Reyes

Tutores: MSc. Radel Calzada Pando
Ing. Marilys Valiente González

Co-tutores: MSc. Delly Lien González Hernández
Lic. José Manuel de León Cano

La Habana, Junio de 2014.



"La inteligencia consiste no sólo en el conocimiento, sino también en la destreza de aplicar los conocimientos en la práctica."

Aristóteles

DECLARACIÓN DE AUTORÍA

Declaración de Autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Autor: Manuel Alejandro Valdivié Fernández

Autor: José Manuel Hernández Reyes

Tutor: MSc. Radel Calzada Pando

Tutor: Ing. Marilys Valiente González

Cotutor: MSc. Delly Lien González Hernández

Cotutor: Lic. José Manuel de León Cano

Datos de Contacto

Autores:

- Manuel Alejandro Valdivié Fernández.
Universidad de las Ciencias Informáticas, La Habana, Cuba.
Email: mavaldivie@estudiantes.uci.cu.
- José Manuel Hernández Reyes.
Universidad de las Ciencias Informáticas, La Habana, Cuba.
Email: jhreyes@estudiantes.uci.cu.

Tutores:

- MSc. Radel Calzada Pando.
Universidad de las Ciencias Informáticas, La Habana, Cuba.
Email: rcalzada@uci.cu.
- Ing. Marilys Valiente González.
Universidad de las Ciencias Informáticas, La Habana, Cuba.
Email: valiente@uci.cu.

Co-tutores:

- MSc. Delly Lien González Hernández.
Universidad de las Ciencias Informáticas, La Habana, Cuba.
Email: delly@uci.cu.
- Lic. José Manuel León Cano.
Universidad de las Ciencias Informáticas, La Habana, Cuba.
Email: dleon@uci.cu.

De Manuel Alejandro

A mi mamá Gladys por ser una persona que siempre está conmigo incondicionalmente.

A mi papá Carlos Manuel por apoyarme en los momentos más importantes de mi vida.

A Pedro, Mora, Junet, Eduardo y Raciél por toda la ayuda que me brindaron a lo largo de la tesis.

A mis amigos Lianet, Yetmary, Alian, Eduardo Surí, Alberto, José Ángel.

A Mireya y su familia por ser mi familia también.

A mi familia por estar siempre presente y ayudarme a convertirme en un Ingeniero.

A mi compañero de tesis y amigo José Manuel por convertir el trabajo en momentos alegres.

A mis compañeros de aula Edwin, Maidelis, Yailin, Nolberto, Yansel, Fernando, Luizon y Sucl.

A los Pupos, a Yani, a Eric por siempre estar ahí.

A mi tutor Radel por el apoyo y la confianza que deposito en mí.

A mi tutora Marilys por toda la ayuda que nos brinda

A todos los que de alguna forma u otra me brindaron su ayuda.

De José Manuel

A mi madre y mi padrastro por el apoyo incondicional durante toda mi vida, por su amor, por ser mi fuente de inspiración diaria y por confiar en mí.

A mi familia en general, por su apoyo e inspiración.

A mi compañero de tesis y amigo Manuel Valdivié por no rendirse jamás.

A mi tutores Marilys Valiente y Radel Calzada por su apoyo, dedicación y confianza en estos meses de trabajo.

A la Universidad y su colectivo de profesores que me han formado durante estos cinco años.

A todas las amistades que he conocido en la escuela.

En fin, quiero agradecer a todas aquellas personas que de una forma u otra han contribuido al desarrollo de este trabajo de diploma y a mi formación tanto profesional como personal.

De Manuel Alejandro

En primer lugar quiero dedicarle mi trabajo de diploma a una persona muy importante para mí que fue por muchos años mi luz, mi guía, todo para mí, a mi abuela Fina, que aunque no esté aquí conmigo hoy, yo sé que me está viendo y sonriendo dando la mayor de las bendiciones, un beso para ti abuela.

A mis padres Gladys y Carlos Manuel por guiarme y apoyarme en todas mis decisiones, por confiar en mí y estar siempre que los necesité.

A todos los familiares que me han apoyado y querido, mis tíos, primos, hermanos.

A mis amistades, las de antes, de ahora y de siempre, gracias por todo y tanto

A todos los profesores que de una manera u otra, me han guiado en este camino que es el aprendizaje.

De José Manuel

A mis bisabuelos Emilia y Alipio...donde quiera que estén hoy pueden sentir orgullo de su bisnieta preferido, manolito. Gracias por darme todo sin recibir nada a cambio, los amo.

A mi madre Carmen Reyes Labrador por darme la educación que me ha permitido llegar hasta aquí, por formarme como un hombre de bien, por sus mimos y sus desvelos y por todos los sacrificios que ha tenido que hacer para que nunca me faltara nada y pudiera estudiar siempre sin ninguna preocupación.

A mi padrastro Idael Cruz Chala, sé que nunca le digo papá pero no es por no sentir que lo eres, has sido más que eso para mí, nunca lo dudes.

A mis hermanos Dayron, Yossel y Yossiel porque con cada sonrisa me dan fuerzas para continuar y me hacen ser mejor ser humano con la esperanza de poder ser su ejemplo, su guía.

Resumen

La optimización de los recursos energéticos a través de una correcta gestión de la energía, permitirá a las Instituciones, no sólo mejorar el comportamiento ambiental mediante la reducción de emisiones de gases de efecto invernadero, sino también, conseguir importantes reducciones en los costos económicos. En la Universidad de las Ciencias Informáticas (UCI) existe una Dirección de Transporte que brinda servicios de transportación al personal de la Universidad, que consta de una flota de equipos de transporte realizando diferentes actividades diariamente, ocasionando que se genere gran cantidad de información sobre el combustible que se distribuye y se consume por parte de estos equipos. Se realizó un estudio donde se determinó que era necesario implementar un sistema que controle los flujos de información relacionados con la distribución y consumo del combustible, para lograr inmediatez y retroalimentación en el proceso de gestión de la información en las áreas: Dirección de Transporte, Grupo de Gestión Energética y Vicerrectoría Económica en la UCI. Se realizó un estudio de las herramientas, metodologías y lenguajes de programación que permitieran la construcción de una aplicación acorde a las necesidades y condiciones tecnológicas existentes en las áreas involucradas. En el trabajo de diploma se propone desarrollar un sistema que permita lograr el control sobre la información del combustible que se genera alrededor de este proceso, y que incide directamente en el control de este recurso energético y en la disponibilidad de la información, generada en la ejecución de este proceso.

Palabras clave: combustible, recurso energético, flota de equipos de transporte.

ÍNDICE

INTRODUCCIÓN.....	1
CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE.	6
1.1.- INTRODUCCIÓN.....	6
1.2.- PRINCIPALES CONCEPTOS ASOCIADOS A LA GESTIÓN DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE.	6
1.2.1.- Recurso energético.....	6
1.2.2.- Combustible.....	6
1.2.3.- Gestión de distribución y consumo del combustible.....	6
1.2.5.- Sistema de gestión de flotas de equipos de transporte.....	7
1.3.- SISTEMAS DE GESTIÓN DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE A NIVEL INTERNACIONAL.....	7
1.4.- SISTEMAS DE GESTIÓN DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE A NIVEL NACIONAL	9
1.5.- ACTUALIDAD Y NECESIDAD DEL TRABAJO	11
1.6.- HERRAMIENTAS, TECNOLOGÍAS Y METODOLOGÍAS DE DESARROLLO DE SOFTWARE	11
1.6.1.- Metodología de desarrollo de software.....	11
1.6.2.- Herramientas y tecnologías.....	13
1.7.- NOTACIÓN DE MODELADO DE PROCESOS DE NEGOCIO (BPMN).....	22
1.8.- CONCLUSIONES	23
CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE...24	24
2.1.- INTRODUCCIÓN.....	24
2.2.- MODELADO DEL NEGOCIO	24
2.3.- MAPAS DE LOS PROCESOS	26
2.4.- PROPUESTA DEL SISTEMA	27
2.5.- REQUISITOS FUNCIONALES Y NO FUNCIONALES DEL SISTEMA.....	27
2.5.1.- Técnicas para la captura de requisitos.....	27
2.5.2.- Requisitos funcionales	28
2.5.3.- Requisitos no funcionales.....	28
2.5.4.- Validación de requisitos.....	30
2.6.- MODELO DE CASOS DE USO DEL SISTEMA	31
2.6.1.- Descripción de los actores.....	31
2.6.2.- Diagrama de casos de uso.....	31
2.6.3.- Descripción de casos de uso	34
2.7.- CONCLUSIONES	35
CAPÍTULO III: DISEÑO DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE.....36	36
3.1.- INTRODUCCIÓN.....	36
3.2.- MODELO DEL DISEÑO	36
3.2.1.- Diagramas de clases del diseño.....	36
3.2.2.- Diagramas de Interacción.....	37
3.2.3.- Descripción de las clases del diseño.....	37
3.2.4.- Modelo físico de la base de datos.....	38
3.2.5.- Descripción de las entidades	39
3.3.- MODELO DE DESPLIEGUE.....	40
3.3.1.- Descripción de los nodos.....	40
3.3.2.- Descripción de elementos e interfaces de comunicación.....	41
3.4.- ARQUITECTURA Y PATRONES DE DISEÑO UTILIZADOS.....	41
3.4.1.- Arquitectura de software.....	41

3.4.2.- Patrones de diseño empleados	43
3.5.- CONCLUSIONES	45
CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE	46
4.1.- INTRODUCCIÓN.....	46
4.2.- MODELO DE IMPLEMENTACIÓN.....	46
4.2.1.- Diagrama de componentes	46
4.2.2.- Estándar y estilo de codificación	47
4.4.- PANTALLAS PRINCIPALES DE LA APLICACIÓN	49
4.5.- PRUEBAS AL SISTEMA PROPUESTO.	51
4.6.- CONCLUSIONES	60
CONCLUSIONES.....	61
RECOMENDACIONES	62
REFERENCIAS BIBLIOGRÁFICAS.....	63
GLOSARIO DE TÉRMINOS.....	69
ANEXO	70

ÍNDICE DE FIGURAS

FIGURA 1: DIAGRAMA DE PROCESOS GENERAL DEL GRUPO DE GESTIÓN ENERGÉTICA

FIGURA 2: DIAGRAMA DEL PROCESO ASIGNACIÓN ANUAL DEL COMBUSTIBLE DEL GRUPO DE GESTIÓN ENERGÉTICA

FIGURA 3: DIAGRAMA DEL PROCESO ENVIAR PLAN OPERATIVO DEL GRUPO DE GESTIÓN ENERGÉTICA.

FIGURA 4: DIAGRAMA DEL PROCESO REGISTRAR EFICIENCIA DE CONSUMO DEL COMBUSTIBLE DEL GRUPO DE GESTIÓN ENERGÉTICA.

FIGURA 5: DIAGRAMA DE PROCESO GENERAL DE LA DIRECCIÓN DE TRANSPORTE

FIGURA 6: DIAGRAMA DEL PROCESO PLANIFICAR ASIGNACIÓN MENSUAL DEL COMBUSTIBLE DE LA DIRECCIÓN DE TRANSPORTE.

FIGURA 7: DIAGRAMA DEL PROCESO REALIZAR RESUMEN DE CONSUMO DEL COMBUSTIBLE POR TIPO DE ACTIVIDAD DE LA DIRECCIÓN DE TRANSPORTE

FIGURA 8: DIAGRAMA DE PROCESO GENERAL DE LA VICERRECTORÍA ECONÓMICA

FIGURA 9: DIAGRAMA DEL PROCESO APROBAR DISTRIBUCIÓN MENSUAL DEL COMBUSTIBLE DE LA VICERRECTORÍA ECONÓMICA.

FIGURA 10: DIAGRAMA DEL PROCESO APROBAR ENTREGA DEL COMBUSTIBLE ADICIONAL DE LA VICERRECTORÍA ECONÓMICA.

FIGURA 11: MAPA DEL PROCESO DEL GRUPO DE GESTIÓN ENERGÉTICA.

FIGURA 12: MAPA DEL PROCESO DE LA DIRECCIÓN DE TRANSPORTE

FIGURA 13: MAPA DEL PROCESO DE LA VICERRECTORÍA ECONÓMICA

FIGURA 14: DIAGRAMA CASOS DE USO MÚLTIPLES USUARIOS CON LAS MISMAS RESPONSABILIDADES

FIGURA 15: DIAGRAMA DEL CASO DE USO DEL SISTEMA DEL MÓDULO DE ENERGÍA

FIGURA 16: DIAGRAMA DEL CASO DE USO DEL SISTEMA DEL MÓDULO DE TRANSPORTE

FIGURA 17: DIAGRAMA DEL CASO DE USO DEL SISTEMA DEL MÓDULO DE LA VICERRECTORÍA ECONÓMICA

FIGURA 18: DIAGRAMA DEL CASO DE USO DEL SISTEMA DEL MÓDULO DE CONFIGURACIÓN

FIGURA 19: DIAGRAMA DE CLASES DEL DISEÑO DEL CASO DE USO “BUSCAR PLAN OPERATIVO MENSUAL”.

FIGURA 20: DIAGRAMA DE SECUENCIA DEL ESCENARIO “VER PLAN OPERATIVO MENSUAL” DEL CASO DE USO “BUSCAR PLAN OPERATIVO MENSUAL”.

FIGURA 21: DIAGRAMA DE SECUENCIA DEL ESCENARIO “MODIFICAR PLAN OPERATIVO MENSUAL” DEL CASO DE USO “BUSCAR PLAN OPERATIVO MENSUAL”

FIGURA 22: DIAGRAMA DEL MODELO ENTIDAD RELACIÓN DE LA BASE DE DATOS.

FIGURA 23: MODELO DE DESPLIEGUE DEL SISTEMA.

FIGURA 24: MODELO VISTA CONTROLADOR.

FIGURA 25: DIAGRAMA DE COMPONENTES DEL SISTEMA

FIGURA 26: PANTALLA DE LA VISTA AUTENTICAR USUARIO.

FIGURA 27: PANTALLA DE LA VISTA DEL MÓDULO DE ENERGÍA.

FIGURA 28: PANTALLA DE LA VISTA DE GESTIONAR USUARIOS.

FIGURA 29: GRÁFICO DE LAS ITERACIONES DE LAS PRUEBAS REALIZADAS.

FIGURA 30: CÓDIGO DE LA FUNCIONALIDAD DELETEACTION.

FIGURA 31: GRAFO DEL FLUJO ASOCIADO A LA FUNCIONALIDAD DELETEACTION.

FIGURA 32: RESULTADOS DE LAS PRUEBAS DE CARGA Y ESTRÉS PARA 20 USUARIOS

FIGURA 33: RESULTADO DE LA PRUEBAS DE SEGURIDAD GENERADO POR LA HERRAMIENTA ACUNETIX.

ÍNDICE DE TABLAS

TABLA 1: REQUISITOS FUNCIONALES DEL SISTEMA.

TABLA 2: DESCRIPCIÓN DE LOS ACTORES DEL SISTEMA.

TABLA 3: DESCRIPCIÓN DE LAS CLASES DEL DISEÑO.

TABLA 4: DESCRIPCIÓN DE LAS ENTIDADES DE LA BASE DE DATOS.

TABLA 5: ESTÁNDAR Y ESTILOS DE CODIFICACIÓN DEL SISTEMA.

TABLA 6: CASO DE PRUEBA: BUSCAR PLAN OPERATIVO MENSUAL.

TABLA 7: CAMINOS LÓGICOS.

TABLA 8: CASO DE PRUEBA CAMINO #1.

TABLA 9: CASO DE PRUEBA CAMINO #2.

TABLA 10: DESCRIPCIÓN DEL CASO DE USO: BUSCAR PLAN OPERATIVO MENSUAL.

INTRODUCCIÓN

Introducción

Al paso del tiempo, el hombre ha ido dependiendo cada vez más de los recursos energéticos para la satisfacción de necesidades como: iluminación, calefacción, refrigeración y transporte. Esta dependencia energética ha acarreado un abuso de consumo de combustibles fósiles y recursos no renovables. La sociedad de consumo se extiende cada vez más utilizando recursos energéticos para disfrutar de un mayor confort, y a esta calidad de vida son llevados los países en vías de desarrollo donde la mayor parte de la estructura de oferta de energía primaria está basada en petróleo y gas en casi el 90 % a nivel mundial [1].

A nivel internacional la gestión energética se ha convertido en una pieza clave para que una organización, independientemente de su naturaleza o tamaño pueda alcanzar niveles deseables de eficiencia y ahorro de energía; de compromiso con el medio ambiente y con la sociedad; así como de mejora de la competitividad en sus procesos productivos [2]. La economía cubana enfrenta los déficits de suministro energético, lo cual se extiende en mayor o menor grado a todos los sectores de la actividad económica. Esta situación obliga a la dirección del país a tomar medidas y programas para enfrentar esta crisis, cuyo alcance es sectorial y nacional. En virtud de las prioridades asignadas a las empresas exportadoras y servicios sociales básicos, se enfoca dicho suministro, ocasionando gran impacto sobre el resto de las empresas [1].

Dentro de las prioridades de asignación de suministro energético se encuentran las instituciones docentes y de investigación como la UCI. Tanto fuera como dentro de la Universidad, se desarrollan actividades que demandan el traslado diario de un grupo de personas, ocasionando que se requiera distribuir y consumir combustible. En el proceso de distribución y consumo del combustible de la Universidad, intervienen las áreas de: la Dirección de Transporte, el Grupo de Gestión Energética y la Vicerrectoría Económica. Las cuales interactúan entre sí, mediante el envío y respuesta de planificaciones y solicitudes. Por su parte, cada una de ellas desarrolla una serie de actividades para garantizar el correcto flujo y control de la información y los recursos. Estas actividades son realizadas en las áreas de forma manual, ocasionando: que se registre la información en formato duro, retardo en la elaboración y entrega de la documentación, las búsquedas de datos importantes sea lenta y puede existir información duplicada, la información se ve expuesta a un mayor riesgo de deterioro y se presentan dificultades en su almacenamiento, y en consecuencia de lo anteriormente planteado atenta contra el control del recurso energético en la UCI. El

INTRODUCCIÓN

proceso distribución y consumo del combustible se realiza mensualmente, provocando que no posea la inmediatez y retroalimentación necesaria, afectando el servicio de transportación de la Universidad.

Todo lo anteriormente planteado induce a determinar el siguiente **problema de la investigación**: ¿Cómo garantizar el control de los flujos de información relacionados con la distribución y consumo del combustible, para lograr inmediatez y retroalimentación en el proceso de gestión de la información del combustible en las áreas: Dirección de Transporte, Grupo de Gestión Energética y Vicerrectoría Económica en la UCI?.

El **objeto de estudio de la investigación** se enmarca en los procesos de gestión de la información de los recursos energéticos y el **campo de acción** en el proceso de gestión de la información del combustible en la UCI.

Atendiendo a la situación problemática descrita y al problema de la investigación anteriormente planteado, el trabajo investigativo se desarrolla en función del siguiente **objetivo general**: Desarrollar un sistema de gestión de la información del combustible que controle los flujos de información relacionados con la distribución y consumo del combustible, para lograr inmediatez y retroalimentación en el proceso de gestión de la información del combustible en las áreas: Dirección de Transporte, Grupo de Gestión Energética y Vicerrectoría Económica en la UCI.

Para darle solución al objetivo general se definen los siguientes **objetivos específicos**:

1. Investigar los Sistemas de gestión de distribución y consumo del combustible existentes a nivel nacional e internacional.
2. Fundamentar la selección de la metodología de desarrollo de software, herramientas y tecnologías a utilizar en el desarrollo del Sistema de gestión de la información del combustible.
3. Realizar el diseño del Sistema para la gestión de la información del combustible en la UCI.
4. Implementar el Sistema para la gestión de la información del combustible en la UCI.
5. Realizar pruebas al Sistema para la gestión de la información del combustible en la UCI.

Se defiende la idea de que con el desarrollo de un Sistema para la gestión de la información asociado a la distribución y consumo del combustible en la UCI, permitirá alcanzar la inmediatez y retroalimentación necesaria en la dinámica del proceso de distribución y consumo del combustible en la UCI.

Para darle cumplimiento a los objetivos específicos se plantean las siguientes **tareas de la investigación**:

1. Revisión de documentos y entrevista a directivos de las áreas: Dirección de Transporte, Grupo de Gestión Energética y Vicerrectoría Económica en la UCI.

INTRODUCCIÓN

2. Investigación de los procesos para la gestión del combustible en el ámbito nacional e internacional.
3. Identificación de los procesos de distribución y consumo del combustible en la UCI.
4. Caracterización de las metodologías de desarrollo de software, herramientas y tecnologías a utilizar en el desarrollo del sistema.
5. Realización del modelo de negocio para llevar a cabo la solución.
6. Identificación de los Requisitos funcionales y no funcionales que le dan solución al problema de la investigación.
7. Descripción de los Requisitos funcionales y no funcionales para lograr una mayor comprensión de la solución.
8. Realización del Modelo de diseño según la metodología de desarrollo de software seleccionada.
9. Elaboración del modelo de datos.
10. Realización del diagrama de despliegue.
11. Realización del diagrama de componentes.
12. Implementación de los componentes necesarios para desarrollar el Sistema de gestión de la información del combustible.
13. Elaboración y realización de las pruebas para validar el comportamiento del sistema.

Con el propósito de desarrollar las tareas planteadas para la investigación se utilizaron los siguientes **métodos científicos**:

Métodos Teóricos:

- **Analítico-Sintético:** con la utilización del método se han buscado, investigado y analizado documentos para extraer los elementos del proceso de distribución y consumo del combustible a nivel nacional e internacional, que pudieran estar relacionados con el objeto de estudio; realizando consultas a diversas fuentes bibliográficas.
- **Análisis Histórico-Lógico:** se utiliza en el estudio de los antecedentes, la evolución y el desarrollo que han tenido los sistemas de distribución y consumo del combustible.
- **Modelación:** se utiliza para representar los procesos definidos por el sistema mediante la construcción de modelos y diagramas a lo largo del desarrollo de la investigación, simplificando la realidad y facilitando la comprensión de los mismos.

INTRODUCCIÓN

Métodos Empíricos:

- **Entrevista:** se realiza una entrevista a los especialistas de las áreas: Dirección de Transporte, Grupo de Gestión Energética y Vicerrectoría Económica, con el objetivo de lograr una mayor comprensión de las particularidades y características del proceso de distribución del combustible. De la entrevista, se obtuvo la lista de requisitos funcionales del sistema, así como una serie de recomendaciones necesarias para el desarrollo del software.
- **Observación:** se emplea este método para conocer la esencia de la problemática definida, que sirvió de base para el planteamiento del problema de la investigación, además permitió conocer el proceso delimitado a través del objeto de estudio.

Los **aportes prácticos esperados** con la solución que se implementa son los siguientes:

- Brindar funcionalidades que permitan tener la administración del combustible en la UCI.
- Proporcionar la información necesaria con rapidez y calidad entre las áreas involucradas en el proceso.

La estructura del trabajo de diploma está compuesta por cuatro capítulos, un resumen, una introducción, conclusiones, recomendaciones, referencias bibliográficas, los anexos y un glosario de términos. A continuación, se describen los principales aspectos abordados en cada uno de los capítulos:

• **Capítulo I: Fundamentación teórica de los Sistemas de distribución y consumo del combustible.**

En este capítulo se puntualizan los principales conceptos relacionados con el tema, se realizará el estudio del estado del arte a nivel nacional e internacional sobre los sistemas de distribución y consumo del combustible, además de definirse las tecnologías, metodologías de desarrollo de software y herramientas a utilizar durante el desarrollo de la aplicación.

• **Capítulo II: Características del Sistema de gestión de la información del combustible.**

En este capítulo se realiza una descripción general de la solución propuesta y su funcionamiento. Además se hace un análisis del modelo del negocio correspondiente al sistema. Se capturan los requisitos funcionales y no funcionales y se formalizan los artefactos derivados de la metodología de desarrollo de software que se seleccione.

INTRODUCCIÓN

- **Capítulo III: Diseño del Sistema de gestión de la información del combustible.**

En este capítulo se realiza la modelación detallada y la construcción de la estructura de la aplicación. Se define la estructura y los elementos del diseño, se muestran los diagramas correspondientes al diseño del sistema, así como el modelo de datos.

- **Capítulo IV: Implementación y pruebas del Sistema de gestión de la información del combustible.**

En este capítulo se abordan aspectos relacionados con la implementación del sistema en base a la arquitectura de desarrollo de software. Se documentan las pruebas de software al Sistema de gestión de la información del combustible, para verificar que responda a un correcto funcionamiento, garantizar la confidencialidad y detectar fallas en el sistema.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

Capítulo I: Fundamentación teórica de los Sistemas de distribución y consumo del combustible.

1.1.- Introducción

En el capítulo se presentan los conceptos fundamentales que constituyen la base teórica de los procesos de gestión de la distribución y consumo del combustible. Se realizará un estudio de los sistemas homólogos sobre las diferentes alternativas aplicadas durante el proceso de distribución y consumo del combustible existentes a nivel nacional e internacional. Se analizarán las metodologías de desarrollo de software, tecnologías, y herramientas actuales que guiarán el proceso de desarrollo de software y se seleccionarán las más adecuadas a utilizar durante el desarrollo de la solución que se propone.

1.2.- Principales conceptos asociados a la gestión de distribución y consumo del combustible.

1.2.1.- Recurso energético

Se denomina Recurso energético a los medios o recursos que ofrece la naturaleza, y a partir de los cuales, mediante un proceso industrial, se obtiene alguna forma de energía que puede ser directamente utilizada por el consumidor o por alguna actividad productiva. Los recursos energéticos pueden ser:

- Sólidos, como el carbón o la biomasa.
- Líquidos, como el petróleo o el gas natural [3].

1.2.2- Combustible

El combustible es cualquier material que es capaz de liberar energía una vez que se oxida de manera violenta y con desprendimiento de calor. Los combustibles fluidos son mayormente empleados a instancias de motores de combustión interna, destacándose el gasóleo, el querosene, la gasolina o nafta. Otro combustible es el llamado combustible fósil, que es aquel que se ha formado hace millones de años en el planeta a partir de los restos orgánicos de plantas y animales muertos y tras miles de años se produjeron las reacciones químicas que transformaron tales restos en carbón, gas y petróleo [4].

1.2.3.- Gestión de distribución y consumo del combustible

Se entiende por gestión de distribución y consumo del combustible al diseño y la puesta en práctica de un sistema de control, supervisión y seguimiento de la distribución y consumo del combustible global e individual de los vehículos de una flota de transporte. La gestión del combustible permite aprovechar de la manera más rentable cada litro de combustible adquirido, contribuyendo con ello no sólo a la economía de

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

la empresa, sino también al ahorro energético y a la conservación del medio ambiente. Una adecuada gestión del combustible está ligada 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 [5].

1.2.5.- Sistema de gestión de flotas de equipos de transporte

Un Sistema de gestión de flotas de equipos de transporte, le permite a una institución controlar los vehículos pertenecientes a la misma, tales como: coches, camiones y motores. Este sistema de gestión puede incluir una serie de funciones tales como: mantenimiento de vehículos, financiación, seguimiento de los mismos, gestión del combustible, gestión de conductores, control de las fechas de vencimiento de las revisiones técnicas de los vehículos, vencimiento de las licencias de conducir de los conductores [6] [7].

1.3.- Sistemas de gestión de distribución y consumo del combustible a nivel internacional

En el mundo existen numerosos Sistemas de gestión del combustible que han sido desarrollados con el propósito de satisfacer necesidades existentes en diferentes organizaciones, los cuales permiten controlar de forma efectiva la distribución y consumo del combustible y otros indicadores de los vehículos asociados a dichas entidades. Algunos de estos son: Gestión de Flota, KreaFlota y FlotaWeb.

Gestión de Flota

El Sistema de gestión de flotas de Informática EUGCOM¹ 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 [8]. Sus principales módulos son los siguientes:

- **Combustibles:** Módulo de ingreso y mantención de los datos que componen el movimiento de cargas de combustible efectuados a los vehículos [8].
- **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

¹ Empresa informática dedicada al desarrollo de software.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

sumamente personalizado según los requerimientos del usuario y permite calcular los datos más importantes por cada vehículo [8].

- **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 de errores de las cargas de combustible registradas en el sistema, y es capaz de reparar automáticamente los errores más usuales [8].

KreaFlota

Es 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. 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. Además, se lleva un control exhaustivo del petróleo, llevando un registro diario de consumo para los vehículos, permitiendo obtener rendimientos de los mismos [9].

FlotaWeb

Es una plataforma de Administración de Flotas, que permite gestionar todos los procesos relacionados con la flota vehicular. Puede controlar todos los costos, evaluar rendimientos y costos por kilómetros u horas, administrar eficientemente las llantas y el combustible [10]. Sus principales módulos son los siguientes:

- **Vehículos:** Centraliza la información completa de los vehículos y todos los procesos que tienen que ver con su correcta administración [10].
- **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 [10].

Los sistemas internacionales anteriormente mencionados se analizaron con el objetivo de conocer cómo gestionan toda la información relacionada con el control del combustible. Los sistemas de 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, 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 Sistema de gestión de la información del combustible de la UCI, debido a que no planifican la asignación del combustible y por ende no se pueden calcular los

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

índices de consumos de acuerdo al plan y al real, esta información es relevante en la Dirección de Transporte de la UCI para realizar los cierres de consumo. Gestión de Flota y KreaFlota son aplicaciones de escritorio, por lo que su acceso se limita al ordenador donde están instaladas. 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 a ella todas las áreas involucradas en el proceso de distribución y consumo del combustible.

1.4.- Sistemas de gestión de distribución y consumo del combustible a nivel nacional

En Cuba a lo largo de los años se han implementado diversos sistemas informáticos para el control estadístico y financiero, particularmente en la administración de finanzas en diversos sectores como el transporte. Algunos de los sistemas son: Apolo, Sistema de Gestión de Mantenimiento Vehicular y SISCOMPA.

Apolo

Es un producto nacional perteneciente al Ministerio de la Informática y las Comunicaciones (MIC) y la Empresa Cubana Nacional de Software (DESOFT), 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 de equipos de transporte 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. 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 equipos de transporte, de la asignación de combustibles, lubricantes y líquidos de freno, la cantidad de Km programados para cada marca o línea e información del personal de la entidad [11].

Sistema de Gestión de Mantenimiento Vehicular

Es un producto realizado por el Centro 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

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

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 [12]. Presenta los siguientes módulos:

- **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 regir todas las unidades policiales para la realización de los mantenimientos preventivos planificados que le corresponden [12].
- **Vehículo:** Es el módulo desde el cual se va a gestionar toda la información de las unidades policiales (adicionar, modificar, consultar, asignar a dependencias, registrar accidentes, realizar inspecciones técnicas y recepción de unidades) [12].

SISCOMPA

El sistema fue creado por la Empresa de desarrollo de software para el transporte (TRANSOFT), garantiza el control y la gestión de la flota automotor de transporte de carga, contribuyendo al ahorro de recursos materiales, combustible y tiempo. Está compuesto por módulos técnicos que intercambian información entre ellos, los cuales son: tráfico, técnica, seguridad automotor y portadores energéticos. También posee un módulo de dirección para los directivos y un módulo de administración. Esta herramienta permite potenciar la actividad de ahorros en combustibles, lubricantes, gomas, baterías, etc. Ofrece a los directivos la información necesaria para la toma de decisiones, siendo una poderosa herramienta de dirección [13].

Los sistemas nacionales anteriormente mencionados se analizaron con el objetivo de conocer cómo gestionan toda la información relacionada con el control del combustible. Los sistemas Apolo, Sistema de mantenimiento vehicular y SISCOMPA cuentan con un módulo para la gestión del combustible que permiten realizar los registros del consumo de los vehículos en función de obtener su rendimiento y generar informes. Sin embargo, sus funcionalidades no se pueden reutilizar como parte de la propuesta de solución del Sistema de gestión de la información del combustible de la UCI, debido a que el proceso de distribución y consumo del combustible que realizan los sistemas no están acorde con el proceso que realiza la Universidad y por ende no manejan la información necesaria para gestionar la información del combustible en la UCI.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

Luego de realizar una investigación sobre los Sistemas de gestión del combustible anteriormente mencionados, se arriba a la conclusión de que no existe un sistema que responda correctamente a las necesidades específicas del proceso de distribución y consumo del combustible de la UCI, por lo que se hace necesario crear un sistema de gestión que responda a las necesidades del proceso de distribución y consumo del combustible. Igualmente que sea consecuente con el proceso de conseguir la soberanía tecnológica, haciéndose necesario el desarrollo del sistema sobre herramientas y tecnologías libres basadas en la web.

1.5.- Actualidad y necesidad del trabajo

En la actualidad no existe una herramienta informática que permita gestionar la información del proceso de distribución y consumo del combustible en la UCI, por lo que se hace necesario el desarrollo de un sistema web que realice el proceso. El sistema de gestión del proceso de distribución y consumo del combustible permitirá controlar la distribución y consumo del combustible en la Universidad y disponer de la operatividad de la flota de equipos de transporte en las actividades diarias de la UCI.

1.6.- Herramientas, tecnologías y metodologías de desarrollo de software

1.6.1.- Metodología de desarrollo de software

Las metodologías para el desarrollo del software imponen un proceso disciplinado sobre el desarrollo de software con el fin de hacerlo más predecible y eficiente. Este tipo de metodología tiene como principal objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. Las metodologías de desarrollo de software se encuentran divididas en dos grandes grupos: las ágiles y las robustas o tradicionales. Las primeras se enfocan en dar mayor valor al individuo, a la colaboración con el cliente y al desarrollo incremental del software con iteraciones muy cortas, mostrando su efectividad en proyectos con requisitos muy cambiantes y cuando se exige reducir drásticamente los tiempos de desarrollo. En cambio, las robustas son conocidas por la forma tradicional de desarrollar software y se basan en un proceso unificado, con un gran flujo de trabajo, obteniéndose una amplia documentación y una detallada planificación inicial que debe seguirse estrictamente; además, son adaptables a proyectos a largo plazo [14].

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

Metodología ágil

Inicialmente las metodologías se caracterizaron por tener elementos en común y ser excesivamente pesadas o rígidas por su carácter normativo. Es por ello que surge una corriente de desarrollo llamada Metodologías ágiles. Esta definición moderna tiene como principales objetivos:

- Flexibilidad ante los cambios que quisieran efectuar los clientes a lo largo del desarrollo de un sistema.
- Depositar en manos del cliente los elementos desarrollados en cada etapa del trabajo para que este realice las verificaciones pertinentes.
- Evitar exceso de documentación y malentendidos trabajando en conjunto entre el cliente y el equipo de desarrollo con una comunicación directa [15].

Entre las metodologías ágiles comúnmente usadas se encuentran: XP (por sus siglas en inglés *Extreme Programming, Programación Extrema*) y OpenUp (por sus siglas en inglés *Open Unified Process, Proceso Unificado Abierto*).

XP

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. XP se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Es recomendable emplearlo solo en proyectos a corto plazo y en caso de fallar existen altas comisiones por el desarrollo de software [16].

OpenUp

Es un modelo de desarrollo de software, desarrollado por la fundación Eclipse. Esta preserva las mejores prácticas de RUP (según sus siglas en inglés *Rational Unified Process, Proceso Unificado de Rational*), por lo que entre sus principales características se mantiene un desarrollo iterativo e incremental, dirigido por casos de uso y centrado en la arquitectura para reducir al mínimo los riesgos y organizar el desarrollo. Esta metodología está diseñada para equipos pequeños ya que se espera obtener resultados en un corto período de tiempo y utilizar los procesos, productos, roles, y tareas que sean indispensables para el mismo. Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas. El ciclo de vida de OpenUp consiste de cuatro fases: Concepción, Elaboración, Construcción y Transición [17].

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

Entre las metodologías de desarrollo de software descritas con anterioridad se decidió utilizar OpenUp, por ser adaptable a las necesidades del producto, lo que facilita que no se añada un trabajo excesivo por la cantidad de roles o documentación que se genera con el uso de la metodologías pesadas. También es una metodología de desarrollo de software diseñada para pequeños equipos organizados. OpenUp permite detectar errores tempranos a través de un ciclo iterativo y como metodología ágil es idónea para realizar el sistema de gestión en cuestión, ya que se destaca por ser un proceso de desarrollo de software simplificado, basada en las mejores prácticas de RUP, las cuales ya han probado su efectividad. Se hace necesario aplicar la metodología de desarrollo de software OpenUp debido a que a pesar de ser una metodología ágil tiene principios de las metodologías pesadas como es que brinda una documentación detallada de la información, y muy a pesar de que se requiere la obtención del producto de forma inmediata, se hace necesario obtener una documentación detallada del proceso de gestión de la información de la distribución y consumo del combustible en la UCI, debido a que es un proceso que fluye información sensible a través de diversos Departamentos anteriormente mencionados, y se requiere dejar plasmado la dinámica del proceso en vistas a seguir implementando subprocesos que se derivan del proceso inicial.

1.6.2- Herramientas y tecnologías

En la actualidad existe una gran revolución de los medios informáticos, donde los avances en las nuevas tecnologías tienen el propósito de ofrecer soluciones adaptables a las exigencias del usuario final. La integración de las tecnologías de desarrollo con las herramientas, ha surgido como una alternativa que permite elaborar aplicaciones web para la gestión de la información. Además permiten resolver problemas actuales de una organización, facilitando así, la interacción de los usuarios con los sectores involucrados en la problemática.

Algunas de las tecnologías y herramientas empleadas para el desarrollo actual de aplicaciones web son: JSP (por sus siglas en inglés *Java Server Pages*, *Páginas de Servidor Java*) y PHP (por sus siglas en inglés *Hypertext Pre-Processor*, *preprocesador de hipertexto*) como tecnologías de programación del lado del servidor, como lenguaje de modelado UML (por sus siglas en inglés, *Unified Modeling Language*, *Lenguaje Unificado de Modelado*), EXTJS y Bootstrap como framework de interfaz, Symfony y Zend Framework como framework de desarrollo; como servidor web Apache y IIS (por sus siglas en inglés *Internet Information Server*, *Servidor de Información de la Internet*); como sistema gestor de base de datos

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

PostgreSQL y MySQL; como herramienta CASE (por sus siglas en inglés *Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora*) Visual Paradigm y Rational Rose y como IDE (por sus sigla en inglés *Integrated Development Environment, Entorno de Desarrollo Integrado*) NetBeans. Para poder realizar una aplicación web es necesario tener conocimiento de las tecnologías y herramientas que aportan mejoras, garantizando así un desarrollo y mantenimiento estándar, de este modo se establecen criterios de selección para las mismas.

Tecnologías de programación del lado del servidor

JSP

En su versión 2.1, es una tecnología web del lado del servidor, que se usa habitualmente para crear documentos XHTML (por sus siglas en inglés *eXtensible HyperText Markup Language, Lenguaje de Marcado de Hipertexto Extensible*) y XML (por sus siglas en inglés *eXtensible Markup Language, Lenguaje de marcas extensible*) dinámicos. Con JSP se pueden crear aplicaciones web que se ejecuten en varios servidores web, además es compatible con variadas plataformas, o sea, se puede ejecutar sobre varios sistemas operativos como: Linux y Windows. Esta tecnología presenta similitudes con otras como: PHP y ASP (por sus siglas en inglés *Active Server Pages, Página de Servidor Activo*), pues admite que su código sea embebido dentro del HTML (por sus siglas en inglés *HyperText Markup Language, Lenguaje de marcas de hipertexto*) con el propósito de crear información dinámicamente, basándose en instrucciones o acceso a bases de datos. Esta tecnología hace uso del lenguaje de programación Java, este es un lenguaje multiplataforma, el cual posee capacidad multihilo y tiene integrado el protocolo TCP/IP (por sus siglas en inglés *Transmission Control Protocol/Internet Protocol, Protocolo de Control de Transmisión/Protocolo de Internet*), lo que lo hace un lenguaje ideal para Internet [18].

Ventajas que proporciona:

- Facilita la conexión a bases de datos.
- Permite crear aplicaciones web que se ejecuten en varios servidores web.
- Permite integrar los registros de órdenes con clases del lenguaje de programación Java.
- Es libre y multiplataforma.
- Tiene el código bien estructurado.
- Se puede integrar con los módulos de Java.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

Desventajas:

- Requiere de una memoria más amplia debido a que está basado en Java y consume gran cantidad de recursos.

PHP

En su versión 5.4.9 es una tecnología del lado del servidor, de código abierto y diseñado en sus inicios para el desarrollo de páginas web dinámicas. Una de sus principales características es la capacidad de soportar gran cantidad de bases de datos. También permite la integración con varias bibliotecas externas permitiéndole al programador analizar código XML y generar documentos en diferentes formatos. Este lenguaje es rápido, libre, orientado a objetos y multiplataforma, pues permite ser utilizado sobre diferentes sistemas operativos como: Linux y Windows. Posee además una amplia librería de funciones y cuenta con una extensa documentación, la cual permite un rápido aprendizaje [19].

Ventajas que proporciona:

- Permite la integración con disímiles tipos de servidores de bases de datos tales como: MySQL, PostgreSQL y Oracle.
- Permite emplear técnicas de programación orientada a objetos.
- Es libre y multiplataforma.

Desventajas:

- Todo el trabajo lo realiza el servidor y no delega al cliente. Por tanto puede ser más ineficiente a medida que las solicitudes aumenten de número.

Se determinó como tecnología de programación PHP versión 5.4.9, el cual está orientado al desarrollo web, es de gran velocidad por lo que no requiere de muchos recursos de hardware y además se integra perfectamente con muchos servidores. Es libre y está disponible bajo la licencia GPL (por sus siglas en inglés *General Public License, Licencia Pública General*), es multiplataforma por lo que no tendrá ningún inconveniente al usarlo en cualquier computadora de la Universidad. Se caracteriza por la simplicidad de su código y por la amplia documentación que existe. Además se requiere utilizar este lenguaje de programación debido a que es necesario desarrollar una aplicación web para acceder al sistema a través de un servidor web por diversos Departamentos de la Universidad y resulta muy engorroso tener que instalar en todas las estaciones de trabajo la aplicación para que sea accedida por los usuarios.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

Lenguaje de modelado

UML en su versión 2.0, es el lenguaje gráfico más conocido y utilizado en la actualidad para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. Este lenguaje es semejante al de la vida real, claro y uniforme para el diseño orientado a objetos, ya que permite la fuerte integración entre herramientas, procesos y dominios. UML se especializa en el modelado de elementos conceptuales como son: procesos de negocio y funciones de los sistemas. También este lenguaje permite escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables [20].

Framework de Interfaz

ExtJS

En su versión 4.0, es un *framework* que logra trabajar con otras librerías JavaScript empleando adaptadores. Este *framework* ha demostrado ser muy eficaz y competente en la realización de interfaces complejas ya que tiene embebido la mayoría de los controles de los formularios web incluyendo celdas para mostrar datos y elementos semejantes a la programación desktop como: formularios, paneles, barras de herramientas, árboles y menús [21].

Ventajas que proporciona:

- Relación entre cliente-servidor balanceada: Se distribuye la carga de procesamiento permitiendo que el servidor pueda atender más clientes al mismo tiempo.
- Permite realizar complejos módulos en una página web.

Bootstrap

En su versión 3.0.1 es un entorno de trabajo definido con artefactos o módulos de software concreto, desarrollado por Twitter. Esta tecnología simplifica el proceso de creación de la interfaz de usuario combinando HTML, JavaScript y CSS (por sus siglas en inglés *Check Cascading Style Sheets, Hojas de Estilo en Cascada*). Además se adapta a los distintos navegadores con numerosos componentes webs como: botones, etiquetas, alertas, entre muchos otros definidos en la web. Bootstrap fue programado para dar soporte a CSS 3 y HTML 5. Bootstrap está diseñado para todos los niveles: diseñador, desarrollador y principiante. Este framework se utiliza para hacer más fácil y rápida su implementación. Bootstrap está definido por módulos que son reutilizables e independientes en la página web y es una gran comunidad abierta y seguida por millones de personas [22].

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

Ventajas que proporciona:

- Permite crear interfaces que se adapten a los diferentes navegadores, tanto de escritorio como tablets y móviles a distintas escalas y resoluciones.
- Se integra perfectamente con las principales librerías Javascript, por ejemplo JQuery.
- Ofrece un diseño sólido usando LESS y estándares como CSS3/HTML5.
- Es un framework ligero que se integra de forma limpia en nuestro proyecto actual.
- Funciona con todos los navegadores, incluido Internet Explorer usando HTML Shim para que reconozca los *tags* HTML5.
- Dispone de distintos *layout* predefinidos con estructuras fijas a 940 píxeles de distintas columnas o diseños fluidos.

Como *framework* de interfaz se determinó hacer uso de Bootstrap versión 3.0.1 por ser muy eficaz y competente en la realización de interfaces web, además de ser recomendable para sitios web, se adapta a los distintos navegadores de la Universidad. Los diseños creados con Bootstrap son simples, limpios e intuitivos, permitiéndole ganar al sistema en agilidad a la hora de cargar y al adaptarse a otros dispositivos. Se requiere utilizar el framework debido a que brinda componentes para realizar el diseño de las vistas del sistema de forma organizada y según los requerimientos del sistema.

Framework de desarrollo

Symfony

En su versión 2.1.8 es un framework diseñado para optimizar, el desarrollo de las aplicaciones web basado en el patrón de arquitectura de software Modelo Vista Controlador (MVC). Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Symfony puede ser completamente personalizado para cumplir con los requisitos de las empresas que disponen de sus propias políticas y reglas para la gestión de proyectos y la programación de aplicaciones.

Ventajas que proporciona:

- Tiene su propia forma de trabajo, con variantes del MVC clásico como la capa de abstracción de base de datos, el controlador frontal y las acciones.
- Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- El completo sistema de log permite a los administradores acceder hasta el último detalle de las actividades que realiza la aplicación
- Fácil de instalar y configurar en la mayoría de plataformas de programación.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

- Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.

Desventajas

- Gran parte de la velocidad de Symfony se debe a un uso extensivo del caché por lo que cuando estás desarrollando tiende a ser algo tedioso tener que estar limpiando el caché de vez en cuando.
- Se necesita un VPS (por sus siglas en inglés *Virtual Private Server*, *Servidor Virtual Privado*) para poder publicar aplicaciones en la web porque es necesario poder descargar e instalar cosas en un servidor para que symfony funcione apropiadamente [24].

Zend Framework

En su versión 2.0.6, es un framework para el desarrollo de aplicaciones y servicios web con PHP. Brinda soluciones para construir sitios web modernos, robustos y seguros. Además, implementa el patrón MVC, es orientado a objetos y sus componentes tienen un bajo acoplamiento por lo que se puede usar en forma independiente.

Ventajas que propone:

- Zend Framework permite reutilizar el código desarrollado.
- El mantenimiento de las aplicaciones es muy sencillo.

Desventajas:

- Zend Framework necesita de un servidor dedicado para poder sacar el máximo provecho a las funciones con las que este cuenta, porque como se menciona anteriormente, se configura mediante la línea de comandos, entonces cualquier cambio que se necesitara hacer se realizaría a través de la línea de comandos del servidor dedicado, en el caso de un servidor compartido por ejemplo, se tendría que bajar, configurar en localhost, modificar, subir, configurar en servidor.
- Requiere de una larga curva de aprendizaje.
- La comunidad de este lenguaje es muy pobre, por lo que podría ser difícil encontrar algún buen manual para realizar un proceso o aclarar una duda [25].

Entre los frameworks para PHP se escogió Symfony versión 2.1.8 debido a que posee una amplia documentación tanto en español como en inglés, tiene una gran variedad de plugins, que permiten agilizar el trabajo, reduce el tiempo de desarrollo pues proporciona varias herramientas y clases que pueden ser reutilizadas por los desarrolladores. A diferencia de Zend Framework que necesita de una gran configuración antes de poder comenzar a trabajar, la estructura de archivos no está definida por sí misma,

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

cualquier desarrollador puede modificarla, esto indica que no sería compatible con otra aplicación desarrollada con Zend Framework. Se requiere utilizar el framework debido a que ofrece características permitiendo obtener aplicaciones web robustas y seguras.

Apache

En su versión 2.2.20, es un servidor web de código abierto, altamente configurable y modular. Utiliza Perl, PHP y otros lenguajes *scripts*. Su función principal es analizar cualquier archivo solicitado por un navegador y mostrar resultados correctos de acuerdo con el código del archivo. Permite configurar los informes de errores, presenta visualización de códigos en numerosos niveles y la capacidad de determinar qué nivel del navegador puede aceptar el contenido. Es uno de los primeros servidores en soportar host basados en direcciones IP (por sus siglas en inglés *Internet Protocol*, *Protocolo de Internet*) y host virtuales. Tiene un elaborado índice de directorios, un directorio de alias, informe de errores HTTP HTML (por sus siglas en inglés *Hypertext Transfer Protocol*, *Protocolo de Transferencia de Hipertexto*) configurable, gestión de recursos para procesos hijos, reescritura de las URL (por sus siglas en inglés *Uniform Resource Locator*, *Localizador de Recurso Uniforme*), comprobación de ortografía de las URL y manuales *online* [26].

Ventajas que proporciona:

- Esta incluye formatos de configuración no estándar.
- Es multiplataforma.
- Permite elegir el servidor web Apache.
- Soporta múltiples plataformas por lo que genera mayor usabilidad, dando la opción de utilizar diferentes sistemas operativos sin ningún problema.

Desventajas:

- No dispone de un entorno integrado con una sofisticada interfaz de usuario, asistente y ayuda en línea [27].

IIS

En su versión 7.5, es un servidor web capaz de brindar servicios web, los cuales transforman a una PC en un servidor web tanto para intranet como para Internet, permitiendo de esta forma publicar páginas web. Este servidor proporciona gran seguridad, capacidad de administración y escalabilidad para aplicaciones web sobre todas las versiones de *Windows Server 2003*. Además proporciona las herramientas y funciones necesarias para administrarlo de forma sencilla [28].

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

Ventajas que proporciona:

- Proporciona un ambiente confiable para lograr una mejor seguridad, ya que incluye la verificación del cambio de las aplicaciones.
- Proporciona una alta capacidad de administración, ya que incluyen cambios tecnológicos y de procesamiento de solicitudes.

Desventajas:

- Solo se puede utilizar en Sistemas operativos de Windows.

Se determinó como servidor web Apache versión 2.2.20 porque no requiere la utilización de muchos recursos y permite además que muchos lenguajes puedan ser utilizados del lado del servidor. Posee además la capacidad de permitir la protección por contraseñas de las páginas de un gran número de usuarios, la visualización del código HTML en varios niveles, registra los errores en varios formatos y tiene soporte para *host* virtuales. Se requiere utilizar el servidor web Apache debido a que la información que maneja el proceso de gestión de la información del combustible es accedida por diversos Departamentos de la Universidad y resulta muy engorroso tener que instalar en todas las estaciones de trabajo la aplicación para que sea accedida por los usuarios. Además, se prevé en el futuro que se incremente el número de usuarios conectados concurrentemente, por lo que se requiere la implantación de una aplicación cliente-servidor.

Visual Paradigm

Es una herramienta CASE que utiliza el lenguaje de modelado estándar UML, permite la generación de códigos e ingeniería inversa. Esta herramienta cumple con las políticas de migración a software libre en Cuba, ya que es una herramienta multiplataforma que se puede utilizar tanto en Linux como en Windows. La misma propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código para entornos integrados de desarrollo tales como: NetBeans, Eclipse, Oracle JDeveloper y JBuilder [29].

Ventajas que proporciona:

- Tiene una interfaz muy intuitiva y es de fácil aprendizaje para los desarrolladores.
- Permite la generación automática de diagramas a partir de descripciones de casos de usos.
- Permite hacer descripción de los casos de usos dando una gran variedad de plantillas predeterminadas permitiendo personalizarlas.

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

- Combina las funcionalidades de todas las ediciones en una amplia plataforma de modelado visual.

Rational Rose Enterprise

Proporciona un lenguaje común de modelado y facilita la creación de software de calidad rápidamente. Es una herramienta de desarrollo que utiliza UML para realizar el desarrollo de software de aplicaciones, modelado de datos, servicios de diseño web, modelado de negocios y el modelado basado en componentes. Pero su principal desventaja es que solo puede ser utilizado en Sistemas operativos de Windows, además de ser una herramienta privativa [30].

Como herramienta CASE fue seleccionada Visual Paradigm versión 8.0, pues propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código para entornos integrados de desarrollo tales como: NetBeans, Eclipse, Oracle JDeveloper y JBuilder. Esta herramienta a diferencia de Rational Rose es libre y se destaca por ser multiplataforma. Se requiere utilizar la herramienta CASE debido a que la información que maneja el proceso de gestión de la información del combustible es extensa y se hace necesario modelar todos los procesos que se realizan para realizar la distribución y consumo del combustible en la UCI.

Netbeans

En su versión 7.1, es un IDE bajo licencia GPL y de código abierto. Esta herramienta tiene la finalidad de permitirle a los desarrolladores crear diferentes sistemas y proyectos orientados sobre todo a la creación de soluciones en lenguaje Java, ya sea que se encuentren en Java SE (Edición Estándar) o Java EE (Edición Empresarial), además de soportar otros lenguajes tales como PHP y JavaScript [31]. Se requiere utilizar Netbeans debido a que facilita el desarrollo del sistema, al proveer un entorno de desarrollo profesional para desarrollar sistemas web.

PostgreSQL

En su versión 9.1, es uno de los motores de base de datos relacionales más potentes que existen actualmente. Permite ejecutar consultas SQL, las cuales posibilitan actualizar, insertar, eliminar y realizar reportes sobre los datos almacenados en ficheros o bases de datos. Ofrece la posibilidad de ejecutar y trabajar varios procesos al mismo tiempo sobre la misma tabla sin ser dañada, donde cada usuario obtiene una versión de lo último que ha hecho evitando la pérdida de información. Tiene su propio lenguaje PL/PgSQL, pero también se pueden usar lenguajes como C, C++, Gambas, Java PL, Java Web, Perl, Php, Python [32]. Este gestor ofrece muchas ventajas respecto a otros sistemas de bases de datos como son:

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

- Mejor soporte que los proveedores comerciales: tiene una importante comunidad de profesionales.
- El código fuente está disponible para todos de manera gratuita.
- Multiplataforma: PostgreSQL está disponible en varias plataformas como son Linux y Windows.

MySQL

En su versión, es un Sistema de gestión de base de datos relacional y multiusuario, fue creado por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca. MySQL es propietario y está patrocinado por una empresa privada. Es muy utilizado en aplicaciones web y en plataformas Linux y Windows [33].

Ventajas que proporciona:

- Alta velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.
- Facilidad de configuración e instalación.
- Soporta gran variedad de sistemas operativos.

Fue seleccionado como sistema gestor de bases de datos PostgreSQL versión 9.1, ya que es uno de los motores de base de datos relacionales más potentes que existen actualmente, además de ser multiplataforma, y a diferencia de MySQL, es libre. Como administrador de bases de datos para este gestor se seleccionó pgAdmin en su versión 1.16.1, ya que está diseñado para satisfacer las necesidades de todos los usuarios, teniendo en cuenta la realización de consultas SQL desde la más sencilla, hasta el desarrollo de bases de datos de alta complejidad. Se requiere utilizar el motor de base de datos debido a que la información que maneja el proceso de gestión de la información del combustible es creciente en el tiempo y se hace necesario almacenarla y acceder a ella a través del sistema.

1.7.- Notación de Modelado de Procesos de Negocio (BPMN)

La notación BPMN (por sus siglas en inglés *Business Process Model and Notation*, *Notación de Modelado de Procesos de Negocio*) 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 [34]. Por la importancia que tiene este tipo de modelado se utiliza para la modelación del proceso de distribución y consumo del combustible

CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA DE LOS SISTEMAS DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE

en la Universidad. 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 éstos; 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. Se requiere utilizar la notación BPMN para lograr una mayor comprensión y representación del modelado del proceso de gestión de la información del combustible que abarca Departamentos de la Universidad anteriormente mencionados.

1.8.- Conclusiones

- El análisis de las características y funcionamiento de los Sistemas de gestión de información de la distribución y consumo del combustible en el ámbito nacional e internacional, permitió comprobar que los sistemas estudiados no poseen las características específicas de la flota de equipos de transporte de la Universidad para realizar el proceso de distribución y consumo del combustible en la UCI.
- El estudio sobre las herramientas, tecnologías y metodologías de desarrollo de software utilizadas en la implementación de sistemas de gestión de información, permitió la elección de las herramientas, tecnologías y metodología de desarrollo de software adecuada para desarrollar el sistema. Se seleccionó como lenguaje de programación PHP en su versión 5.4.3, como herramienta de modelado Visual Paradigm en su versión 8.0, como sistema gestor de bases de datos PostgreSQL en su versión 9.1 y pgAdmin en su versión 1.14.0 para la administración, como servidor web Apache en su versión 2.2.20, con el propósito de recrear una interfaz fácil y sencilla de usar se decidió utilizar Bootstrap en su versión 3.0.1 y como metodología de desarrollo de software OpenUp.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

Capítulo II: Características del Sistema de gestión de la información del combustible

2.1.- Introducción

En el capítulo se realiza una breve descripción de los principales procesos identificados en cada área. Además ofrece la posibilidad de explicar el flujo de los procesos a través de los diagramas correspondientes modelados con la herramienta CASE Visual Paradigm. Igualmente, se describen los requisitos funcionales y no funcionales del sistema, finalizando con la representación del diagrama de caso de usos del sistema y la descripción de los mismos.

2.2.- Modelado del negocio

El modelo de negocio es esencial para alcanzar un mayor entendimiento de la situación planteada. Describe de forma clara y concisa lo que el negocio ofrece a sus clientes y sirve de referencia para el desarrollo de las ventajas competitivas de la institución. La fase de modelado de negocio es la primera y más importante del ciclo de vida correspondiente al desarrollo de software. Es justamente la fase donde se describen los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización [35]. Además se identifican los objetos que forman parte del negocio. Para modelar los procesos de negocio identificados se utilizó la Notación BPMN².

Procesos del Grupo de Gestión Energética

En el Grupo de Gestión Energética se desarrollan tres procesos del negocio: Planificar asignación anual del combustible, Enviar el plan operativo mensual y Registrar eficiencia de consumo del combustible (Ver Figura 1).

- **Planificar asignación anual del combustible:** El proceso de Planificar la asignación anual del combustible comienza cuando el Especialista de Energía consulta los resúmenes de consumo mensual que le envía la Dirección de Transporte y a partir de la consulta elabora la asignación anual del combustible. Luego envía la planificación de la asignación anual del combustible a la

² BPMN “es un nuevo estándar de modelado de procesos de negocio, en donde se presentan gráficamente las diferentes etapas del proceso del mismo. La notación ha sido diseñada específicamente para coordinar la secuencia de procesos y los mensajes que fluyen entre los diferentes procesos participantes.” [35].

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

Vicerrectoría Económica. En la Vicerrectoría Económica es revisada por el especialista de esa área, si no es aprobada se envía al Grupo de Gestión Energética por correcciones y luego de volver a revisarla, si se encuentra bien elaborada la asignación se envía al Grupo de Gestión Energética, para que el Especialista de Energía envíe la Información al Ministerio de Educación Superior (MES), terminando así este proceso (*Ver Figura 2*).

- **Enviar plan operativo mensual:** El proceso de Enviar plan operativo comienza cuando el Especialista de Energía recibe el plan operativo del mes en curso. Luego de archivarlo lo envía a la Dirección de Transporte y así termina este proceso (*Ver Figura 3*).
- **Registrar eficiencia de consumo del combustible:** El proceso de Registrar eficiencia de consumo del combustible comienza cuando el Especialista de Energía recibe el resumen de consumo mensual del combustible. Luego de consultarlo, elabora la información de la eficiencia de consumo del combustible y así termina este proceso (*Ver Figura 4*).

Procesos de la Dirección de Transporte

En la Dirección de Transporte se desarrollan dos procesos del negocio: Planificar asignación mensual del combustible y Realizar resumen de consumo del combustible por tipo de actividad (*Ver Figura 5*).

- **Planificar asignación mensual del combustible:** El proceso de Planificar la asignación mensual del combustible comienza cuando el Especialista de Transporte consulta el plan operativo mensual que le envía el Grupo de Gestión Energética. Luego chequea el estado operativo de todos los equipos de la Universidad, posteriormente elabora la asignación mensual del combustible para los vehículos de la universidad. La asignación del combustible elaborada se le envía directamente al Vicerrector Económico o se la envía al Vicerrector de Ciudad para que la revise y luego se la envía al Vicerrector económico. En el caso de enviárselo directamente al Vicerrector Económico, éste la revisa y si es necesario hace modificaciones y luego este último se le envía a la Rectora para que la apruebe, de igual manera, cuando la recibe el Vicerrector de Ciudad la revisa y si es necesario la modifica y se le envía al Vicerrector Económico y luego se la envía a la Rectora. La Rectora revisa y si se encuentra bien la aprueba, o la envía para la Dirección de Transporte por correcciones y comienza de nuevo todo el proceso hasta que la asignación del combustible se encuentre correctamente elaborada. Después de estar correctamente elaborada la Rectora aprueba la asignación y se la envía a la Dirección de Transporte. El Especialista de Transporte envía la asignación aprobada a la Caja Central y así termina este proceso (*Ver Figura 6*).

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

- **Realizar resumen de consumo del combustible por tipo de actividad:** El proceso de Realizar resumen de consumo del combustible por tipo de actividad comienza cuando el Especialista de la Caja Central notifica a la Dirección de Transporte el cargue de las tarjetas magnéticas. La Dirección de Transporte notifica a los Choferes y Responsables que se presenten en sus oficinas con el resumen de consumo de su vehículo para entregarle la nueva Certificación del combustible habilitado y kilómetros recorridos del mes. En caso de que el resumen de consumo del vehículo tenga incongruencias se le entrega nuevamente al chofer o al responsable para realizar correcciones. Luego se registra el consumo de los vehículos, para posteriormente realizar el resumen de consumo del combustible por tipo de actividad. Este resumen es enviado a la Dirección de Energía, y así termina este proceso (Ver Figura 7).

Procesos de la Vicerrectoría Económica

En el módulo de la Vicerrectoría Económica se desarrollan dos procesos del negocio: Aprobar distribución del combustible y Aprobar entrega del combustible adicional (Ver Figura 8).

- **Aprobar distribución mensual del combustible:** El proceso de Aprobar distribución mensual del combustible comienza cuando el Especialista de la Vicerrectoría recibe la Asignación Mensual de Combustible de la Dirección de Transporte o de la Vicerrectoría de la Ciudad para corregirla en caso de que tengas errores y enviarla luego a la Rectora para que sea aprobada y así termina el proceso (Ver Figura 9).
- **Aprobar entrega de combustible adicional:** El proceso de Aprobar la distribución del combustible adicional comienza cuando el responsable o chofer elabora y entrega la solicitud del combustible adicional en la Vicerrectoría Económica. El especialista de la Vicerrectoría Económica puede no aprobar esta solicitud y así termina este proceso, pero si aprueba la solicitud le entrega el modelo de entrega de Tarjeta Magnética al chofer o el responsable del vehículo, terminando así el proceso (Ver Figura 10).

2.3. - Mapas de los procesos

- **Grupo de Gestión Energética**

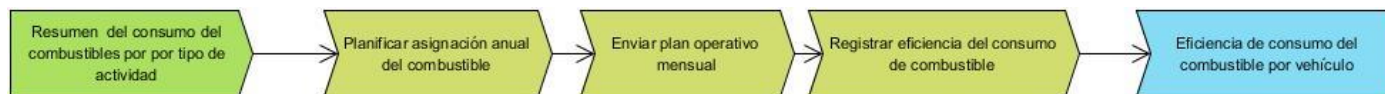


Figura 11: Mapa del proceso del Grupo de Gestión Energética.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

- **Dirección de Transporte**

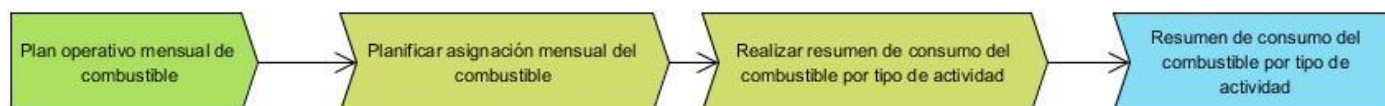


Figura 12: Mapa del proceso de la Dirección de Transporte.

- **Vicerrectoría Económica**



Figura 13: Mapa del proceso de la Vicerrectoría Económica.

2.4.- Propuesta del sistema

Con el propósito de automatizar los procesos de distribución y consumo del combustible en la UCI, se propone la realización del diseño e implementación de un sistema informático que controle el flujo de la información del proceso y eleve la calidad del mismo. El sistema presentará las funcionalidades: Planificar asignación anual del combustible, Enviar el plan operativo a la Dirección de Transporte, Planificar asignación mensual del combustible, Generar informe de eficiencia del consumo del combustible, Registrar el consumo del combustible de los vehículos de la Universidad, Realizar resumen de consumo del combustible por tipo de actividad, Aprobar distribución del combustible mensual y anual, Solicitar combustible adicional y Aprobar la entrega del combustible adicional.

2.5.- Requisitos funcionales y no funcionales del sistema

La captura de requisitos es una de las actividades fundamentales que se desarrolla en la fase de Inicio, los mismos pueden dividirse en requerimientos funcionales y no funcionales. Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable [36].

2.5.1.- 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:

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

- **Entrevista:** Es una técnica muy aceptada dentro de la ingeniería de requisitos y su uso está ampliamente extendido. Las entrevistas le permiten al analista tomar conocimiento del problema y comprender los objetivos de la solución buscada [36]. A través de encuentros planificados con especialistas del área de Transporte de la UCI se realizaron preguntas relacionadas con el proceso de distribución y consumo del 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.
- **Tormenta de ideas:** Es una técnica de reuniones en grupo cuyo objetivo es que los participantes muestren sus ideas de forma libre y ofrecer una visión general de las necesidades del sistema. 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 sistema [37].

2.5.2.- Requisitos funcionales

El sistema está compuesto por 38 requisitos funcionales: 22 con prioridad Alta, 10 con prioridad Media y 6 con prioridad Baja (*Ver Tabla 1*).

2.5.3.- Requisitos no funcionales

Luego de analizar las condiciones que resultan apropiadas para el funcionamiento de la propuesta de solución planteada, se identificaron los requerimientos no funcionales (RnF) que no son más que aquellas características que el sistema debe cumplir para garantizar el despliegue y soporte del mismo [36] .

-Usabilidad:

- **RnF 1:** El sistema debe poseer una arquitectura simple y que posibilite a los usuarios llegar al contenido que desea en un corto tiempo. Siempre que no fuerce la estructura del sistema, ninguna página debe encontrarse a más de tres clic de la página de inicio.
- **RnF 2:** El sistema debe permitirle a los usuarios con pocos conocimientos de informática poder interactuar con el sistema.
- **RnF 3:** Debe poseer una interfaz intuitiva y fácil de navegar.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

-Disponibilidad:

- **RnF 4:** El sistema debe estar disponible las 24 horas del día los 7 días de la semana.

-Confiabilidad:

- **RnF 5:** Los reportes que se obtendrán serán 100% reales y precisos.
- **RnF 6:** La información no podrá ser modificada por ningún usuario no autorizado, protegiendo así la integridad de los datos.

-Fiabilidad:

- **RnF 7:** El sistema debe mostrarle al usuario un mensaje indicándole que ha ocurrido un fallo en la operación que se realice.

-Legales:

- **RnF 8:** La mayoría de las herramientas de desarrollo son libres y el resto de las licencias están avaladas.
- **RnF 9:** Documentación legal de uso como Declaración de Autoría.

-Soporte:

- **RnF 10:** El sistema debe dar la posibilidad de ser mejorado, así como de incorporarle nuevas funcionalidades, en caso de ser necesarias.

-Hardware:

Servidores de Aplicación y BD:

- **RnF 11:** Para garantizar un funcionamiento óptimo del sistema, el servidor donde estará desplegado el mismo deberá ser Corei3, con una velocidad del procesador de 3.00 GHz, 3 GB de RAM y 160 GB de disco duro. El servidor de base de datos debe ser Core 2duo, con 1GB de RAM y 160 GB de disco duro.

PC cliente:

- **RnF 12:** Para el cliente como requerimientos mínimos: Procesador Pentium III a 2.8 GHz con 512 Mb de memoria RAM y una tarjeta de red.

-Software:

- **RnF 13:** Para acceder a la aplicación se requiere usar un Navegador Mozilla Firefox desde la versión 16 o superior, y se requiere un Sistema operativo Nova 3.0 o cualquier otra distribución de Linux y Microsoft Windows XP o superior.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

Servidores de Aplicación:

- **RnF 14:** Para un servidor de aplicación Web se requiere usar Apache versión 2.0 o superior con módulo PHP 5.0 v5.4.3 disponible, este debe estar configurado con las extensiones “ldap” y “pgsql”; y un servidor de base de datos PostgreSQL 9.1 o superior.

Servidores de BD:

- **RnF 15:** Sistema Gestor de base de datos PostgreSQL 9.1.

-Seguridad

- **RnF 16:** El sistema le permitirá al usuario las funcionales en dependencia del rol que posean en el sistema.
- **RnF 17:** Se utilizará el protocolo HTTPS para la comunicación entre el cliente y el servidor en los procesos de envío de datos entrados por el usuario en la autenticación y en las tareas administrativas y de gestión de contenidos.

-Portabilidad:

- **RnF 18:** El sistema será multiplataforma, con el correcto funcionamiento en Linux y Windows.

2.5.4.- 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 [36]. Para la validación de los requisitos funcionales fue aplicada la siguiente técnica:

- **Revisión técnica formal:** Es la técnica más efectiva desde el punto de vista de aseguramiento de la calidad, y es dirigida por los ingenieros de software u otras personas. Para los ingenieros la inspección es un medio efectivo para descubrir errores y mejorar la calidad del software. La revisión de las especificaciones de los requerimientos del proceso de Distribución y consumo del combustible es realizada con especialistas de las áreas que forma parte de este proceso. 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 [38].

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

- **Prototipado:** Esta técnica consiste en construir una maqueta del futuro sistema de software a partir de los requisitos descritos en la especificación. Esta maqueta será evaluada por el cliente y usuarios para comprobar su corrección y completitud [36]. A partir de las especificaciones de los requisitos funcionales fueron conformados prototipos de interfaz de usuario. De esta forma, se validó que los requerimientos funcionales estaban en concordancia con las expectativas de los clientes. 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 de Distribución y consumo del combustible.

2.6.- Modelo de casos de uso del sistema

2.6.1.- Descripción de los actores

Un actor del sistema es un rol que un usuario juega con respecto al sistema, es un agente externo que intercambia información con el sistema y se beneficia del resultado del sistema [40]. Los actores definidos para el sistema en cuestión son:

Actor	Justificación
Administrador	Es la persona que tiene todos los privilegios sobre el sistema, realizando la gestión de usuarios y roles para definir el nivel de accesibilidad de los usuarios en el sistema.
Especialista Energía	Es el encargado de revisar los documentos desarrollados en el Grupo de Gestión Energética; también puede adicionar, modificar, asignar y eliminar cualquier dato o modelo dentro de este departamento.
Especialista Transporte	Es el encargado de revisar los documentos desarrollados en el Departamento de Transporte, también puede adicionar, modificar, asignar y eliminar cualquier dato o modelo dentro de este departamento.
Especialista Vicerrectoría Económica	Es el encargado de revisar finalmente los documentos desarrollados en la Vicerrectoría Económica, para la aprobación de las planificaciones mensuales y anuales del combustible y la entrega del combustible adicional.
Especialista Vicerrectoría Ciudad	Es el encargado de revisar la asignación mensual del combustible.
Responsable	Es la persona responsable del equipo al cual se le permite solicitar combustible adicional.

Tabla 2: Descripción de los actores del sistema.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

2.6.2.- Diagrama de casos de uso

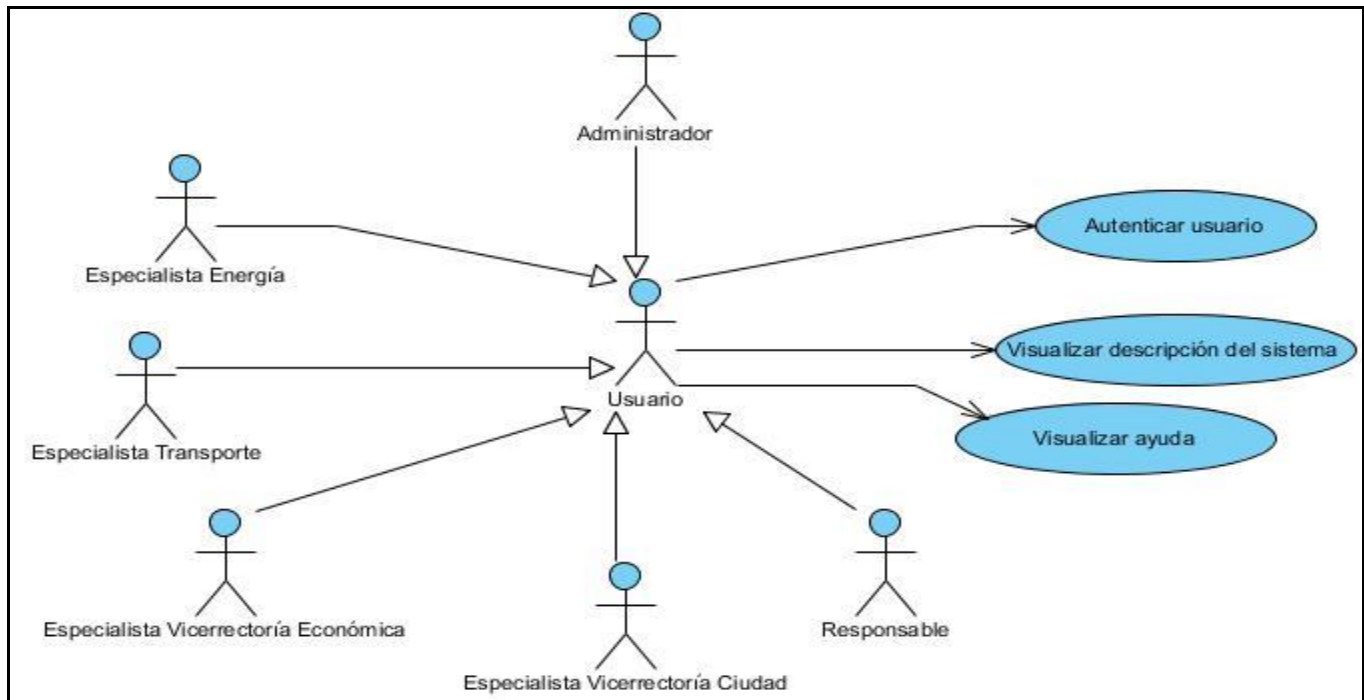


Figura 14: Diagrama casos de uso múltiples usuarios con las mismas responsabilidades.

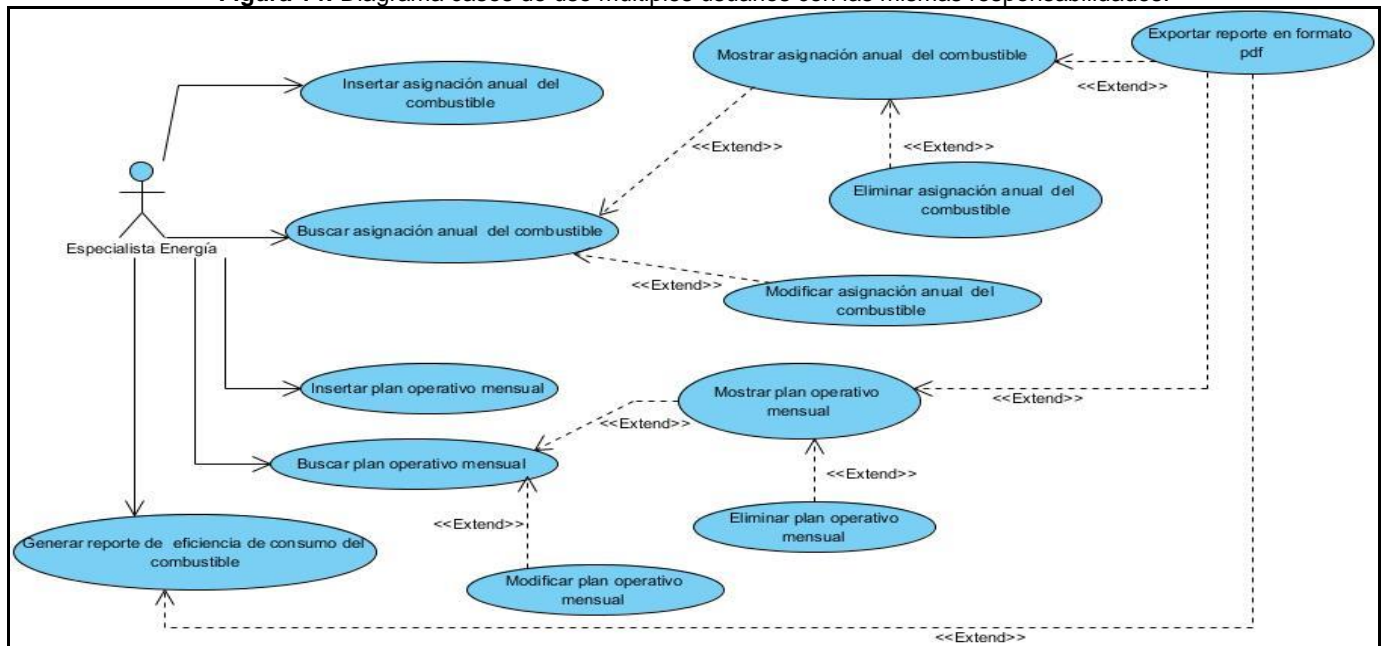


Figura 15: Diagrama del caso de uso del sistema del Módulo de Energía.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

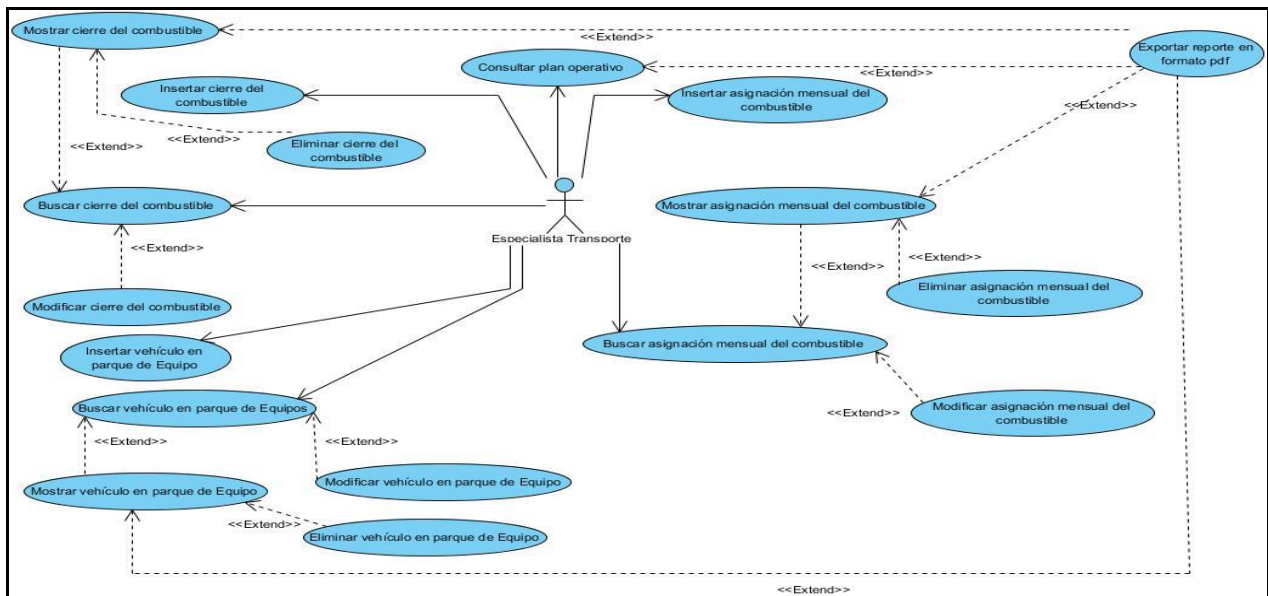


Figura 16: Diagrama del caso de uso del sistema del Módulo de Transporte.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

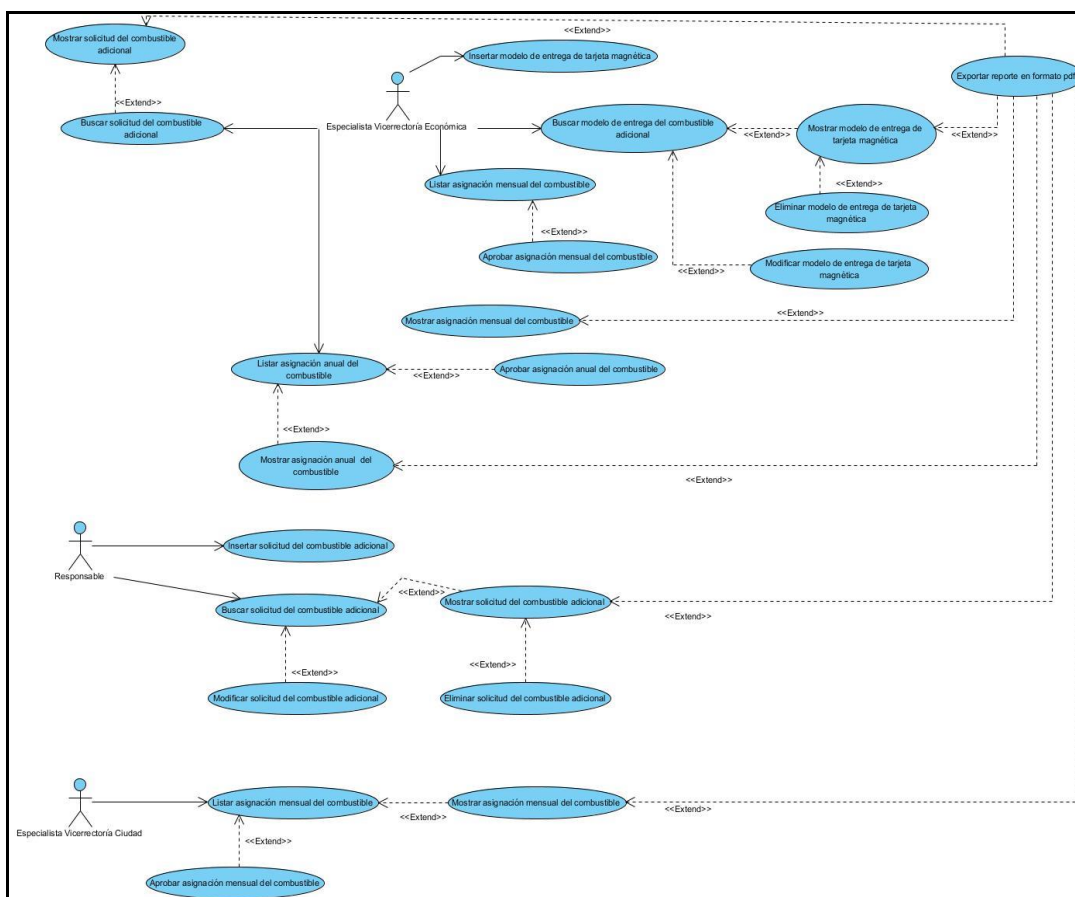


Figura 17: Diagrama del caso de uso del sistema del Módulo de la Vicerrectoría Económica.

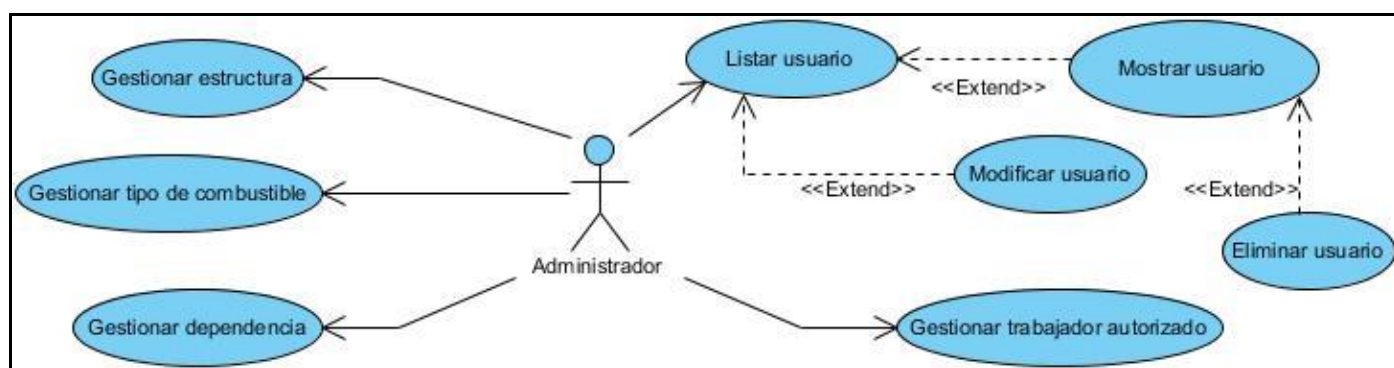


Figura 18: Diagrama del caso de uso del sistema del Módulo de Configuración.

2.6.3.- Descripción de casos de uso

Las descripciones de caso de usos se encuentran en los anexos del documento.

CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

2.7.- Conclusiones

- La modelación del negocio permitió conocer el proceso de distribución y consumo del combustible en la Universidad e identificar los actores del negocio, los casos de uso del negocio y la descripción de los mismos para poder determinar el entorno de negocio para darle solución al problema planteado.
- La identificación de los requisitos funcionales y no funcionales del sistema posibilitó una mejor visión de la implementación y funcionamiento de la solución.
- La revisión técnica formal realizada permitió comprobar que la elaboración de los artefactos generados como parte de la modelación del proceso de negocio del sistema de gestión del combustible es correcta, y que el flujo de actividades de cada proceso coincide con la información brindada.
- A partir de la aplicación de las 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 se corresponden con las expectativas del cliente. Como constancia de las revisiones realizadas fue emitida un Acta de aceptación para cada una de las validaciones efectuadas.

CAPÍTULO III: DISEÑO DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

Capítulo III: Diseño del Sistema de gestión de la información del combustible

3.1.- Introducción

En el capítulo se desarrollará el Diseño de las principales funcionalidades a implementar en el desarrollo del Sistema de gestión del combustible. Se realizarán los diagramas más representativos como son: diagramas de clases del diseño y los diagramas de interacción, concluyendo en el modelo de diseño de la base de datos y el modelo lógico de datos que sienta las bases para la implementación. Se define la Arquitectura del sistema a través de los patrones de diseños para lograr un diseño acorde con estándares.

3.2.- Modelo del Diseño

El modelo del diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas en el entorno de la implementación, tienen un impacto en el sistema a considerar y sirven de abstracción a la implementación y al código fuente del sistema [41].

3.2.1.- Diagramas de clases del diseño

A continuación, se muestra el diagrama de clases del diseño del caso de uso “**Buscar Plan Operativo Mensual**”, los restantes diagramas de clases del diseño se pueden encontrar en los anexos del documento.

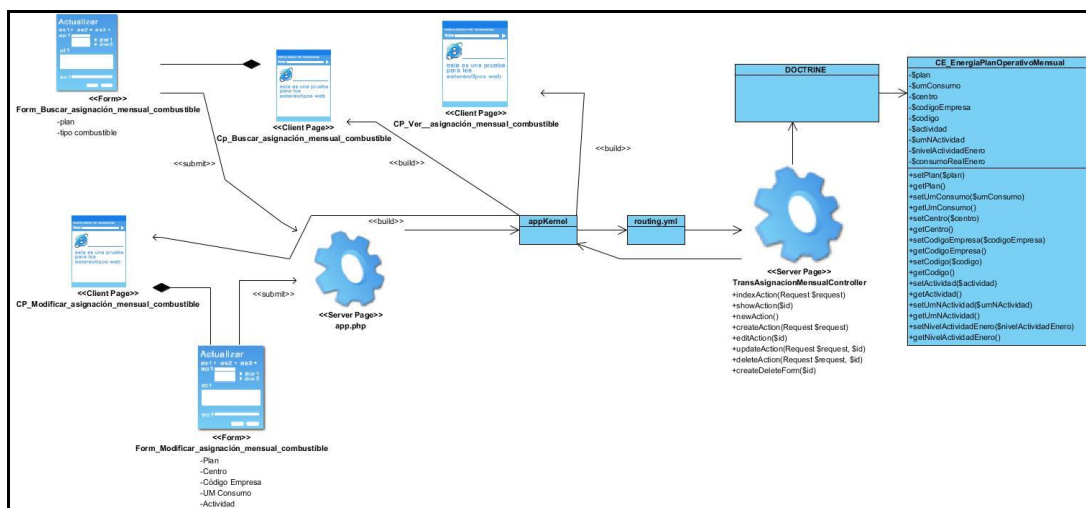


Figura 19: Diagrama de clases del diseño del Caso de Uso “Buscar plan operativo mensual”.

CAPÍTULO III: DISEÑO DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

3.2.2.- Diagramas de Interacción

Los diagramas de interacción modelan los aspectos dinámicos de un sistema, explican gráficamente a través de mensajes, las interacciones existentes entre las instancias de las clases. UML define dos tipos de diagramas de interacción que son: diagrama de colaboración y diagrama de secuencia.

Diagrama de secuencia

Un diagrama de secuencia del sistema es un artefacto creado de manera rápida y fácil, que muestra los eventos de entrada y salida relacionados con el sistema que se está estudiando. Es una notación que puede representar las interacciones de los actores y las operaciones que inician [42]. A continuación, se representan los diagramas de secuencia del caso de uso: Buscar plan operativo mensual, utilizando el patrón arquitectónico MVC, los restantes se pueden encontrar en los anexos del documento.

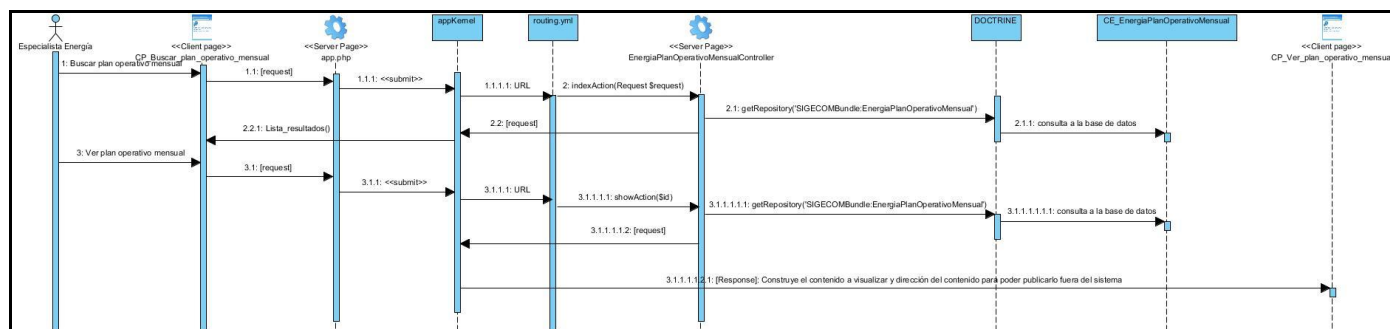


Figura 20: Diagrama de secuencia del escenario “Ver plan operativo mensual” del Caso de Uso “Buscar plan operativo mensual”.

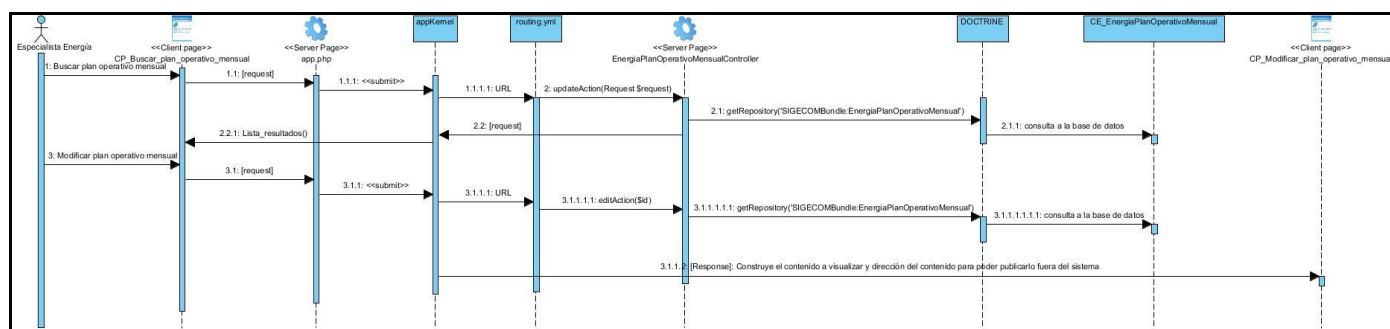


Figura 21: Diagrama de secuencia del escenario “Modificar plan operativo mensual” del Caso de Uso “Buscar plan operativo mensual”.

3.2.3.- Descripción de las clases del diseño

Nombre de la Clase	CP_Buscar_plan_operativo_mensual
Descripción General	
La página cliente CP_Buscar_plan_operativo_mensual permite realizar una búsqueda de los planes operativos mensuales del combustible registrados por el Especialista de Energía.	
Nombre de la Clase	CP_Modificar_plan_operativo_mensual

CAPÍTULO III: DISEÑO DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

Descripción General	
La página cliente CP_Modificar_plan_operativo_mensual permite realizar modificaciones a los planes operativos mensuales del combustible registrados por el Especialista de Energía.	
Nombre de la Clase	CP_Ver_plan_operativo_mensual
Descripción General	
La página cliente CP_Ver_plan_operativo_mensual permite visualizar los planes operativos mensuales del combustible registrados por el Especialista de Energía.	
Nombre de la Clase	CP_Eliminar_plan_operativo_mensual
Descripción General	
La página cliente CP_Eliminar_plan_operativo_mensual permite eliminar los planes operativos mensuales del combustible registrados por el Especialista de Energía.	
Nombre de la clase	EnergiaPlanOperativoMensualController
Descripción General	
La clase controladora EnergiaPlanOperativoMensualController permite la interacción entre las clases vistas y las clases modelos.	
Nombre de la Clase	CEEnergiaPlanOperativoMensual
Descripción General	
La clase entidad CE_EnergiaPlanOperativoMensual permite acceder a los datos persistentes de la base de datos.	
Nombre de la Clase	app
Descripción General	
La clase servidora app permite realizar peticiones al kernel de Symfony a partir de las solicitudes del Especialista de Energía.	
Nombre de la Clase	appKernel
Descripción General	
La clase appKernel contiene las declaraciones del kernel de Symfony.	
Nombre de la Clase	routing.yml
Descripción General	
La clase routing se encarga de enrutar todas las peticiones que se realizan en el sistema.	
Nombre de la Clase	DOCTRINE
Descripción General	
Es una representación de la abstracción de la base datos del sistema, ya que se utiliza Doctrine para acceder a los datos de las entidades de la base de datos.	

Tabla 3: Descripción de las clases del diseño.

CAPÍTULO III: DISEÑO DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

3.2.4.- Modelo físico de la base de datos

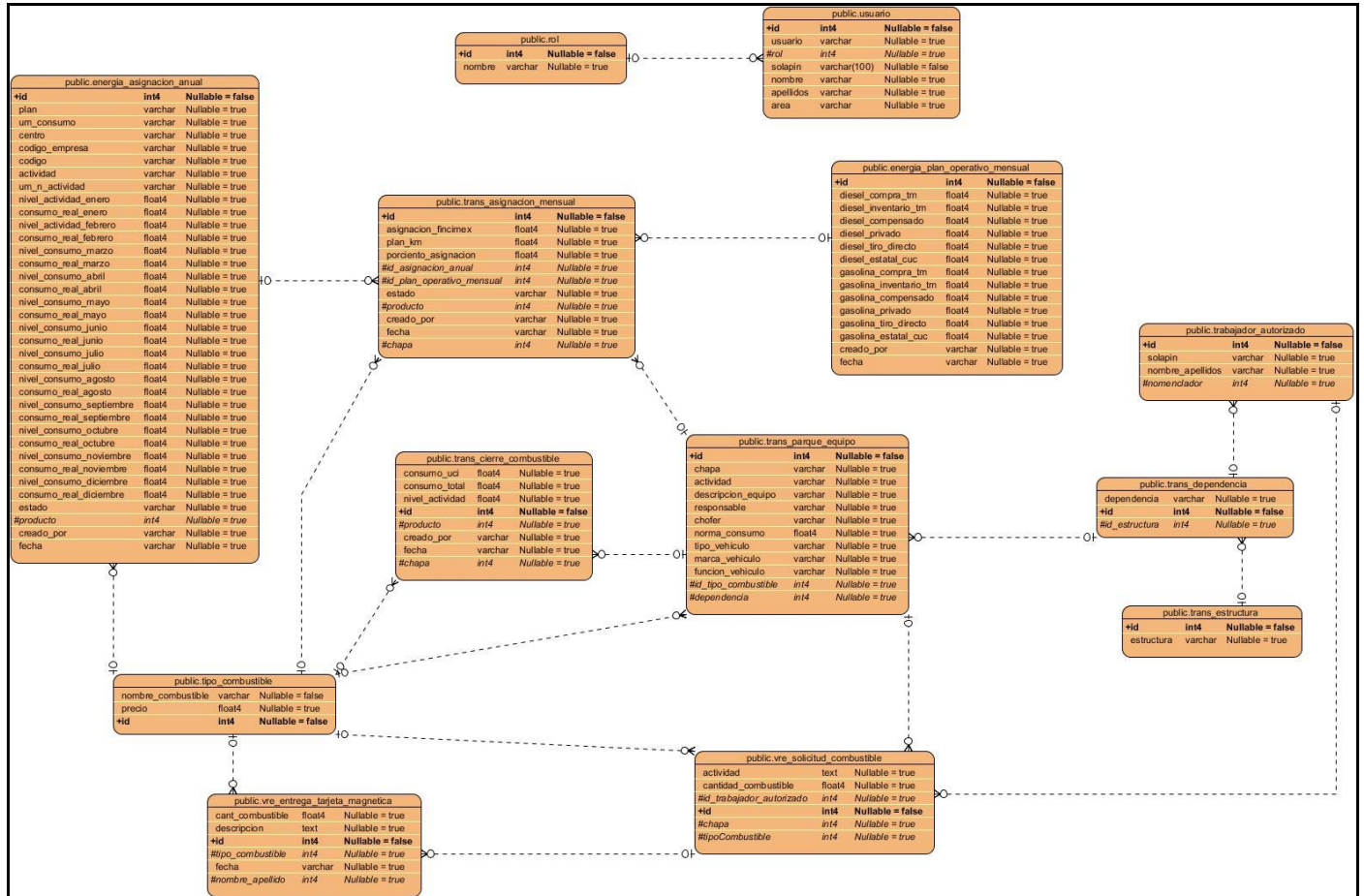


Figura 22: Diagrama del Modelo Entidad Relación de la base de datos.

3.2.5.- Descripción de las entidades

Tabla	Descripción
rol	Esta tabla almacena los roles que serán otorgados a los usuarios en el sistema.
usuario	Esta tabla almacena los datos de los usuarios que acceden al sistema.
energía_asignacion_anual	Esta tabla almacena los datos asociados a la asignación anual del combustible en el sistema.
energía_plan_operativo_mensual	Esta tabla almacena los datos asociados al plan operativo mensual del combustible en el sistema.
trans_asignacion_mensual	Esta tabla almacena los datos asociados a la asignación mensual del combustible en el sistema.
trans_parque_equipo	Esta tabla almacena los datos de cada equipo registrado en el sistema.
trans_cierre_combustible	Esta tabla almacena los datos asociados al cierre del combustible por equipo.
tipo_combustible	Esta tabla almacena los datos asociados al tipo de combustible y el

CAPÍTULO III: DISEÑO DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

	precio asociado al mismo en el sistema.
trabajador_autorizado	Esta tabla almacena los datos de cada responsable y chofer asociado a un vehículo en el sistema.
vre_solicitud_combustible	Esta tabla almacena los datos asociados a las solicitudes realizadas al Vicerrector de Economía del combustible adicional en el sistema.
vre_entrega_tarjeta_magnetica	Esta tabla almacena los datos asociados a las entregas realizadas por el Vicerrector de Economía del combustible adicional en el sistema.
trans_estructura	Esta tabla almacena los datos asociados a la estructura organizacional de la Universidad en el sistema.
trans_dependencia	Esta tabla almacena los datos asociados a la dependencia organizacional de la Universidad en el sistema.

Tabla 4: Descripción de las entidades de la base de datos.

3.3.- Modelo de Despliegue

El Diagrama de Despliegue representa la distribución 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 [43]. A continuación, se presenta el modelo de despliegue elaborado para el Sistema de gestión del combustible de la UCI:

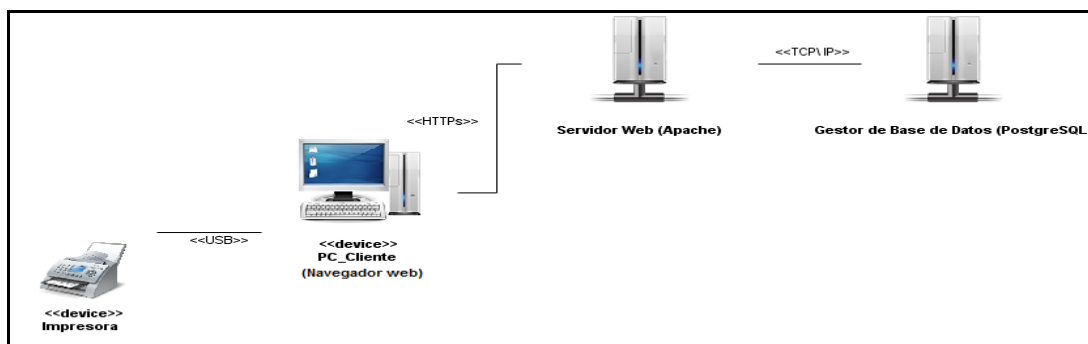


Figura 23: Modelo de despliegue del sistema.

3.3.1.- Descripción de los nodos

- **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 bibliotecas 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.

CAPÍTULO III: DISEÑO DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

- **Impresora:** Utilizada para imprimir los reportes del Sistema de gestión de la información del combustible de la UCI.

3.3.2.- Descripción de elementos e interfaces de comunicación

- **TCP/IP:** Es la base de Internet, y sirve para enlazar computadoras que utilizan diferentes sistemas operativos, incluyendo PC, minicomputadoras y computadoras centrales sobre redes de área local (LAN) y área extensa (WAN) [45].
- **HTTPS:** HTTPS (por sus siglas en inglés *Hypertext Transfer Protocol Secure*, *Protocolo seguro de transferencia de hipertexto*) a través de SSL (por sus siglas en inglés *Secure Socket Layer*, *Capa de conexión segura*). Se trata de un protocolo TCP/IP utilizado por los servidores web para la transferencia y visualización del contenido web de forma segura. Los datos transferidos se cifran por lo que no puede ser leída por nadie excepto el destinatario [44].
- **USB:** USB (por sus siglas en inglés *Universal Serial Bus*, *Bus Universal en Serie*), es un tipo estándar de conexión para muchos tipos de dispositivos. Generalmente, se refiere a los tipos de cables, puertos y conectores utilizados para conectar los dispositivos externos a los ordenadores como: teclados, teléfonos, escáneres e impresoras [46].

3.4.- Arquitectura y patrones de diseño utilizados

La arquitectura de software está afectada no solo por la estructura y el comportamiento, sino también por el uso, la funcionalidad, el rendimiento, la flexibilidad, la reutilización, la facilidad de comprensión, las restricciones y compromisos económicos, tecnológicos y estéticos. Además, abarca decisiones importantes sobre:

- La organización del sistema de software.
- Los elementos estructurales que compondrán el sistema y sus interfaces, junto con sus comportamientos, tal y como se especifican en las colaboraciones entre estos elementos.
- La composición de los elementos estructurales y del comportamiento en subsistemas progresivamente más grandes [47].

3.4.1.- Arquitectura de software

Patrón arquitectónico Modelo-Vista-Controlador (MVC)

Para el desarrollo del Sistema de gestión del combustible se decide utilizar el patrón MVC. Este patrón divide la aplicación en tres componentes: el Modelo, la Vista y el Controlador. El Modelo se encarga de

CAPÍTULO III: DISEÑO DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

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. La Vista maneja la visualización de la información, mientras que el Controlador analiza los mensajes de eventos que recibe del sistema, modifica u obtiene datos de la clase Modelo en respuesta a las peticiones del usuario. En resumen, se encarga de asimilar las peticiones del usuario y realizar los cambios apropiados en el modelo o en la vista. En la siguiente figura se observa la interacción que se efectúa entre las distintas capas, específicamente en el framework Symfony, que es el que se emplea en el trabajo:

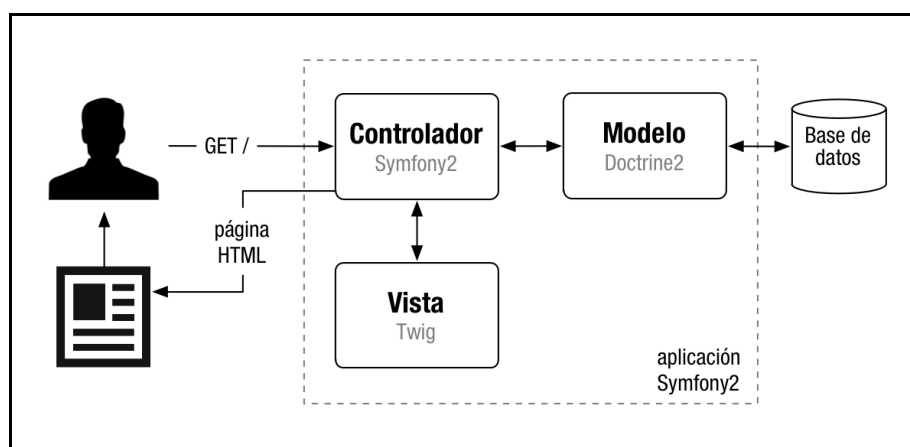


Figura 24: Modelo Vista Controlador.

Las capas del MVC que implementa Symfony, está basado en un patrón clásico de arquitectura conocido como arquitectura MVC, que está formado por tres niveles:

- El Modelo representa la información con la que trabaja la aplicación, es decir, su lógica de negocio: la abstracción de la base de datos y el acceso a los datos.
- La Vista transforma el modelo en una página web que permite al usuario interactuar con ella: la vista, la plantilla y el layout.
- El Controlador se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista: el controlador frontal y las acciones.

En total, cuenta con siete scripts pero Symfony simplifica este proceso, toma lo mejor de la arquitectura MVC y la implementa de forma que el desarrollo de aplicaciones sea rápido y sencillo. El controlador frontal y el layout son comunes para todas las acciones de la aplicación. El controlador frontal es un componente que sólo tiene código relativo al MVC, por lo que no es necesario crear uno, ya que Symfony lo genera de forma automática. Doctrine se encarga de generación de las clases de la capa del modelo, en función de la estructura de datos de la aplicación, y crea la estructura básica de las clases y genera automáticamente el código necesario. Cuando Doctrine encuentra restricciones de claves foráneas (o

CAPÍTULO III: DISEÑO DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

externas) o cuando encuentra datos de tipo fecha, crea métodos especiales para acceder y modificar esos datos, por lo que la manipulación de datos es más sencilla. La abstracción de la base de datos es completamente transparente para el programador, ya que se realiza de forma nativa mediante PDO (por sus siglas en inglés: *PHP Data Objects*), por lo que al cambiar el sistema gestor de bases de datos en cualquier momento, no se debe reescribir ni una línea de código, ya que tan sólo es necesario modificar un parámetro en un archivo de configuración [48].

3.4.2.- Patrones de diseño empleados

Un patrón de diseño es una descripción de clases y objetos comunicándose entre sí, adaptada para resolver un problema de diseño general en un contexto particular, y brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Los patrones son una pareja problema/solución con un nombre, que codifican buenos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades [49]. Entre los patrones que se utilizarán se encuentran los patrones generales de software para la asignación de responsabilidades (*GRASP* por sus siglas en inglés *General Responsibility Assignment Software Patterns*, *Patrones Generales de software para asignación de responsabilidades*) y los patrones GoF (por sus siglas en inglés *Gang Of Four*).

Los Patrones GRASP tienen una importante utilidad en el diseño realizado. Describen los principios fundamentales para asignar responsabilidades a los objetos [50]. Los utilizados en el diseño del sistema son los siguientes:

- **Experto:** El patrón se utiliza para la asignación de responsabilidades a las clases, de forma tal que las mismas contengan la información necesaria para poder ejecutar una acción determinada. Permite a los objetos valerse de su propia información para realizar funcionalidades y favorece la existencia de mínimas relaciones entre las clases. Este patrón se evidencia en la clase *TransAsignacionMensualController*, como experta en el dominio de la información de las asignaciones mensuales de combustibles registradas en el sistema.
- **Bajo Acoplamiento:** El patrón de acoplamiento evalúa la fuerza con que una clase está conectada a otra, de esta forma una clase con bajo acoplamiento debe tener un número mínimo de dependencias con otras clases. Este patrón se evidencian en las clases *TransParqueEquipo* y *TipoCombustible*, las cuales tienen una débil dependencia, y cualquier cambio en una de ellas no afectaría a la otra.
- **Controlador:** El patrón otorga la responsabilidad de controlar el flujo de eventos del sistema en varias clases con las que tiene una alta cohesión. Por lo que no existe una sola clase controladora

CAPÍTULO III: DISEÑO DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

que realice todas las actividades, garantizando de esta forma mejorar las validaciones y seguridad en el sistema. Este patrón se evidencia en la clase TransParqueEquipoController, la cual posee responsabilidades específicas [42].

Patrones de diseño GOF

Los patrones GoF describen soluciones simples y elegantes a problemas específicos en el diseño de software orientado a objetos y se agrupan en tres grandes categorías: creacionales, estructurales y de comportamiento [51]. Los utilizados en el diseño del sistema son los siguientes:

- **Decorador:** Symfony presenta el denominado archivo `base.html.twig` o también conocido como plantilla global, en la que convergen todos los elementos comunes a cada una de las páginas del sistema en construcción. Este fichero se complementa con las plantillas, decorándolas y obteniéndose la interfaz final que será mostrada al usuario.
- **Command (Orden):** La estrategia del patrón Command sugiere proporcionar una interfaz genérica para los componentes en los que el controlador puede delegar responsabilidades, minimizando el acoplamiento entre estos componentes. En el framework Symfony se utiliza este patrón, el cual está representado por las acciones, por tanto cada acción, encapsula una petición en un objeto. En el sistema se evidencia el uso de este patrón al realizar la acción de exportar a formato `.pdf`, a través del cual se construye el documento pdf y se exporta.
- **Front controller (Controlador frontal):** Es el único punto de entrada a la aplicación. Carga la configuración y determina la acción a ejecutarse. Se encarga de despachar las peticiones, lo que implica algo más que detectar la acción que se ejecuta. De hecho, ejecuta el código común a todas las acciones, incluyendo:
 1. Carga la clase de configuración del proyecto y las librerías de Symfony.
 2. Crea la configuración de la aplicación y el contexto de Symfony.
 3. Carga e inicializa las clases del núcleo del framework.
 4. Carga la configuración.
 5. Decodifica la URL de la petición para determinar la acción a ejecutar y los parámetros de la petición.
 6. Si la acción no existe, redireccionará a la acción del error 404.
 7. Activa los filtros (por ejemplo, si la petición necesita autenticación).
 8. Ejecuta los filtros, primera pasada.
 9. Ejecuta la acción y produce la vista.
 10. Ejecuta los filtros, segunda pasada.

CAPÍTULO III: DISEÑO DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

11. Muestra la respuesta.

Patrones de diseño de casos de uso

Son comportamientos que deben existir en el sistema, ayudan a describir qué es lo que el sistema debe hacer, es decir, describen el uso del sistema y cómo este interactúa con los usuarios. Estos patrones son utilizados generalmente como plantillas que describen como debería ser estructurados y organizados los casos de uso. Son patrones que capturan mejores prácticas para modelar casos de uso (Larman, 2008). Los utilizados en el diseño del sistema son los siguientes:

- **CRUD (por sus siglas en inglés Create, Read, Update and Delete, Crear, Obtener, Actualizar y Borrar):** El patrón CRUD consiste en un caso de uso para administrar la información (CRUD Información), nos permite modelar las diferentes operaciones para administrar una entidad de información, tales como crear, leer, cambiar y eliminar o dar de baja. Este patrón deberá ser usado cuando todas las operaciones contribuyen al mismo valor de negocio y todas son cortas y simples [42].
- **Patrón Extensión Concreta:** Consiste en dos casos de uso y una relación extendida entre ellos. Puede ser realizada en sí mismo, así como extendido en el caso de uso base. El referente puede ser concreto o abstracto. Este patrón se aplica cuando un flujo puede extender el flujo de otro caso de uso así como ser realizado en sí mismo [42].
- **Patrón Extensión Inclusión:** Consiste en dos casos de uso y una relación incluida entre ellos. Puede ser realizada en sí mismo, así como del caso de uso base al caso de uso de inclusión. El último puede ser instalado en sí mismo. El caso de uso base puede ser concreto o abstracto. Este patrón se aplica cuando un flujo puede incluir el flujo de otro caso de uso [42].
- **Múltiples Actores:** El patrón es utilizado al existir dos actores que jueguen el mismo rol sobre el caso de uso. Este rol es representado por otro actor, heredado por los actores que comparten este rol, aplicándolo cuando, desde el punto de vista del caso de uso, solo exista una entidad externa interactuando con cada una de las instancias del caso de uso [42].

3.5.- Conclusiones

Como parte del desarrollo del capítulo se determinan las siguientes conclusiones parciales:

- Los artefactos generados según la metodología de desarrollo de software utilizada y los patrones de arquitectura y diseño descritos, permitieron guiar el desarrollo de la solución propuesta.
- El diagrama de despliegue permitió representar la composición de los elementos físicos del sistema.

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

Capítulo IV: Implementación y pruebas del Sistema de gestión de la información del combustible

4.1.- Introducción

En el capítulo se documentan los aspectos fundamentales de la implementación y validación de la propuesta de solución, centrando su desarrollo en el flujo de trabajo Implementación que parte del diseño propuesto en el capítulo anterior. Además se abarca la implementación de las clases y subsistemas encontrados durante el desarrollo, representados a través del diagrama de componente. Se explican los estándares de codificación empleados y las pruebas utilizadas para la validación de la solución y el resultado al aplicarlas.

4.2.- Modelo de implementación

El modelo de implementación describe la forma en que los elementos del modelo de diseño y las clases, se implementan en términos de componentes, ficheros de código fuente, ejecutables, librerías, entre otros. El diagrama de componentes conforma lo que se conoce como un modelo de implementación, al describir los componentes y sus relaciones utilizando patrón de arquitectura MVC [53].

4.2.1.- Diagrama de componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, sean estos componentes de código fuente, binarios o ejecutables. Normalmente contienen componentes, interfaces y relaciones entre ellos, y como todos los diagramas también puede contener paquetes utilizados para agrupar elementos del modelo [54]. En la siguiente figura se muestra el diagrama de componentes del sistema:

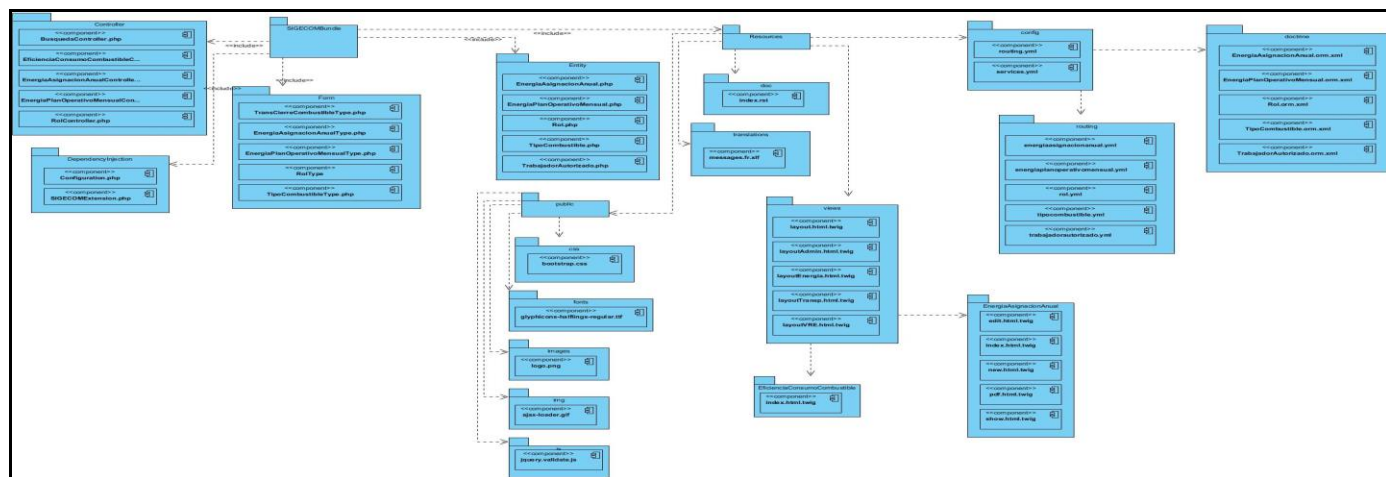


Figura 25: Diagrama de componentes del sistema.

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

Descripción de los componentes generales de la propuesta de solución:

- **SIGECOMBundle**: es el bundle que se encarga de la administración del sistema, roles, gestión de usuario y gestión de etiquetas, entre otras funcionalidades.
- **Controller**: contiene las clases controladoras del sistema que se encargan de la administración del sistema.
- **DependencyInjection**: se encarga de la comunicación de Symfony2 con los servicios implementados en el SIGECOMBundle.
- **Entity**: contiene las entidades relacionadas con la administración y los usuarios del sistema.
- **Form**: contiene los formularios que se generan para las operaciones de gestión.
- **Resources**: contiene las vistas, estilos y archivos javascript, los cuales se les muestra al usuario como respuesta del sistema

4.2.2.- Estándar y estilo de codificación

Los estándares y estilos de codificación permiten que el código fuente de la solución que se esté desarrollando posea mayor calidad, y que realizar el mantenimiento de la misma se torne menos complejo. Poseer un estilo de programación único permite que todos los involucrados puedan entender en un menor tiempo el código del sistema, y que sea más fácil de ofrecer mantenimiento al código del sistema [65]. Luego de analizar lo anteriormente planteado, y de efectuar un estudio de los estilos y estándares, se han identificado aquellos que serán utilizados para la implementación del sistema propuesto.

Identación		
Inicio y fin de bloque	Se recomienda dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque {}. Lo mismo sucede para el caso de las instrucciones if, else, for, while, do while, switch, foreach.	
Aspectos Generales	El indentado debe ser de dos espacios por bloque de código. No se debe usar el tabulador; porque este puede variar según la PC o la configuración de dicha tecla. Los inicios ({} y cierre (}) de ámbito deben estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción. Nunca colocar { en la línea de un código cualquiera, esto requiere una línea propia.	
Comentarios, separadores, líneas, espacios en blanco y márgenes		
Ubicación de comentarios	Al inicio de cada clase o función y al final de cada	Se recomienda comentar al inicio de la clase o función especificando el objetivo de la

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

	bloque de código.	misma.
Líneas en blanco	Se emplean antes y después de métodos, clases y estructuras.	Se recomienda dejar una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función.
Espacios en blanco	Entre operadores lógicos y aritméticos.	Se recomienda usar espacios en blanco entre estos operadores para una mayor legibilidad en el código. Ejemplo: asignatura = nomasignatura
Aspectos generales	Sobre el comentario:	Se debe evitar comentar cada línea de código. Cuando el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco. En caso de que se necesite comentar una sola instrucción se suprime la línea en blanco o se escribe a continuación de la instrucción.
	Sobre los espacios en blanco:	No se debe usar espacio en blanco: Después del corchete abierto y antes del cerrado, de un arreglo. Después del paréntesis abierto y antes del cerrado. Antes de un punto y coma. <ul style="list-style-type: none">No agregues espacios al final de las líneas.
Variables y constantes		
Aspectos generales	El nombre empleado debe permitir que con sólo leerlo se conozca el propósito de la misma.	
Clases y Objeto		
Apariencia de clases y objetos	Primera letra en mayúscula.	<ul style="list-style-type: none">Los nombres de las clases deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing. Ejemplo: MiClase ().Utiliza guiones bajos para definir opciones, argumentos y nombres de parámetros;Utiliza los namespace para todas las clases.Utiliza <i>Symfony</i> como el <i>namespace</i> de primer nivel;Añade como sufijo <i>Interface</i> a las interfaces.Utiliza caracteres alfanuméricos y guiones bajos para nombres de archivos.Los namespaces y las clases deben tener la siguiente estructura <Vendor name>(<Namespace>)*<Class Name>.Cada namespace debe tener un namespace

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

		<p>superior ("Vendor name").</p> <ul style="list-style-type: none"> • Cada namespace puede tener tantos sub-namespaces como se quiera. • Los nombres de los namespaces o clases deben ser separados por un guion bajo (_). • Todos los archivos deben tener la extensión .php. • Los nombres de los namespaces o clases deben ser ordenadas alfabéticamente. • Define una clase por archivo. • Declara las propiedades de las clases antes de los métodos. • Declara los métodos públicos primero, luego los protegidos y finalmente los privados.
Apariencia de atributos	Primera letra en minúscula.	El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamellCasing. Utiliza constantes de tipo PHP nativas en minúsculas: <i>false</i> , <i>true</i> y <i>null</i> . Lo mismo aplica para <i>array()</i> . Declara explícitamente la visibilidad de clases, métodos y propiedades (el uso de <i>var</i> está prohibido).
Apariencia de las funciones	Primera letra en mayúscula.	Para nombrar las funciones se debe tratar de utilizar verbos que denoten la acción que hace la función. Se empleará notación PascalCasing. Ejemplo: function BuscarUnidad (). Si son funciones que obtienen un dato se emplea el prefijo get y si fijan algún valor se emplea el prefijo set.
Aspectos generales	El nombre empleado para las clases, objetos, atributos y funciones debe permitir que con solo leerlo se conozca el propósito de los mismos.	

Tabla 5: Estándar y estilos de codificación del sistema.

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

4.4.- Pantallas principales de la aplicación

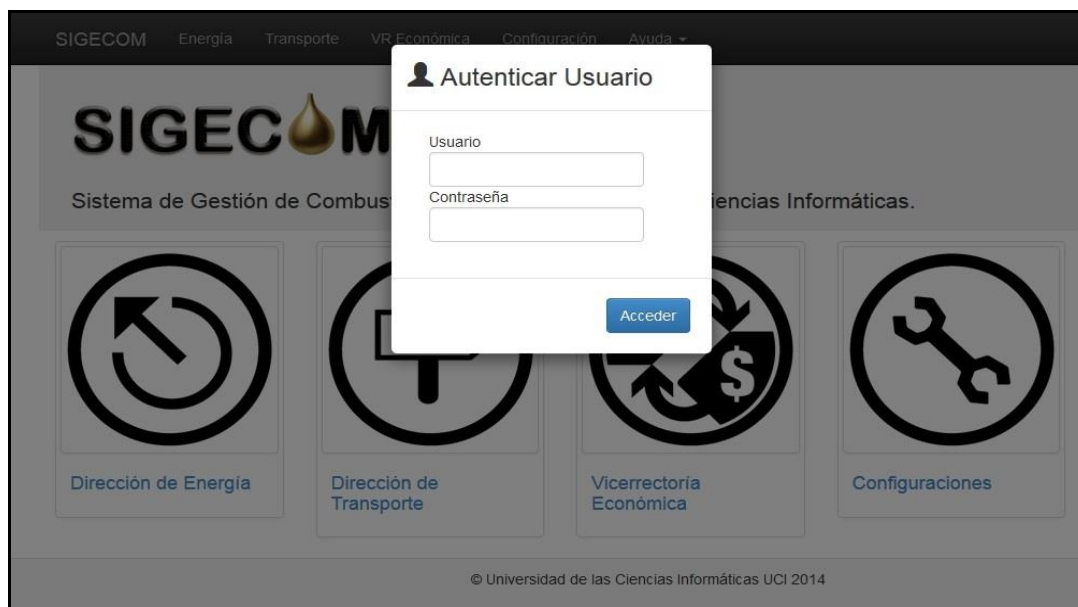


Figura 26: Pantalla de la vista autenticar usuario.



Figura 27: Pantalla de la vista del módulo de Energía.

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

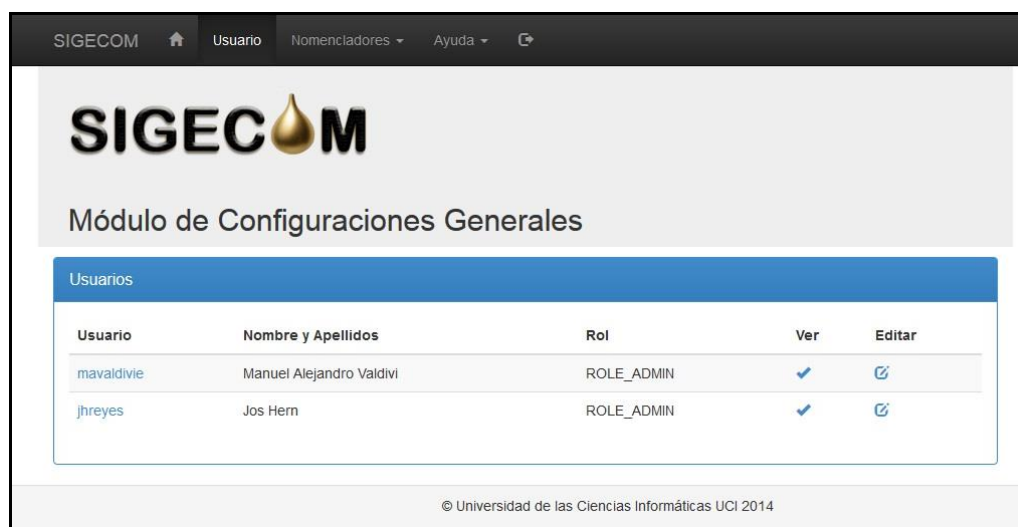


Figura 28: Pantalla de la vista de gestionar usuarios.

4.5.- Pruebas al sistema propuesto.

Para la validación del Sistema de Gestión de la Información del Combustible (SIGECOM) se realizaron pruebas funcionales, pruebas de rendimiento y pruebas de seguridad, las mismas tienen como objetivo de probar el correcto funcionamiento del sistema. Para la realización de estas pruebas se seleccionó el **nivel de prueba**:

- **Prueba de Unidad:** La prueba de unidad es la primera fase de las pruebas dinámicas y se realizan sobre cada módulo del software de manera independiente. El objetivo es comprobar que el módulo, entendido como una unidad funcional de un programa independiente, está correctamente codificado. En estas pruebas cada módulo será probado por separado y lo hará, generalmente, la persona que lo creó [66].
- **Prueba de Sistema:** las pruebas del sistema tienen como objetivo verificar la funcionalidad del sistema a través de sus interfaces externas, comprobando que dicha funcionalidad sea la esperada en función de los requisitos del sistema [55].

Utilizando los **tipos de prueba**:

- **Pruebas unitarias:** El objetivo es comprobar que el módulo, entendido como una unidad funcional de un programa independiente, está correctamente codificado. En estas pruebas cada módulo será probado por separado y lo hará, generalmente, la persona que lo creó [66].

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

- **Pruebas de funcionalidad:** Este tipo de prueba examina si el sistema cubre sus necesidades de funcionamiento, acorde a las especificaciones de diseño. En ellas se debe verificar si el sistema lleva a cabo correctamente todas las funciones requeridas, se debe verificar la validación de los datos y realizar las pruebas de comportamiento ante distintos escenarios [56].
- **Pruebas de seguridad:** La prueba de seguridad comprueba que los mecanismos de protección integrados en el sistema realmente lo protejan de irrupciones inapropiadas. En ellas se debe intentar conseguir las claves de acceso por cualquier medio, debe producir a propósito errores del sistema para ser corregidos luego por los desarrolladores del sistema [57].
- **Pruebas de rendimiento:** Mediante las pruebas de rendimiento es posible hallar tendencias y comportamientos para los elementos de una aplicación, los cuales generan bajo rendimiento. Este tipo de pruebas permiten identificar cuellos de botella, capacidad de concurrencia de usuarios, tiempos de respuesta de operaciones de negocio a nivel de sistema, establecer un marco de referencia para pruebas futuras, determinar el cumplimiento de los objetivos de rendimiento y requerimientos no funcionales, entre otros. Dentro de estas pruebas se realizan las pruebas de carga y las pruebas de estrés [58].
 - **Pruebas de estrés:** Mediante las pruebas de estrés es posible identificar la capacidad de respuesta de un sistema bajo condiciones de carga extrema, representadas por una alta concurrencia de usuarios y/o procesos. Una vez realizadas las pruebas de estrés se podrá conocer el punto de quiebre del aplicativo en términos de capacidad de respuesta, con lo cual será posible establecer acciones de optimización en diferentes niveles para asegurar una mejor capacidad de concurrencia de usuarios y/o procesos [59].
 - **Pruebas de carga:** Mediante la ejecución de las pruebas de carga es posible identificar la capacidad de recuperación de un sistema cuando es sometido a cargas variables tanto de usuarios como de procesos. Al realizar las pruebas de carga se puede determinar el tiempo de respuesta de todas las transacciones críticas del sistema y encontrar cuellos de botella en la aplicación [60].

Para llevar a cabo las pruebas de funcionalidad se utilizó el método de caja negra. Utilizando la técnica de particiones equivalentes:

- **Prueba de caja negra:** Se refiere a las pruebas que se llevan a cabo sobre la interfaz del sistema donde se diseñan casos de prueba de aceptación demostrando que las funciones del sistema son

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

operativas. Se centran principalmente en los requisitos funcionales del sistema. Para desarrollar la prueba se utilizó la técnica de partición equivalente [61].

- **Técnica de partición equivalente:** divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos de prueba antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar [61].

Diseño de Casos de Prueba de Aceptación (DCP):

Un caso de prueba es un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular [61]. A continuación, se les muestra los diseños de caso de pruebas del Caso de uso Buscar plan operativo mensual, los restantes se pueden encontrar en los anexos del documento.

Condiciones de ejecución: el usuario debe estar autenticado en el sistema con el rol “Energía”. El usuario debe haber seleccionado el módulo de Energía. Se debe haber seleccionado la opción: “Buscar” en el menú vertical superior y se selecciona la opción: “Plan Operativo Mensual”:

Escenario	Descripción	Fecha	Respuesta del sistema	Flujo central
EC 1.1 Buscar plan operativo mensual de forma correcta.	El sistema busca el plan operativo mensual de forma correcta.	V 4-2014	El sistema muestra un listado de planes operativos mensuales y permite realizar las siguientes acciones: 1.- Ver el plan operativo mensual al seleccionar la opción: "Ver" , ubicado en el extremo derecho del resultado de la búsqueda. 2.- Modificar el plan operativo mensual al seleccionar la opción: "Modificar" , ubicada en el extremo derecho del resultado de la búsqueda.	El usuario se autentica en el sistema con privilegios de administración. 1.-El sistema le muestra la interfaz correspondiente al módulo de Energía. 2.-El usuario selecciona la opción: "Buscar" , ubicada en el menú horizontal superior, y luego selecciona la opción de: "Plan

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

EC 1.2 Buscar plan operativo mensual de forma correcta sin resultados.	El sistema busca el plan operativo mensual y no encuentra información que cumpla con los criterios de búsqueda.	V 3-2015	El sistema no muestra ningún resultado y muestra el mensaje de error: "No se encontró información que cumpla con los criterios de búsqueda."	Operativo Mensual". 3.-El sistema le muestra al usuario un formulario para realizar la búsqueda. 4.-El usuario introduce la información y presiona el botón: "Buscar" .
--	---	-------------	---	---

Tabla 6: Caso de prueba: Buscar plan operativo mensual.

Resultado de las pruebas

Las pruebas efectuadas se realizaron en 2 iteraciones y una revisión final. Dichas pruebas se llevaron a cabo con la ayuda de los diseños de casos de pruebas por cada funcionalidad identificada. En la primera iteración de pruebas de 70 requisitos funcionales implementados, se identificaron 20 no conformidades de las cuales se solucionaron solo 18, en la segunda iteración de 70 requisitos identificados se identificaron 36 no conformidades corregidas en su totalidad y en la revisión final se encontraron 0 no conformidades.

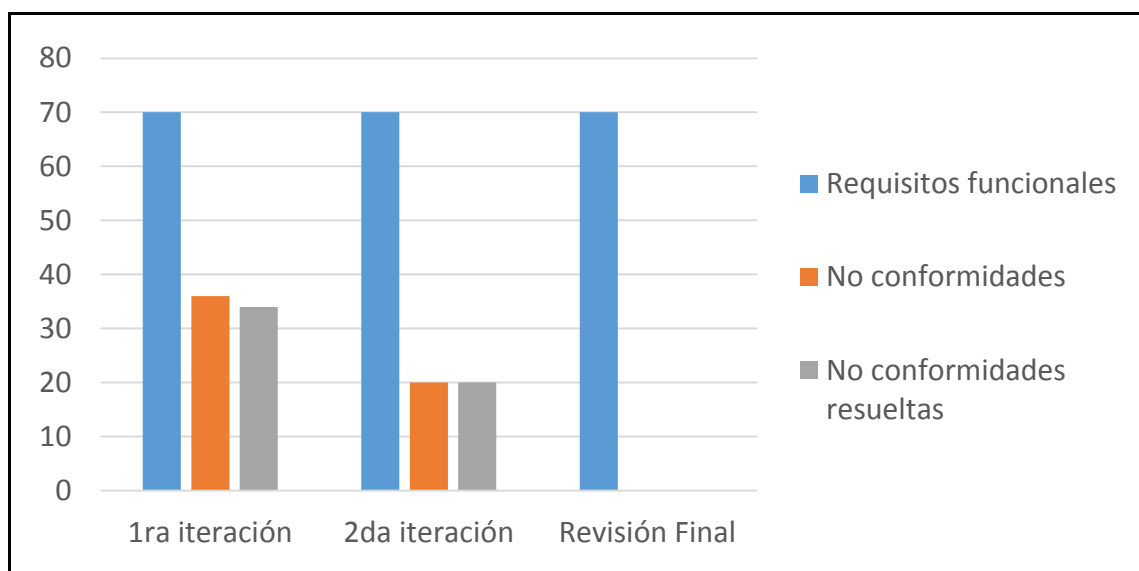


Figura 29: Gráfico de las iteraciones de las pruebas realizadas.

Para llevar a cabo las pruebas de unitarias se utilizó el método de caja blanca. Utilizando la técnica del camino básico:

- **Prueba de caja blanca:** Se comprueban los caminos lógicos del sistema proponiendo casos de prueba que ejerciten conjuntos específicos de condiciones y/o bucles. Se puede examinar el

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

estado del programa en varios puntos para determinar si el estado real coincide con el esperado o mencionado. Para desarrollar la prueba se utilizó la técnica del camino básico [61].

- **Técnica de camino básico:** Esta prueba le permite al diseñador de casos de prueba obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución. Los casos de prueba obtenidos del conjunto básico garantizan que durante la prueba se ejecute por lo menos una vez cada sentencia del programa [61].

El cálculo de la complejidad ciclomática se realiza de tres formas diferentes:

- El número de regiones del grafo de flujo coincide con la complejidad ciclomática.
- La complejidad ciclomática, $V(G)$, de un grafo de flujo G se define como: $V(G) = A - N + 2$, donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.
- La complejidad ciclomática, $V(G)$, de un grafo de flujo G también se define como: $V(G) = P + 1$, donde P es el número de nodos predicado (son los nodos de los cuales parten dos o más aristas) que tiene contenido el grafo de flujo G .

Ejemplo de código de la funcionalidad deleteAction:

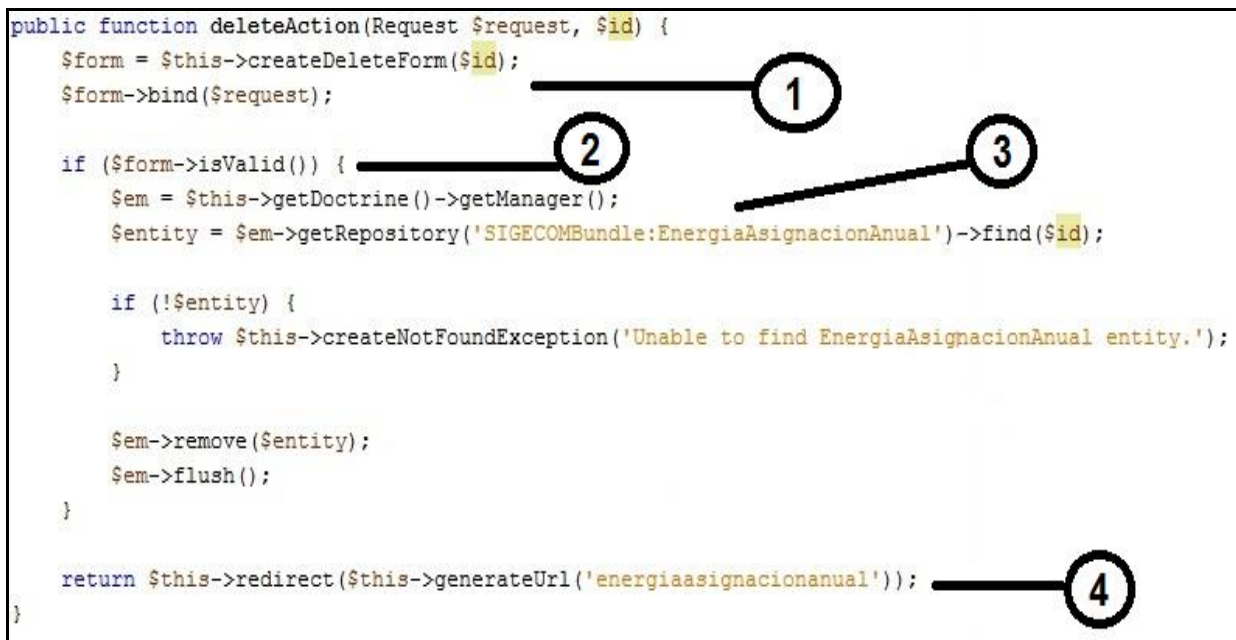


Figura 30: Código de la funcionalidad deleteAction.

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

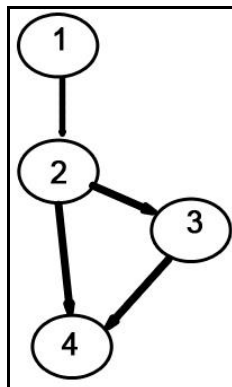


Figura 31: Grafo del flujo asociado a la funcionalidad deleteAction.

Resultado del cálculo de la complejidad ciclomática:

- El grafo de flujo tiene 3 regiones.
- $V(G) = A - N + 2 = 4 - 4 + 2 = 2$
- $V(G) = P + 1 = 1 + 1 = 2$

A partir del resultado obtenido, se determina que la funcionalidad presenta una complejidad ciclomática de 2, lo que deriva que presenta a lo sumo 2 caminos lógicos por donde ejecutarse dicha funcionalidad. En la siguiente tabla se presentan los caminos básicos:

Nro.	Camino básico
1	1, 2, 4
2	1, 2, 3, 4

Tabla 7: Caminos lógicos.

Se procede a realizar los casos de prueba para la funcionalidad en cuestión. Se debe elaborar al menos un caso de prueba por cada camino básico.

Casos de pruebas

Camino	1, 2, 4
Descripción	Eliminar asignación anual de combustible.
Condición de ejecución	Deben existir asignaciones anuales registradas en el sistema.
Entrada	Seleccionar la asignación anual que se eliminará.
Resultados esperados	La asignación anual ha sido eliminada.

Tabla 8: Caso de prueba camino #1.

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

Camino	1, 2, 3, 4
Descripción	Eliminar asignación anual de combustible.
Condición de ejecución	Deben existir asignaciones anuales registradas en el sistema.
Entrada	Seleccionar la asignación anual que se eliminará.
Resultados esperados	La asignación no ha sido eliminada.

Tabla 9: Caso de prueba camino #2.

Los casos de pruebas permitieron garantizar que:

- Se ejecuten por lo menos una vez todos los caminos.
- Todas las decisiones lógicas en sus vertientes verdadera y falsa.
- Las estructuras internas de datos para asegurar su validez.
- Todos los bucles en sus límites operacionales.

Para llevar a cabo las pruebas de carga y estrés se utilizó la herramienta Apache JMeter. Este tipo de prueba se utiliza para simular una carga pesada en un servidor, en la red o a un objeto, para poner a prueba su resistencia o para analizar el rendimiento en diferentes tipos de carga. Para realizar estas pruebas, se simula el comportamiento de usuarios conectados al sistema realizando varias transacciones.

Las pruebas se llevarán a cabo haciendo uso de un ordenador con:

- Microprocesador Intel DualCore a 2.6GHz.
- Memoria RAM 3GB.

Se considerarán:

- 20 hilos (simulación de 10 usuarios) y un período de subida de 1 segundo.

Se añadirá el elemento "Summary Report" que brindará la siguiente información:

- **Etiqueta:** El nombre de la muestra (conjunto de muestras). # Muestras: El número de muestras para cada URL.
- **Muestra:** El número de peticiones al URL.
- **Media:** El tiempo medio transcurrido para un conjunto de resultados.
- **Mín:** El mínimo tiempo transcurrido para las muestras de la URL dada.
- **Máx:** El máximo tiempo transcurrido para las muestras de la URL dada.
- **Std. Prog:** La desviación estándar de la muestra el tiempo transcurrido
- **% Error:** Porcentaje de las peticiones con errores.
- **Rendimiento:** Rendimiento medido en base a peticiones por segundo/minuto/hora.
- **Kb/sec:** Rendimiento medido en Kilobytes por segundo.

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

- **Media de Bytes:** Tamaño medio de la respuesta de la muestra medido en bytes.

Tras haber introducido las variaciones indicadas, El elemento “Summary Report” quedará como sigue:

Simulación de 20 usuarios con un período de subida de 1 segundo

Summary Report									
Nombre: Summary Report									
Comentarios									
Escribir todos los datos a Archivo									
Nombre de archivo				Navegar...	<input type="checkbox"/> Escribir en Log Sólo Errores			Configurar	
Label	# Muestras	Media	Mín	Máx	Std. Dev.	% Error	Rendimiento	Kb/sec	Avg. Bytes
/sigecom/web/...	60	13924	581	60581	20804,93	0,00%	30,2/min	10,44	21230,0
/sigecom/web/...	60	4692	1065	8917	2253,28	0,00%	28,9/min	10,43	22144,0
/sigecom/web/...	120	2504	569	5333	1074,21	0,00%	38,4/min	13,27	21230,0
/sigecom/web/...	60	4336	1047	8496	2153,89	0,00%	19,2/min	6,92	22144,0
/sigecom/web/...	40	4525	1269	7834	1866,36	0,00%	23,6/min	8,50	22144,0
/sigecom/web/...	20	4638	1483	8061	1856,42	0,00%	12,7/min	4,59	22144,0
/sigecom/web/...	20	4446	1384	6372	1695,12	0,00%	11,7/min	4,21	22144,0
/sigecom/web/...	20	4756	1652	7363	1769,60	0,00%	9,9/min	3,57	22144,0
/sigecom/web/...	20	4407	1759	6463	1548,21	0,00%	9,5/min	3,43	22144,0
/sigecom/web/...	20	4649	1533	7224	1760,19	0,00%	9,1/min	3,29	22144,0
/sigecom/web/...	40	4811	1350	10039	2081,39	0,00%	15,3/min	5,53	22144,0
/sigecom/web/...	20	4997	1691	8519	2002,12	0,00%	8,5/min	3,05	22144,0
/sigecom/web/...	20	4939	2015	9421	2030,90	0,00%	7,8/min	2,83	22144,0
/sigecom/web/...	20	4682	1943	7936	1699,67	0,00%	7,9/min	2,84	22144,0
/sigecom/web/...	20	4140	1734	9141	1981,51	0,00%	8,0/min	2,87	22144,0
/sigecom/web/...	20	3755	1369	7982	1684,55	0,00%	8,0/min	2,89	22144,0
/sigecom/web/...	20	3455	911	7817	2145,06	0,00%	8,6/min	3,09	22144,0
/sigecom/web/...	20	3240	775	7154	2036,15	0,00%	9,0/min	3,25	22144,0
TOTAL	640	4833	0	60581	7295,35	0,00%	3,3/sec	68,12	21237,4

Figura 32: Resultados de las pruebas de carga y estrés para 20 usuarios.

Se observa que para esta cantidad de usuarios no se detecta ningún error, lo cual se deduce en la columna que representa el tanto por ciento de errores para cada petición. En cuanto al rendimiento se puede observar que para una simulación de 20 usuarios con un período de subida de 1 segundo, el servidor es capaz de aceptar una media de 3,3 peticiones por segundo. Luego de las pruebas realizadas, se observa que la aplicación no detectó errores en ninguna ocasión y el rendimiento se encuentra en correspondencia con el número de usuarios simulados. El volumen de las pruebas que se realicen, debe ajustarse a las características del ordenador donde se ejecuten ya que los resultados obtenidos podrían estar condicionados por dichas características.

Para garantizar la seguridad del sistema se realizan pruebas de seguridad mediante una lista de chequeo aplicable a aplicaciones web. El objetivo general de esta lista es evaluar a través de indicadores la seguridad de las aplicaciones en un primer nivel (nivel 1), establecido por los especialistas del grupo de seguridad del Departamento de Pruebas de Software (DEPSW) de la UCI. Los especialistas del grupo de Seguridad del DEPSW de la UCI han establecido 3 niveles para realizar las pruebas de Seguridad. Para evaluar la seguridad de las aplicaciones en un primer nivel (nivel 1) se definió una lista de chequeo que cuenta con 15 indicadores separados en 4 tipos de pruebas accesibles a los 3 niveles.

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

El resultado de la evaluación de dicha lista arrojó, para los 4 tipos de pruebas, los siguientes resultados:

- 1. Pruebas de Autorización:** Ningún usuario estándar puede modificar sus privilegios ni los de otro usuario en la aplicación.
- 2. Pruebas de Gestión de Sesiones:** No se puede acceder a la aplicación copiando la URL después de estar autenticado, cerrar el navegador y volver a abrirlo. Así mismo no se puede acceder al cerrar la sesión de un usuario y dar clic en el botón “Atrás” del navegador.
- 3. Validación de Datos:** El sistema valida a través del servicio SoapClient, que los usuarios existan en el dominio UCI y verifica que la contraseña introducida sea correcta. El sistema no muestra mensajes indebidos al colocar en la barra de dirección o en campos de entrada los caracteres: comillas simples (‘’), signos de *ampersand* (&), signos: + - /.
- 4. Comprobación del Sistema de Autenticación:** Los mensajes de error para distintas combinaciones de autenticación muestran la misma información. Los tiempos de respuestas usuario correcto - contraseña incorrecta y usuario - contraseña incorrecta son los mismos. El sistema protege el envío de los datos mediante protocolo seguro (*HTTPS*).

Esta lista de chequeo permitió recoger los puntos eficientes y los ineficientes que tienen los elementos chequeados, así como verificar que el grado de seguridad de la aplicación es adecuado para la protección de la información.

Para evaluar la seguridad del sistema se hizo uso de la herramienta Acunetix Web Vulnerability Scanner 8.0.

Acunetix Web Vulnerability Scanner: es una herramienta que será capaz de escanear sitios web en busca de posibles fallos de seguridad que puedan poner en peligro la integridad de la página publicada en Internet. Esta aplicación ejecuta una serie de pruebas, totalmente configurables por el usuario, para identificar las vulnerabilidades tanto en la programación de la página como en la configuración del servidor [63]. A continuación, se muestran las vulnerabilidades identificadas con el uso de la herramienta:



Figura 33: Resultado de las pruebas de seguridad generadas por la herramienta Acunetix.

CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBAS DEL SISTEMA DE GESTIÓN DE LA INFORMACIÓN DEL COMBUSTIBLE

Durante el escaneo realizado por la herramienta fueron detectadas solo 9 no conformidades, de ellas 5 de prioridad baja, 2 de prioridad media y 2 informativas. Entre las no conformidades de prioridad baja se encontraron: ficheros del servidor Apache vulnerables a ataques de fuerza bruta y *cookies* de sesión sin el indicador de seguridad. Las no conformidades de prioridad media están relacionadas con el auto completamiento de los formularios, se corrigió esta no conformidad. El resto deja al descubierto información acerca de una página de error que contiene la versión del servidor web y una lista de los módulos habilitados en este servidor.

4.6.- Conclusiones

Como parte del desarrollo del capítulo se determinan las siguientes conclusiones parciales:

- El diagrama de componentes del sistema realizado, permitió una apreciación visual de la organización y las dependencias lógicas entre los componentes del sistema.
- El establecimiento de un estándar de codificación le permitió a los desarrolladores integrarse como un solo equipo, al trabajar sobre la misma base de codificación pudiendo mejorar el código o añadir nuevas funcionalidades por parte de otros desarrolladores.
- Las pruebas realizadas permitió una correcta detección de las deficiencias en el funcionamiento del sistema para propiciar un mejor funcionamiento del sistema.

Conclusiones

El desarrollo de esta investigación da solución al problema científico que le dio inicio, a través del cumplimiento de sus objetivos específicos y tareas de investigación, concluyendo con:

- El estudio de los sistemas homólogos de gestión de la información del combustible a nivel nacional e internacional permitió definir una propuesta de solución acorde a las necesidades existentes.
- El enfoque ágil propuesto por la metodología de desarrollo de software OpenUp y el uso de las tecnologías y herramientas seleccionadas permitió un correcto análisis y descripción de los procesos que se debían ejecutar.
- La implementación del Sistema de gestión de la información del combustible, luego de ser consecuentemente validada y verificada a partir de las pruebas definidas, permitió lograr la inmediatez y retroalimentación en el proceso de gestión de la información del combustible.

Recomendaciones

Los objetivos del trabajo de diploma han sido alcanzados pero se recomienda:

- Establecer como norma de trabajo el presente sistema en la Dirección de Transporte, el Grupo de Gestión Energética y la Vicerrectoría Económica.
- Continuar con la investigación del sistema con la finalidad de aportar nuevas funcionalidades.
- Implementar en una versión posterior del sistema los procesos relacionados con el control de las tarjetas magnéticas de combustible que maneja la Dirección de Economía.
- Implantar el sistema desarrollado en la Dirección Energética del MES y demás universidades del país.

REFERENCIAS BIBLIOGRÁFICAS

Referencias Bibliográficas

1. **Pérez, O.,González, S., Martínez, Y.** La gestión energética en el contexto empresarial cubano. *Revista Caribeña de Ciencias Sociales*, 2013. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://caribeña.eumed.net/gestion-energetica-contexto-empresarial-cubano/> .
2. **Sánchez, A., García, Wilson, J., Wilson, A.** *Gestión de Portadores Energéticos en la UCI.* s.l. : Serie Científica de la Universidad de las Ciencias Informáticas, 2011, Vol. 4. ISSN/ RNPS. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://publicaciones.uci.cu/index.php/SC/article/view/491>.
3. Recursos Energéticos. *DEMO E-DUCATIVA CATEDU*, 2010. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: http://e-educativa.catedu.es/44700165/aula/archivos/repositorio/3500/3669/html/3_recursos_energticos.html.
4. Definición de Combustible. *Definición abc*. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://www.definicionabc.com/general/combustible.php>.
5. CltET. *Centro de Inovación para la logistica y el Transporte por Carrera*, 2009. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: http://www.enproyecto.com/fsd/download.aspx?app=CITET_Docs&path=571.ppt&hash=EC7756EC1209E2EDE07E9DE8D77D122D.
6. Guia para la gestión de combustible en la flotas de transporte por carretera. *IDEA*, 2009. [En línea]. [Citado: Enero 23,2014]. Disponible en: http://www.idae.es/uploads/documentos/documentos_10232_Guia_gestion_combustible_flotas_carretera_06_32bad0b7.pdf.
7. **Prado, F.** Gestión de Flotas. *Revista Cesvimap*, 2010. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://www.revistacesvimap.com/revista77/pdfs/Ingenieria.pdf>.
8. **EUGCOM.** *Descripción del Producto ADMINISTRACIÓN Y CONTROL DE FLOTAS*. Chile: s.n., 2009. [Citado el: 21 de Octubre de 2012.]
9. KreaSoft. KreaSoft Software y Tecnología. 2010. [En línea]. [Citado el: Octubre 21, 2013.] Disponible en: <http://www.kreasoft.cl/kreaflota.html>.
10. Software Incremental. FlotaWeb. 2010. [En línea]. [Citado: Octubre 21, 2013.]. Disponible en: <http://flotaweb.com/>.
11. Saavedra, A., Rodríguez, R. 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, 2013.

REFERENCIAS BIBLIOGRÁFICAS

- [En línea]. [Citado: Octubre 21, 2013.]. Disponible en: http://bibliodoc.uci.cu/RDigitales/2013/noviembre/26/TD_06861_13.pdf.
12. Saavedra, A., Rodríguez, R. 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, 2013. [En línea]. [Citado: Octubre 21, 2013.]. Disponible en: http://bibliodoc.uci.cu/RDigitales/2013/noviembre/26/TD_06861_13.pdf.
13. Nuestros Productos. TranSoft. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://www.transoft.transnet.cu/?controlador=Producto&accion=index>.
14. Graña, R. Universitat Oberta de Catalunya. Metodologías de desarrollo de proyectos informáticos en entornos web, 2011. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://hdl.handle.net/10609/7924>.
15. Orjuela, A., Rojas, M. *The methodologies of Agil Development like and Opportunity for the Inginneering of Educative Software*. 2008.
16. Ingeniería de software. PROGRAMACION EXTREMA XP. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html.
17. Torres, C. *Establecimiento de una Metodología de Desarrollo de Software para la Universidad de Navojoa Usando OpenUP*. 2008.
18. Martínez, R. Java4310: Getting Started with JSP, 2013. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://cnx.org/content/m48055/1.3>.
19. Doyle, M. Beginning PHP 5.3, 2012. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://dkum.uni-mb.si/lzpisGradiva.php?id=32352>.
20. Ferreira, J. Criando Diagramas UML com o StarUML, 2013. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://cnx.org/content/m47736/1.1>.
21. González, E. Proyecto complejo deportivo, 2013. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://hdl.handle.net/10609/22362>.
22. Booststrap, 2013. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://sistemasweb2013.webcindario.com/bootstrap/caracteristicas.php>.
23. Booststrap, 2013. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://www.genbetadev.com/frameworks/bootstrap>.
- 24.- Potencier, F, Zaninotto, F. Symfony en pocas palabras. 2010. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: http://librosweb.es/symfony_1_2/capitulo_1/symfony_en_pocas_palabras.html.

REFERENCIAS BIBLIOGRÁFICAS

25. **Delgado, C.**, Poyo, A. COMPARATIVA FRAMEWORKS. Desarrollo de Aplicaciones Web II. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://es.scribd.com/doc/189203494/Comparativa-Framework>.
26. **Naramore, Elizabeth. 2005.** *Beginning PHP5, Apache, and MySQL Web Development*. Canadá: Wiley Publishing. 2005. ISBN: 0-7645-7966-5.
27. **Arredondo, K.** Servicio Web 2.0. Apache vs IIS, 2012. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://servicioweb20.blogspot.com/2012/03/apache-vs-iis.html>.
28. **Gavilán, F.** Internet Information Services (IIS), 2011. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://terrafx9.blogspot.com/2011/11/internet-information-services-iis.html>.
29. **Howard, Y.** Systems Design: Use Case Modelling Lab using Visual Paradigm, 2012. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://www.edshare.soton.ac.uk/9651/>.
30. **Marciszack, M.** Modelos de especificación de requerimientos para la obtención de esquemas conceptuales en un dominio restringido: comparación de metodologías, 2011. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://hdl.handle.net/10915/4204>.
31. **Jácome, S., Tituaña, W., Torres, E.** Elaboración de un manual de la plataforma Netbeans Ide para la Disicom, 2009. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://repositorio.espe.edu.ec/handle/21000/4437>.
32. **Novella, J.** Sistema de gestión de base de datos PostgreSQL, 2012. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://hdl.handle.net/10251/17490>.
33. **Gómez, J.** Motores MySQL, 2014. [En línea]. [Citado: Enero 23, 2014.]. Disponible en: <http://hdl.handle.net/10045/36375>.
34. **Bizagi. 2012.** Bizagi. Diciembre de 2012. [En línea]. [Citado: Noviembre 19, 2013.]. Disponible en: <http://www.buenastareas.com/ensayos/Bizagi/4235294.html>.
35. **Colectivo de autores.** SGuía, 2009. [En línea]. [Citado: Enero 24, 2014.]. Disponible en: <http://sg.com.mx/guia/node/1046>.
36. **Pressman, Roger S.** 2005. *Ingeniería del Software Un enfoque práctico*. . s.l.: Sexta edición, 2005.
37. **Gracia, L.** Un poco de Java. Técnicas para la captura de Requisitos, 2013. [En línea]. [Citado: Noviembre 19, 2013.]. Disponible en: <http://unpocodejava.wordpress.com/2013/05/09/tecnicas-para-la-captura-de-requisitos/>.

REFERENCIAS BIBLIOGRÁFICAS

38. Fundamentos de Ingeniería. Inspección, Validación, Completitud, Detección de Conflictos e Inconsistencias de Requerimientos, 2012. *[En línea]*. [Citado: Enero 24, 2014.]. Disponible en: <http://fundamentos-ing-requisitos.blogspot.com/2012/03/inspeccion-validacion-completitud.html>.
39. VALIDACION DE REQUISITOS, 2014. *[En línea]*. [Citado: Enero 24, 2014.]. Disponible en: http://is.ls.fi.upm.es/docencia/masterTI/ARS/docs/Manual_M2C1U11.pdf.
40. **Rivas, J.** Análisis y diseño de sistemas. Casos de Uso, 2010. [Citado: Enero 24, 2013.]. Disponible en: http://pedagogica-analisis.blogspot.com/2010/03/casos-de-uso_17.html.
41. Diseño Orientado a Objetos. OCW Usal - Universidad de Salamanca. *[En línea]*. [Citado: Enero 24, 2014.]. Disponible en: <http://ocw.usal.es/enseñanzas-tecnicas/ingenieria-del-software/contenidos/Tema6-DOO-1pp.pdf>.
42. **Larmam, C.** *UML y Patrones Capítulo 9 MODELO DE CASOS DE USO: REPRESENTACIÓN DE LOS DIAGRAMAS DE SECUENCIA DEL SISTEMA*. s.l. : Felix Varela, 2008.
43. **Cintra, Y.** *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.
44. MICROSOFT CORPORATION. Excel. Editado por: González, C. S. Impresos y Revistas S.A: McGraw McGraw-Hill / Interamericana de España, S.A.U, 2001. 622 p. Diccionario de Informático e Internet de Microsoft. ISBN 84-481-2893-1.
45. **SIYAN, K. S. y PARKER, T.** *Tcp/ip*. Indianapolis: Sams Publishing, 2002. 1-2 p. Tcp/Ipunleashed. ISBN 0-672-32351.
46. DICCIONARIO INFORMÁTICO, C. *HTTP Diccionario de términos informáticos Clarin*. *[En línea]*. [Consultado el: 10 de Febrero de 2012]. Disponible en: <http://diccionario.babylon.com/http/>.
47. **IVAR, J.; GRADY, B.** *El Proceso Unificado del Desarrollo del Software*. Imprenta fareso: Addison Wesley Logan inc, 2000a. 217 p. ISBN 0-201-57169-2.
48. **Potencier, F., Zaninotto, F.** *Symfony 2.1, la guía definitiva*, 2010. *[En línea]*. [Citado: Enero 24, 2014.]. Disponible en: http://librosweb.es/symfony_1_2/.
49. **Allen, R.** "A formal approach to software architectures". s.l.: Proceedings of the IFIP Congress '92, 1992.
50. **Polo, M.** Introducción a los patrones, 2011. *[En línea]*. [Citado: Enero 24, 2014.]. Disponible en: <http://www.inf-cr.uclm.es/www/mpolo/asig/0304/0102/patronesgrasp.ppt>.

REFERENCIAS BIBLIOGRÁFICAS

51. **Pressman, R.** *Software engineering practitioner's approach*. New York: The McGraw-Hill Companies, 2010. ISBN 978-0-07-337597-7.
52. **Garcia, S.** Godit mundo binario. Patrones de diseño GoF, 2012. [En línea]. [Citado: Enero 24, 2014.]. Disponible en: <http://www.godtic.com/blog/2012/11/15/patrones-de-diseno-gof/>.
53. **Pérez, Y., Rosabal, A.** Modelo de implementación. [Citado: Enero 24, 2014.]. Sistema para la Gestión y Análisis de Información Estadística en la salud pública cubana: Subsistema Editor de Plantillas. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas, Junio 2009.
54. **Valdés, J.** Diagrama de Componentes. [Citado: Enero 24, 2014.]. Desarrollo del Módulo de Gestión de Recursos Humanos del Sistema Integral de Gestión Administrativa de la Facultad 7. Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas, Julio 2010.
55. **ZENTENO, A. H. T.** Método de pruebas de sistema basado en modelos navegacionales en un contexto MDWE, 2011. [En línea]. Disponible en: <http://www.lsi.us.es/docs/doctorado/memorias/MemolInvestigArturoHTorres.pdf>.
56. **ULIBARRI.** Capítulo 5: Pruebas al sistema, 2004. [En línea]. .Disponible en: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/moreno_a_jl/capitulo5.pdf.
57. **V&V QUALITY.** PRUEBAS DE SEGURIDAD, 2011. [En línea]. [Citado: Enero 24, 2014.]. Disponible en: http://www.vyvquality.com/w1/index.php?option=com_content&view=article&id=79&Itemid=151.
58. **V&V QUALITY.** PRUEBAS DE RENDIMIENTO, 2011. [En línea]. [Citado: Enero 24, 2014.]. Disponible en: http://www.vyvquality.com/w1/index.php?option=com_content&view=article&id=78&Itemid=150.
59. **V&V QUALITY.** Pruebas de estrés, 2011. [En línea]. [Citado: Enero 24, 2014.]. Disponible en: http://www.vyvquality.com/w1/index.php?option=com_content&view=article&id=89&Itemid=160.
60. **V&V QUALITY.** PRUEBAS DE CARGA, 2011. [En línea]. [Citado: Enero 24, 2014.]. Disponible en: http://www.vyvquality.com/w1/index.php?option=com_content&view=article&id=91&Itemid=162.
61. **Pressman, R.** *Software engineering practitioner's approach*. New York: The McGraw-Hill Companies, 2002.
62. **WEBSEGURIFY.** *Innovative Web Application Security Tools*, 2011. [En línea]. [Citado: Enero 24, 2014.].Disponible en: <http://www.websecurify.com/>.

REFERENCIAS BIBLIOGRÁFICAS

63. Marco de desarrollo de la Junta de Andalucía. Acunetix Web Vulnerability Scanner, 2014. *[En línea]*. [Citado: Enero 24, 2014.]. Disponible en: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/656>.
64. **Pérez, J.** Fundamentos de Ingeniería. Modelado del Negocio, 2012. *[En línea]*. [Citado: Febrero 24, 2014.]. Disponible en: <http://fundamentos-ing-requisitos.blogspot.com/2012/03/modelado-del-negocio.html>.
65. **Tamames, V.** Optimyth. Estándares de codificación y buenas prácticas, 2013. *[En línea]*. [Citado: Febrero 24, 2014.]. Disponible en: <http://blog.optimyth.com/es/2013/11/estandares-de-codificacion-y-buenas-practicas>.
66. **Natalia, A.** Técnicas de evaluación de software. 2005. *[En línea]*. [Citado: Febrero 24, 2014.]. Disponible en: www.grise.upm.es/sites/trs/1/pdf/Documentacion_Evaluacion_7.pdf.

Glosario de Términos

- **Petróleo:** Es una mezcla homogénea de compuestos orgánicos, principalmente hidrocarburos insolubles en agua. También es conocido como petróleo crudo o simplemente crudo.
- **Gasóleo:** También denominado gasoil, es un hidrocarburo líquido, compuesto fundamentalmente por parafinas y utilizado principalmente como combustible en calefacción y en motores diésel.
- **Querosene:** Es un líquido transparente (o con ligera coloración amarillenta) obtenido por destilación del petróleo.
- **Gasolina:** Es una mezcla de hidrocarburos alifáticos obtenida del petróleo por destilación fraccionada, que se utiliza como combustible en motores de combustión interna con encendido por chispa convencional o por compresión, así como en estufas, lámparas, limpieza con solventes y otras aplicaciones.
- **Nafta:** Un rango de destilados más ligeros que la querosina utilizada como carga para la producción de gasolina para motores y para la industria química.
- **Fósil:** Son los restos o señales de la actividad de organismos pretéritos. Dichos restos, conservados en las rocas sedimentarias, pueden haber sufrido transformaciones en su composición o deformaciones más o menos intensas. La ciencia que se ocupa del estudio de los fósiles es la Paleontología.
- **Carbón:** es una roca sedimentaria de color negro, muy rica en carbono y con cantidades variables de otros elementos, principalmente hidrógeno, azufre, oxígeno y nitrógeno, utilizada como combustible fósil.
- **Gas:** al estado de agregación de la materia en el cual, bajo ciertas condiciones de temperatura y presión, sus moléculas interrelacionan solo débilmente entre sí, sin formar enlaces moleculares, adoptando la forma y el volumen del recipiente que las contiene y tendiendo a separarse, esto es, expandirse, todo lo posible por su alta energía cinética.
- **Bundle:** Un bundle es un conjunto estructurado de archivos que se encuentran en un directorio y que implementan una sola característica. Es un concepto similar al de los plugins en otras aplicaciones. La diferencia clave es que en Symfony2 todo es un bundle, incluyendo tanto la funcionalidad básica de la plataforma como el código escrito de la aplicación.

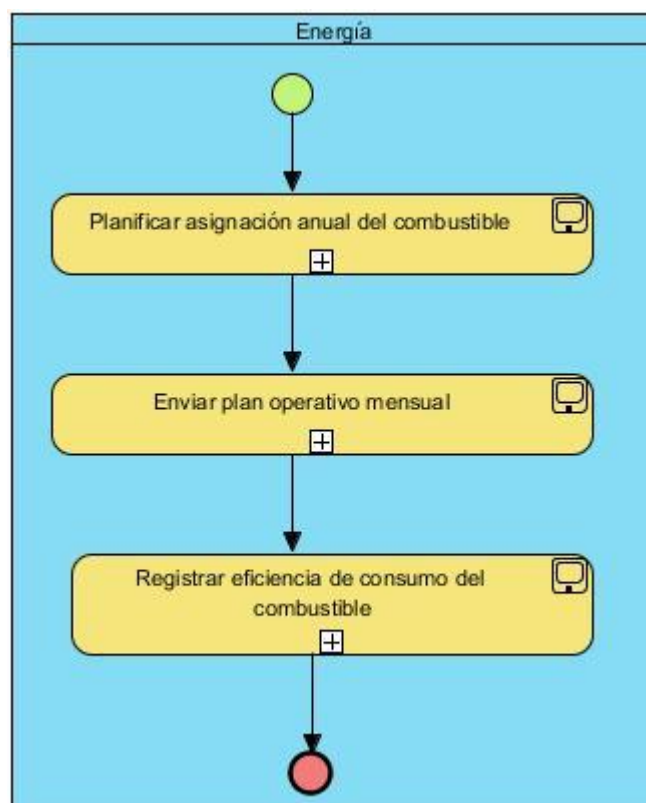


Figura 1: Diagrama de procesos general del Grupo de Gestión Energética.

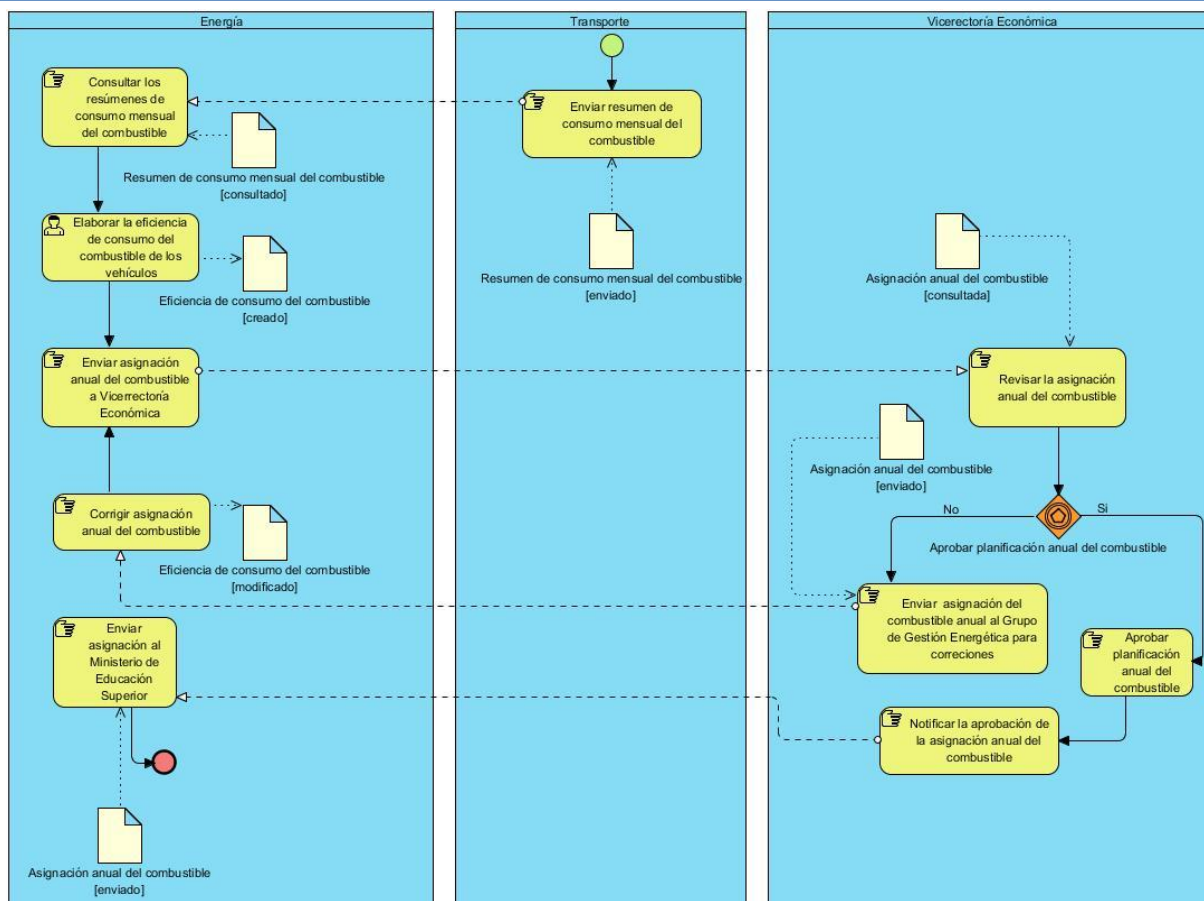


Figura 2: Diagrama del proceso asignación anual del combustible del Grupo de Gestión Energética.

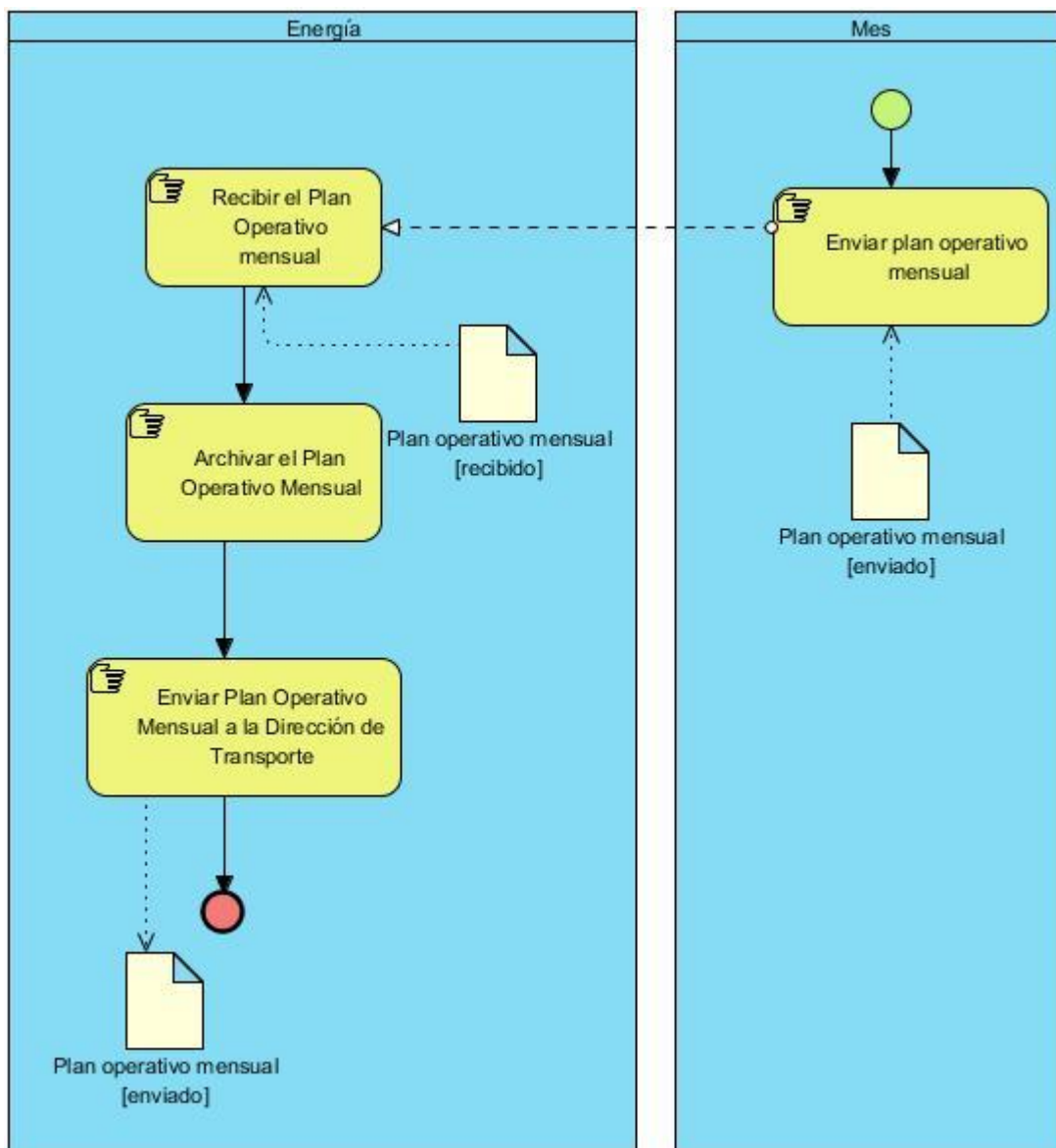


Figura 3: Diagrama del proceso enviar plan operativo del Grupo de Gestión Energética.

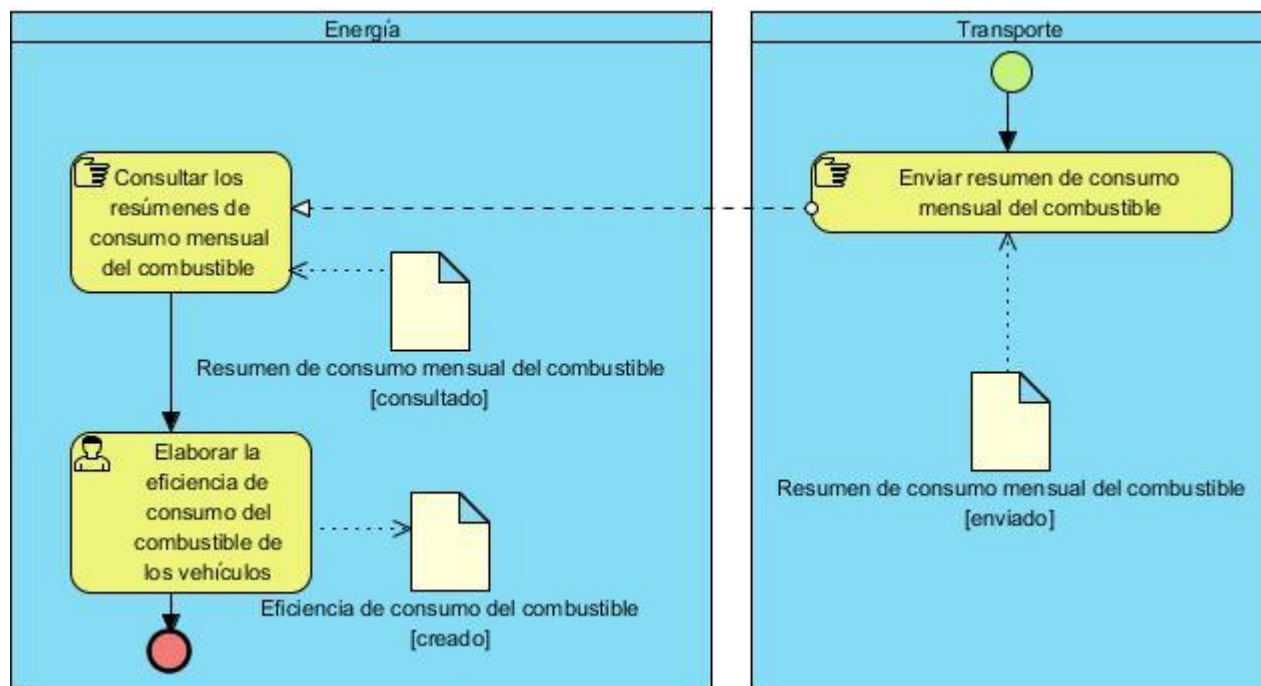


Figura 4: Diagrama del proceso registrar eficiencia de consumo del combustible del Grupo de Gestión Energética.

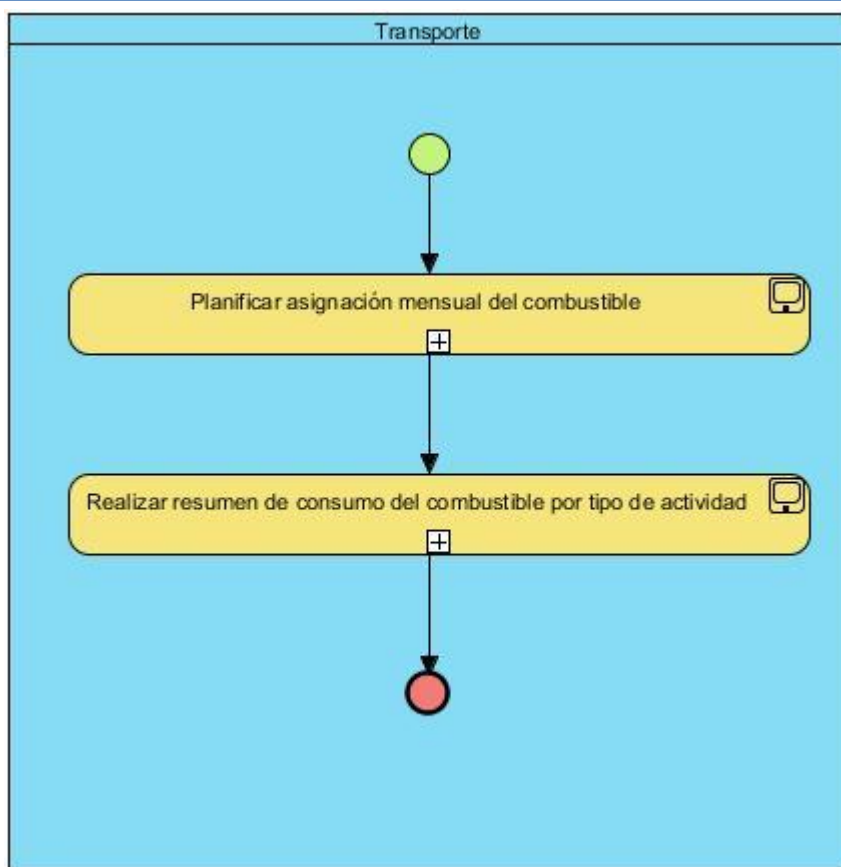


Figura 5: Diagrama de proceso general de la Dirección de Transporte.

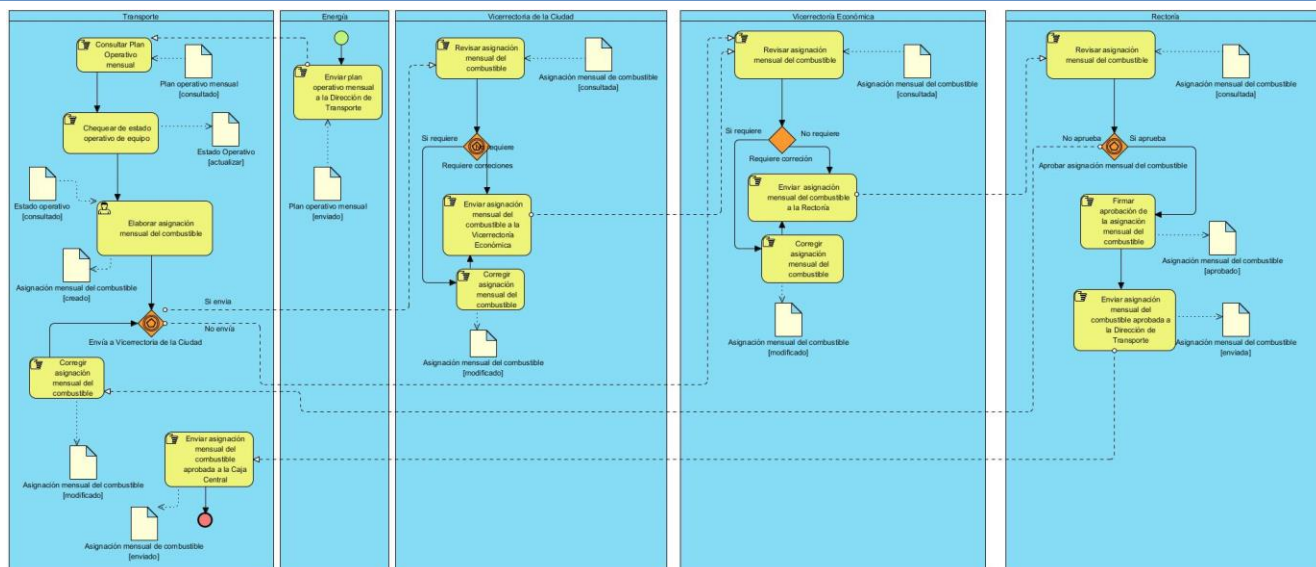


Figura 6: Diagrama del proceso planificar asignación mensual del combustible de la Dirección de Transporte.

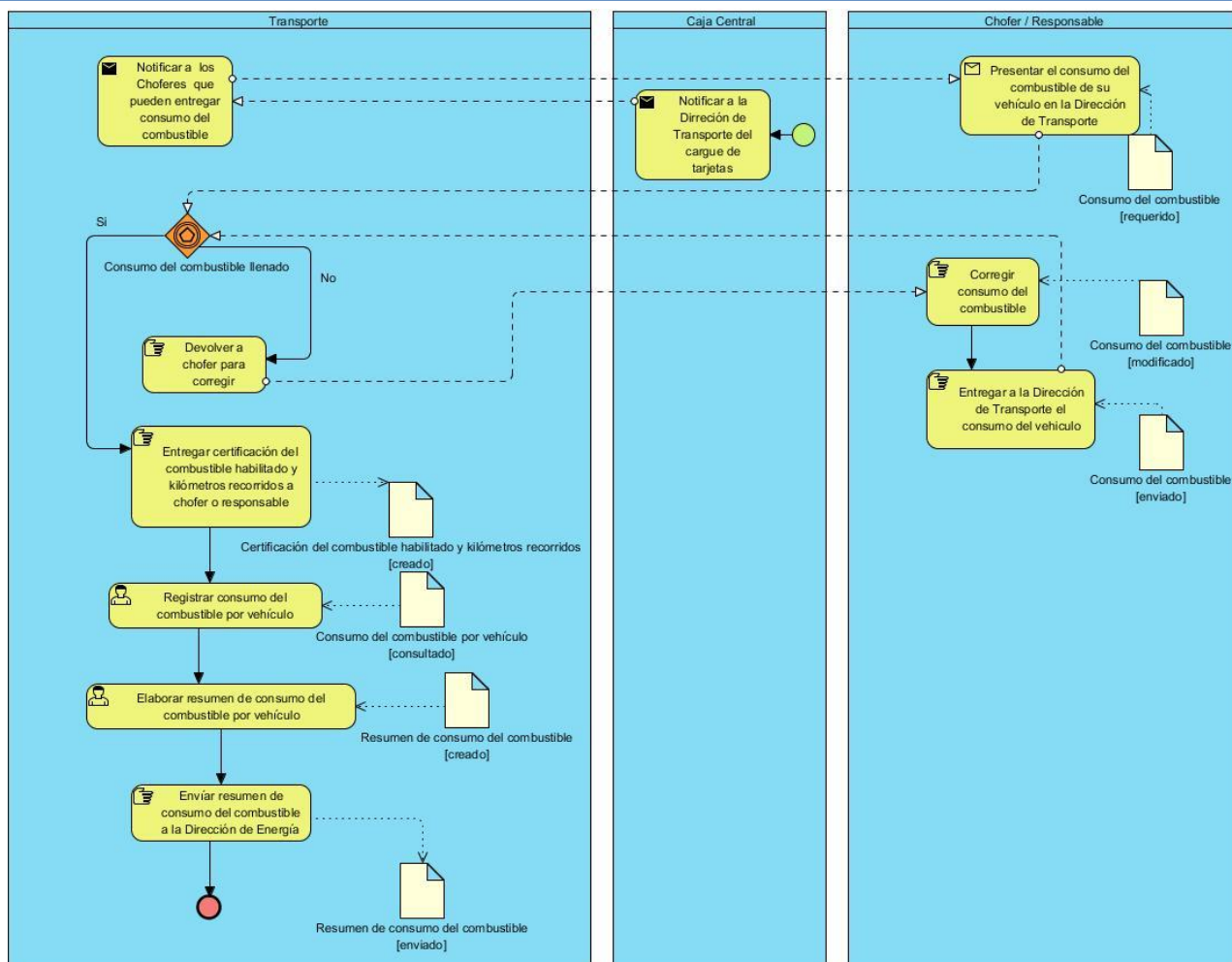


Figura 7: Diagrama del proceso realizar resumen de consumo del combustible por tipo de actividad de la Dirección de Transporte.

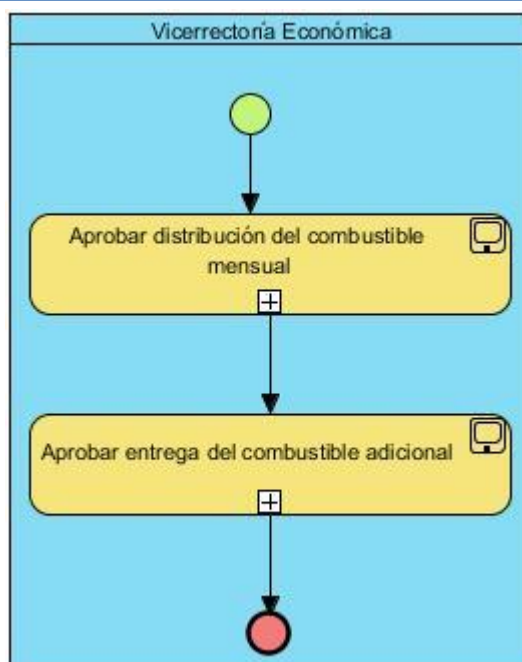


Figura 8: Diagrama de proceso general de la Vicerrectoría Económica.

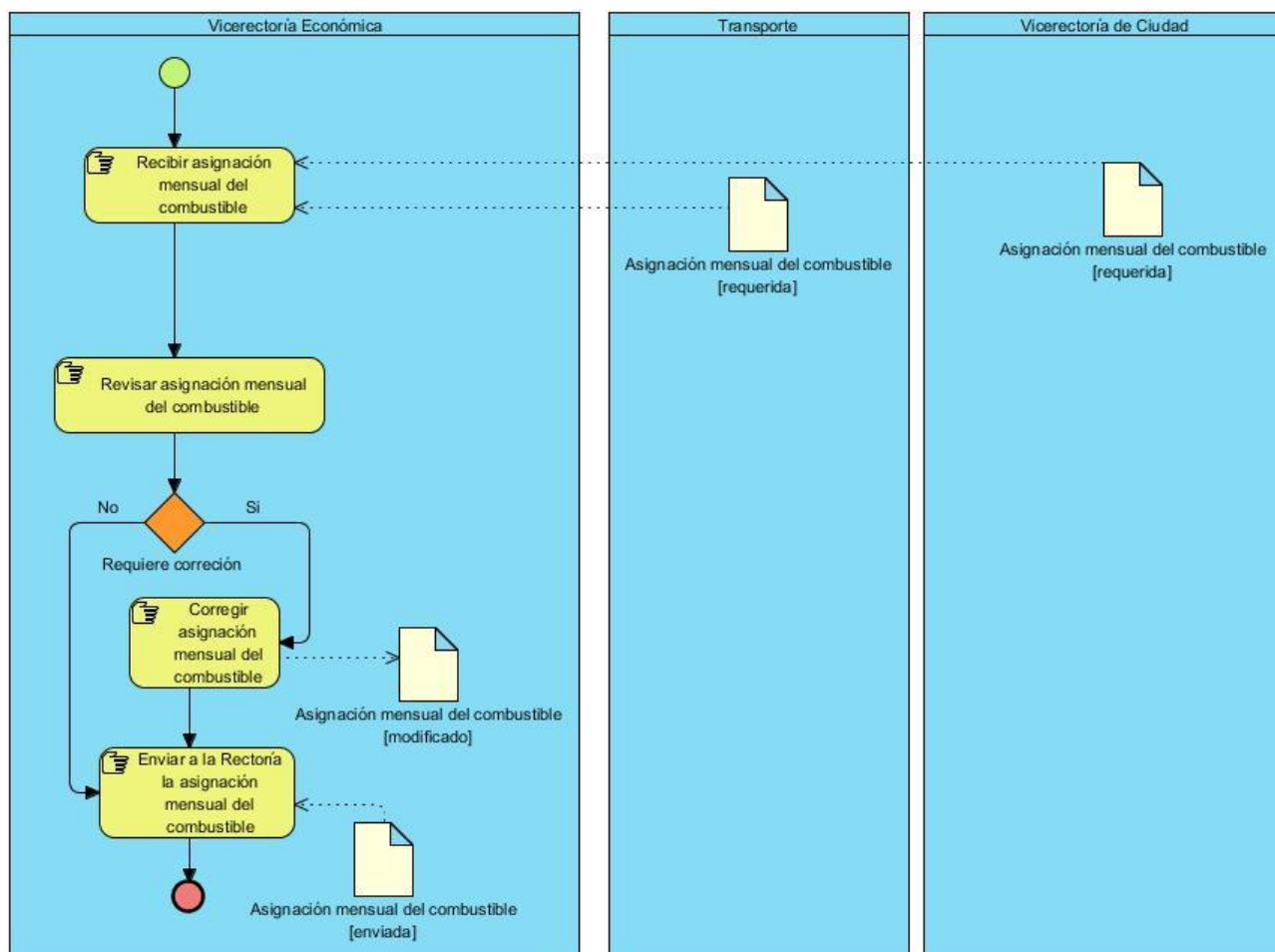


Figura 9: Diagrama del proceso aprobar distribución mensual del combustible de la Vicerrectoría Económica.

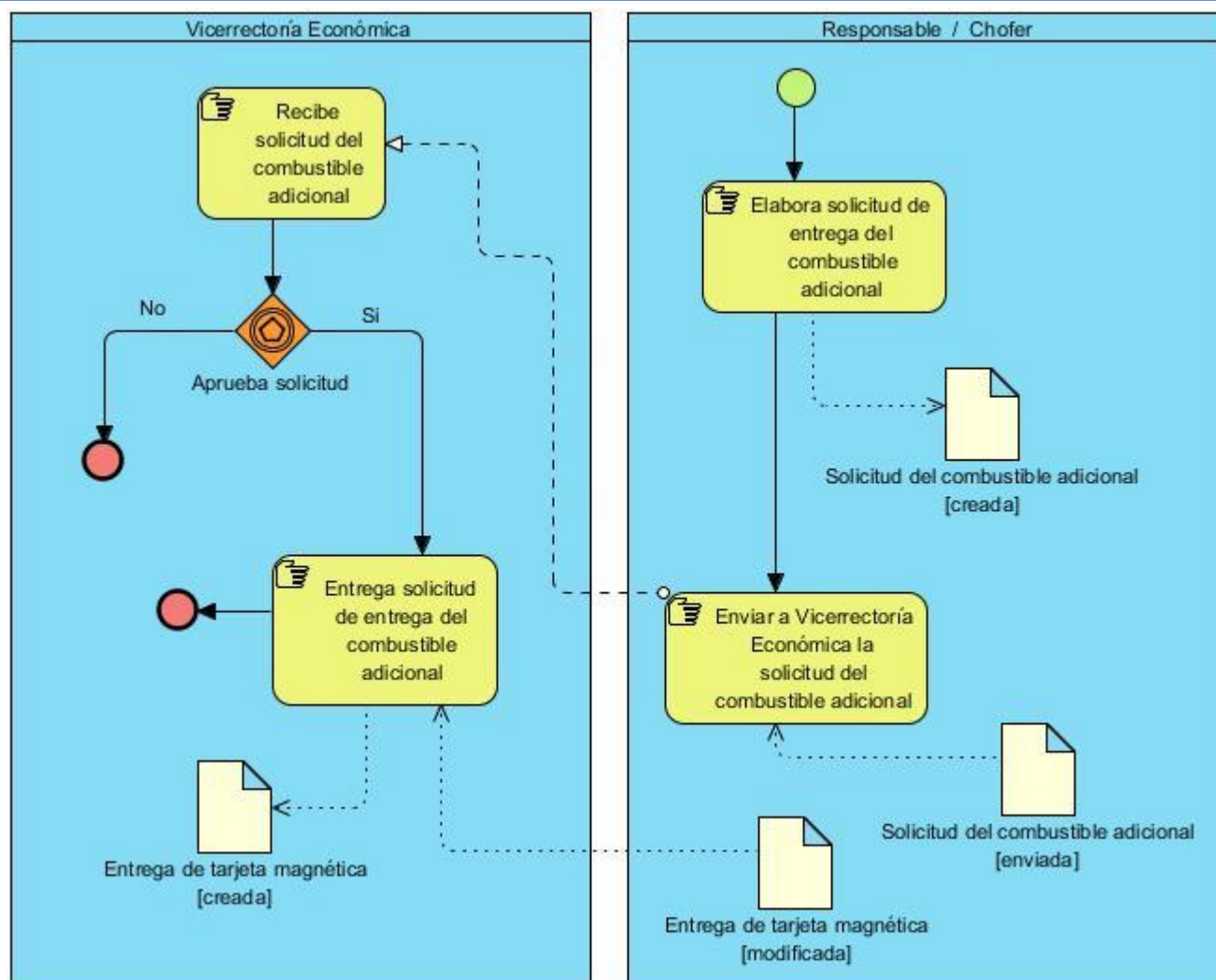


Figura 10: Diagrama del proceso aprobar entrega del combustible adicional de la Vicerrectoría Económica.


No. RF	Descripción del requisito	Prioridad
RF 1	Autenticar usuario	Alta
RF 2	Exportar reporte en formato .pdf	Media
RF 3	Visualizar ayuda	Baja
RF 4	Visualizar descripción del sistema	Baja
Módulo Configuración		
RF 5	Modificar usuario	Media
RF 6	Mostrar usuario	Media
RF 7	Eliminar usuario	Media
RF 8	Listar usuario	Media
RF 9	Gestionar estructura	Alta
RF 10	Listar estructura	Media

RF 11	Gestionar dependencia	Alta
RF 12	Listar dependencia	Media
RF 13	Gestionar tipo de combustible	Alta
RF 14	Listar tipo de combustible	Media
RF 15	Gestionar tipo de combustible	Alta
RF 16	Gestionar trabajador autorizado	Media
RF 17	Listar trabajador autorizado	Baja
Módulo de la Dirección de Energía		
RF 18	Gestionar planificación anual de asignación del combustible	Alta
RF 19	Buscar planificación anual del combustible	Alta
RF 20	Gestionar plan operativo mensual	Alta
RF 21	Buscar plan operativo mensual	Alta
RF 22	Generar reporte de eficiencia del consumo del combustible	Alta
RF 23	Gestionar asignación mensual del combustible	Alta
RF 24	Buscar asignación mensual del combustible	Alta
Módulo de la Dirección de Transporte		
RF 25	Gestionar cierre del combustible	Alta
RF 26	Buscar cierre del combustible	Baja
RF 27	Gestionar vehículo en parque de equipo	Alta
RF 28	Buscar vehículo en parque de equipo	Baja
RF 29	Consultar plan operativo mensual	Alta
Módulo de la Vicerrectoría Económica		
RF 30	Gestionar solicitud del combustible adicional	Alta
RF 31	Buscar solicitud del combustible adicional	Alta
RF 32	Gestionar modelo de entrega de tarjeta magnética	Alta
RF 33	Buscar modelo de entrega de combustible adicional	Alta
RF 34	Aprobar solicitud del combustible adicional	Alta
RF 35	Consultar asignación mensual del combustible	Alta
RF 36	Aprobar asignación mensual del combustible	Alta
RF 37	Consultar asignación anual del combustible	Baja
RF 38	Aprobar asignación anual del combustible	Media

Tabla 1: Requisitos funcionales del sistema.

Objetivo	Buscar plan operativo mensual de combustible.
Actor	Especialista de Energía.
Resumen	El caso de uso se inicia cuando el actor accede a la opción: “Buscar” , y luego selecciona la opción: “Plan Operativo” . El sistema brinda la posibilidad de introducir criterios de búsqueda para mostrar el plan operativo mensual, el Especialista de Energía introduce los datos que considera como criterios para realizar una búsqueda. El sistema busca y muestra los planes que cumplen con los criterios de búsqueda seleccionados, y le permite al Especialista de Energía Ver y Modificar los planes operativos mensuales, y así finaliza el caso de uso.
Complejidad	Media
Prioridad	Alta

Precondiciones		El usuario debe estar autenticado en el sistema con privilegios de administración en el Módulo de Energía.
Postcondiciones		Se listaron los planes operativos mensuales del combustible que cumplen con los criterios de búsqueda introducidos por el Especialista de Energía.
Flujo de eventos		
Flujo básico Buscar plan operativo mensual		
	Actor	Sistema
1.	El caso de uso se inicia cuando el actor accede a la opción: “Buscar” , y luego selecciona la opción: “Plan Operativo” .	
2.		Brinda la posibilidad de introducir los criterios elementales de búsqueda: <ul style="list-style-type: none"> • Mes • Año y permite realizar las siguientes operaciones: <ul style="list-style-type: none"> • Buscar plan operativo mensual dado criterios. • Cancelar operación.
3.	Introduce y selecciona los datos que considera como criterios para realizar una búsqueda y selecciona la opción de “Buscar” , dado criterios.	
4.		Valida los datos según el criterio de búsqueda introducido por el usuario.
5.		Busca los datos del Plan operativo mensual del combustible que cumplen con los criterios de búsqueda seleccionados.
6.		Comprueba que se encuentre información que cumpla con los criterios de búsqueda seleccionados.
7.		Muestra un listado de los planes operativos mensuales del combustible que cumplen con los criterios de búsqueda seleccionados, mostrando los siguientes atributos: <ul style="list-style-type: none"> •Creado por •Fecha de creación Ordenados ascendentemente por la Fecha de creación.
8.		Permite: <ul style="list-style-type: none"> • Ver: los planes operativos mensuales del combustible, son visualizados. <u>Ver Sección 1: “Mostrar plan operativo mensual”.</u> • Modificar: los planes operativos mensuales del combustible, son modificados. <u>Ver Sección 2: “Modificar plan operativo mensual”.</u>

9.		Termina el caso de uso
Flujos alternos		
1.- Cancelar operación		
	Actor	Sistema
2.1	Selecciona la opción de “Cancelar”.	
2.2		Regresa a la vista anterior.
2.3		El caso de uso finaliza.
2.- Validar los datos según criterio de búsqueda introducidos		
4.1		Muestra un indicador sobre los campos incorrectos.
4.2		Regresa al paso 3 del Flujo básico de Buscar plan operativo mensual .
3.- No se encuentra información que cumpla con los criterios de búsqueda		
5.1		Muestra el mensaje de confirmación de la acción: “ No se encontró información que cumpla con los criterios de búsqueda ”.
5.2		Regresa al paso 2 del Flujo básico de Buscar plan operativo mensual .
		
Sección 1: “Mostrar plan operativo mensual”		
Flujo básico Mostrar plan operativo mensual		
	Actor	Sistema
1.		Se muestran los datos del plan operativo mensual del combustible seleccionado. <u>Ver CU Mostrar plan operativo mensual</u>
2.		Termina Sección 1.
Flujos alternos		

1.- Cancelar operación		
1.1	Selecciona la opción de “Cancelar”.	
1.2		Regresa a la vista anterior.

SIGECOM
Grupo de Gestión Energética

Plan Operativo Mensual

Fecha 05-2014

Diesel Compra TM	11212	.00	Gasolina Compra TM	12515	.00
Diesel Inventario TM	545	.00	Gasolina Inventario TM	454	.00
Diesel Compensado	545	.00	Gasolina Compensado	44	.00
Diesel Privado	545	.00	Gasolina Privado	45	.00
Diesel Tiro Directo	454	.00	Gasolina Tiro Directo	54	.00
Diesel Estatal CUC	66666	.00	Gasolina Estatal CUC	45	.00

Sección 2: “Modificar plan operativo mensual”		
Flujo básico Modificar plan operativo mensual		
	Actor	Sistema
1.		Se modifican los datos del plan operativo mensual del combustible seleccionado. <u>Ver CU Modificar plan operativo mensual</u>
2.		Termina Sección 2.
Flujos alternos		
1.- Cancelar operación		
1.1	Selecciona la opción de “Cancelar”.	
1.2		Regresa a la vista anterior.

SIGECOM

Asignación Anual

Plan Operativo Mensual

Eficiencia de Consumo

Buscar

Ayuda

SIGECOM

Grupo de Gestión Energética

Editar Plan Operativo Mensual

Fecha

05-2014

Diesel Compra TM	11212	.00	Gasolina Compra TM	12515	.00
Diesel Inventario TM	545	.00	Gasolina Inventario TM	454	.00
Diesel Compensado	545	.00	Gasolina Compensada	44	.00
Diesel Privado	545	.00	Gasolina Privada	45	.00
Diesel Tiro Directo	454	.00	Gasolina Tiro Directo	54	.00
Diesel Estatal CUC	66666	.00	Gasolina Estatal CUC	45	.00

Aceptar

Cancelar

Relaciones	CU Incluidos	
	CU Extendidos	<ul style="list-style-type: none">Mostrar plan operativo mensual en el CU Mostrar plan operativo mensual.Modificar plan operativo mensual en el CU Modificar plan operativo mensual.
Requisitos no funcionales		
Asuntos pendientes		

Tabla 10: Descripción del caso de uso: Buscar plan operativo mensual.