



Universidad de las Ciencias Informáticas

Facultad 3

**Sistema de Gestión para la distribución y consumo del combustible en
la Universidad de las Ciencias Informáticas**

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



Autores:

Rachel Pérez Cruz

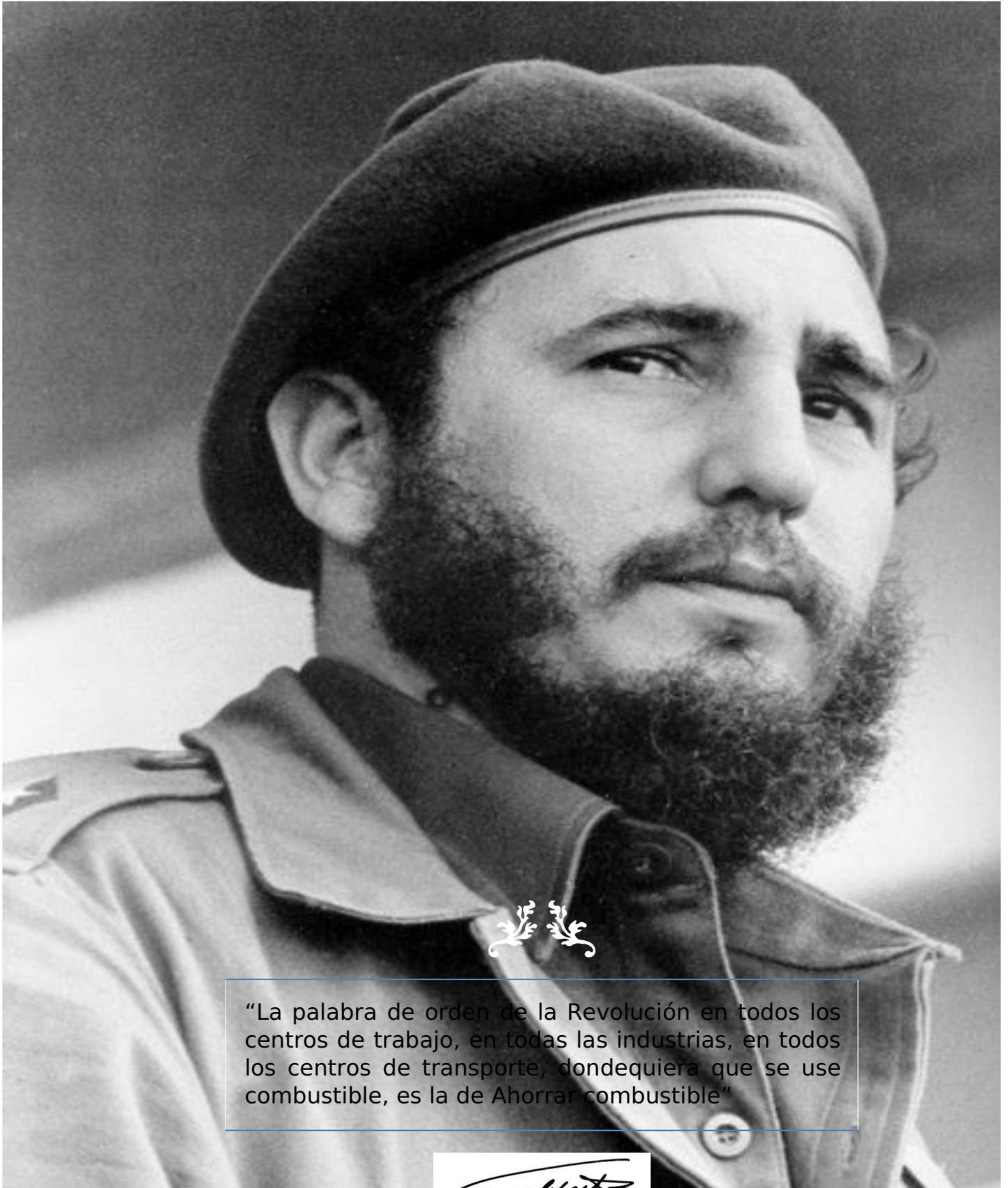
Neysi Maryolys Linares Forten

Tutores:

MSc. Radel Calzada Pando

Ing. Raciel Roche Escobar

Lic. José Manuel de León Cano



“La palabra de orden de la Revolución en todos los centros de trabajo, en todas las industrias, en todos los centros de transporte, dondequiera que se use combustible, es la de Ahorrar combustible”

Escobar

DECLARACIÓN JURADA DE AUDITORÍA

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

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

Autores:

Rachel Pérez Cruz

Forten

Neysi Maryolys Linares

Tutores:

MSc. Radel Calzada Pando

Lic. José Manuel León

Ing. Raciél Roche Escobar

AGRADECIMIENTOS

Son muchas las personas a las cuales quisiera agradecer, porque cada una de ellas me ha brindado enseñanzas que atesoraré por siempre. Empezaré por agradecer a la persona que aprecio y cuyas palabras siempre dejan eco en mí, a mi papá Gabriel Linares Gaitán, él ha sido siempre mi impulso, el primer motivo por el cual decidí cursar estudios universitarios y la primera persona que me venía a la mente cada vez que me desesperaba a la mitad del camino o disminuían mis esperanzas, y sé que nunca se cansó de darme ánimos.

Agradecer a mis madres Mercedes Forten Pérez e Isis Berniz Rodríguez por darme su apoyo incondicional y estar en las buenas y malas conmigo.

A mi tía Deysi Linares, a mi abuela Mercedes y a todos mis hermanos, que cada uno de ellos me han aportado grandes enseñanzas y profundas reflexiones que me han servido de gran ayuda en momentos en los que no he sabido qué hacer, y me brindaron apoyo cada vez que lo he necesitado.

Mis agradecimientos también van para mis tutores Raciél Roche y Radel Calzada que más que tutores han sido una familia más para mí, por siempre estar a mi lado cuando más los necesitaba.

A mi compañera de tesis Rachel Pérez por soportarme todos los días con mis cambios de humor, un beso grande para ti, unidas hacemos un gran equipo.

A mi ahijada, amiga y hermana Karla que llevamos juntas desde la secundaria que nuestra amistad es para la historia, agradecerte a ti por siempre tenerme presente en tu vida.

A mis ahijados Carlos, Yoel, Pedro y Joenis, los quiero mucho, por brindarme su hombro cuando más lo necesitaba, por entregarme ese cariño inmenso, no puedo hablar mucho sobre estudio porque cuando nos unimos es para fiesta y vacilón nada más.

Al grupo de la Gran familia, el grupo más loco y versátil de la universidad, por estar siempre juntos y cuidarnos entre nosotros.

A mi amiga Dailet Manuela que es una persona en la cual puedo confiar ciegamente, una persona que me apoyó todos estos años en mi carrera, a ella también va dedicado mi triunfo.

A todas las amistades del 3503, especialmente a Mebys, Alexei, Yordan y Alejandro por las casas de estudio que conformamos para salir bien en lo exámenes.

Neysi Maryolys Linares Forten

AGRADECIMIENTOS

A mi familia le dedico este día y estas palabras, porque gracias a ustedes me he convertido en la persona que soy. Fueron, son y siempre serán mi escuela, mi modelo a seguir, mi fuerza. Gracias por enseñarme a lograr mis metas en la vida, a superar los obstáculos sin importar qué, a sacudirme el polvo después de caer, levantarme y caminar con la cabeza erguida. Por su apoyo, esfuerzo y amor incondicional, GRACIAS.

Especialmente a mi mamá, te quiero mucho por ser una madre especial, por cuidar de mí y guiarme por el buen camino. A mi papá y a mi abuelito les dedico este triunfo aunque ya no están conmigo, yo sé que desde el cielo ellos me envían sus bendiciones. A mi hermano Abdel, por ser mi hermanito lindo, que siempre cuida de mí. A mi futuro esposo Duniel por todos estos años que llevamos juntos, por cuidarme y preocuparse por mí.

Mis agradecimientos también van a todas mis amistades, que se preocuparon por la evolución de mi tesis. A mis tutores Raciél y Radel por su apoyo incondicional.

A todas las amistades del 3503 por ser un grupo tan unido y a mi profesor guía Leandro por ser tan ejemplar.

Rachel Pérez Cruz

RESUMEN

En la actualidad se hace necesario llevar un control del consumo del combustible, pues el uso de este recurso genera grandes gastos para las empresas, por lo que un ahorro representaría una reducción significativa en sus gastos. La Universidad de las Ciencias Informáticas (UCI) no está exenta de dicho problema, ya que diariamente desarrolla una serie de actividades que demandan un consumo de combustible, ocasionando que se requiera distribuir y consumir el mismo. El área principal encargada de este proceso, es la Vicerrectoría económica, que a través de sus áreas fluyen los procesos de planificación, distribución y consumo del combustible. Actualmente la información relacionada con este proceso se gestiona de forma manual y a través de documentos Excel, lo que trae consigo problemas en cuanto a su seguridad, integridad y el acceso a la misma, además se dificulta la supervisión, debido a que los datos que se manejan en cada área se encuentran descentralizados. Como solución a esta problemática, se desarrolló el Sistema de Gestión para la distribución y consumo del combustible en la UCI, bajo la metodología AUP-UCI y siguiendo las políticas de soberanía tecnológica establecidas por la universidad. El funcionamiento de la solución desarrollada, fue verificado mediante la aplicación de pruebas de caja negra y caja blanca. Se realizó la validación de la investigación mediante las pruebas de aceptación y la aplicación de la técnica de ladov, donde se evidenció un alto nivel de satisfacción por parte del cliente.

Palabras claves: combustible, sistema de gestión.

INDICE

INTRODUCCIÓN.....	11
CAPÍTULO I: Fundamentación Teórica.....	16
1.1. INTRODUCCIÓN.....	16
1.2. PRINCIPALES CONCEPTOS ASOCIADOS A LA GESTIÓN DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE.....	16
1.3. ESTUDIO DEL ESTADO DEL ARTE.....	17
1.4. METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	21
1.5. TÉCNICAS PARA LA CAPTURA Y VALIDACIÓN DE REQUISITOS.....	22
1.6. PATRONES ARQUITECTÓNICOS.....	23
1.7. PATRONES DE DISEÑO.....	23
1.8. MÉTRICAS PARA LA VALIDACIÓN DEL DISEÑO.....	25
1.9. HERRAMIENTAS, TECNOLOGÍAS Y LENGUAJES UTILIZADOS.....	27
1.10. PRUEBAS DE SOFTWARE.....	31
1.11. TÉCNICA IADOV.....	33
1.12. CONCLUSIONES PARCIALES.....	34
CAPÍTULO II: Análisis y Diseño del sistema.....	36
2.1. INTRODUCCIÓN.....	36
2.2. MODELADO DEL NEGOCIO.....	36
2.2.1. Descripción de los Procesos de Negocios en la UCI.....	36
2.3. MODELO CONCEPTUAL.....	40
2.4. REQUISITOS DEL SOFTWARE.....	40
2.4.1. Listado de los Requisitos Funcionales.....	41
2.4.2. Evaluación de requisitos.....	44
2.4.3. Descripción de Requisitos.....	45
2.4.4. Técnica para Validar los requisitos funcionales.....	46
2.4.5. Listado de requisitos no funcionales.....	47
2.5. DISEÑO DE LA SOLUCIÓN.....	49
2.5.1. Arquitectura de software del sistema.....	49
2.5.2. Patrones de diseño y arquitectónico.....	51
2.5.3. Diagrama de clases.....	54
2.5.4. Modelo de datos.....	56
2.5.5. Mapeo de objetos a Base de datos (ORM).....	58
2.5.6. Validación del diseño.....	58
2.6. CONCLUSIONES PARCIALES.....	60
CAPÍTULO III: Implementación y pruebas del Sistema de Gestión para la distribución y consumo del combustible de la UCI.....	61
3.1. INTRODUCCIÓN.....	61
3.2. IMPLEMENTACIÓN.....	61
3.2.1. Diagrama de componentes.....	61
3.2.2. Diagrama de despliegue.....	62
3.2.3. Estándares de codificación.....	63
3.2.4. Tratamiento de errores.....	64
3.3. VALIDACIÓN DE LA SOLUCIÓN.....	64
3.3.1. Pruebas internas.....	64
3.3.2. Pruebas de aceptación.....	69

3.4. APLICACIÓN DE LA TÉCNICA IADOV.....	70
3.5. VALIDACIÓN DE LAS VARIABLES DE LA INVESTIGACIÓN.....	72
3.6. CONCLUSIONES PARCIALES.....	73
CONCLUSIONES GENERALES.....	75
RECOMENDACIONES.....	76
REFERENCIAS.....	76
ANEXOS.....	80
ANEXO #1. RESULTADO DE APLICACIÓN DE LA MÉTRICA TOC.....	80
ANEXO #2. RESULTADO DE APLICACIÓN DE LA MÉTRICA RC.....	82
ANEXO#3. ENCUESTA.....	83
ANEXO #4: DIAGRAMA DE PROCESO DE NEGOCIO: REALIZAR ASIGNACIÓN MENSUAL DE COMBUSTIBLE	85
ANEXO #5: DIAGRAMA DE PROCESO DE NEGOCIO: APROBAR ENTREGA DE COMBUSTIBLE ADICIONAL.	86
ANEXO #6: DIAGRAMA DE PROCESO DE NEGOCIO: RECOGIDA Y ENTREGA DE TARJETAS MAGNÉTICAS	87
ANEXO #7: DIAGRAMA DE PROCESO DE NEGOCIO: CARGAR COMBUSTIBLE EN TARJETAS MAGNÉTICAS	88
ANEXO #8: DIAGRAMA DE PROCESO DE NEGOCIO: LIQUIDAR COMBUSTIBLE.....	89
ANEXO #9: CUADRO LÓGICO DE IADOV.....	90
ANEXO #10: MÉTRICA TAMAÑO OPERACIONAL DE CLASE.....	90
ANEXO #11: RANGO DE VALORES PARA LA MÉTRICA TOC.....	91
ANEXO #12: RELACIONES ENTRE CLASES.....	91
ANEXO #13: RANGO DE VALORES PARA LA MÉTRICA RC.....	92

ÍNDICE DE TABLAS

TABLA 1.COMPARACIÓN ENTRE LOS SISTEMAS.....	20
TABLA 2.METODOLOGÍA DE DESARROLLO PARA LA ACTIVIDAD PRODUCTIVA DE LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS.....	22
TABLA 3. ÍNDICE DE SATISFACCIÓN DE IADOV.....	34
TABLA 4. LISTADO DE REQUISITOS FUNCIONALES.....	44
TABLA 5.RESULTADOS DE LA COMPLEJIDAD.....	45
TABLA 6.DESCRIPCIÓN DE REQUISITO: ADICIONAR DESAGREGACIÓN POR MESES DEL PLAN.....	46
TABLA 7.LISTADO DE REQUISITOS NO FUNCIONALES.....	49
TABLA 8. CASO DE PRUEBA PARA EL CAMINO #1.....	67
TABLA 9.CASO DE PRUEBA DEL REQUISITO: ADICIONAR VEHÍCULO PESADO.....	68
TABLA 10.CASO DE PRUEBA DEL REQUISITO: ADICIONAR VEHÍCULO PESADO.....	69
TABLA 11.CUADRO LÓGICO DE IADOV.....	70
TABLA 12.CUADRO DE SATISFACCIÓN INDIVIDUAL.....	71
TABLA 13.VALIDACIÓN DE LA VARIABLE DE INVESTIGACIÓN.....	73

ÍNDICE DE FIGURAS

FIGURA 1. DIAGRAMA DE PROCESO DE NEGOCIO: PLANIFICAR ASIGNACIÓN ANUAL Y MENSUAL DE COMBUSTIBLE.....	37
FIGURA 2. MODELO CONCEPTUAL DEL PROCESO DE DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE EN LA UCI.....	40
FIGURA 3. PROTOTIPO DE INTERFAZ DE USUARIO DEL REQUISITO: ADICIONAR DESAGREGACIÓN POR MESES DEL PLAN.....	47
FIGURA 4. PROTOTIPO DE INTERFAZ DE USUARIO DEL REQUISITO: ADICIONAR DESAGREGACIÓN POR MESES DEL PLAN.....	47
FIGURA 5. FUNCIONAMIENTO INTERNO.....	50
FIGURA 6. ESTRUCTURA DE CARPETAS.....	52
FIGURA 7. CÓDIGO DONDE SE EVIDENCIA PATRÓN CREACIONAL.....	53
FIGURA 8. CÓDIGO DONDE SE EVIDENCIA PATRÓN COMPORTAMIENTO.....	53
FIGURA 9. CÓDIGO DONDE SE EVIDENCIA PATRÓN DECORATOR.....	54
FIGURA 10. DIAGRAMA DE CLASES.....	55
FIGURA 11. DIAGRAMA DE ENTIDAD RELACIÓN.....	57
FIGURA 12. FRAGMENTO DE LA CLASE ACTIVIDAD DEL SISTEMA DE GESTIÓN PARA LA DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE EN LA UCI.....	58
FIGURA 13. REPRESENTACIÓN EN % DE LOS RESULTADOS DE LA EVALUACIÓN MEDIANTE LA MÉTRICA TOC EN LOS ATRIBUTOS: REUTILIZACIÓN, COMPLEJIDAD DE IMPLEMENTACIÓN Y RESPONSABILIDAD.....	59
FIGURA 14. REPRESENTACIÓN EN % DE LOS RESULTADOS DE LA EVALUACIÓN MEDIANTE LA MÉTRICA RC EN LOS ATRIBUTOS: ACOPLAMIENTO, COMPLEJIDAD DE MANTENIMIENTO, CANTIDAD DE PRUEBA Y REUTILIZACIÓN.....	59
FIGURA 15. DIAGRAMA DE COMPONENTES DEL SISTEMA DE GESTIÓN PARA LA DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE EN LA UCI.....	62
FIGURA 16. DIAGRAMA DE DESPLIEGUE DEL SISTEMA DE GESTIÓN PARA LA DISTRIBUCIÓN Y CONSUMO DEL COMBUSTIBLE EN LA UCI.....	63
FIGURA 17. EJEMPLO DE ESTÁNDAR DE CODIFICACIÓN EN ODOO.....	64
FIGURA 18. MÉTODO DEPENDENCIA_CALC.....	65
FIGURA 19. GRAFO DE FLUJO ASOCIADO AL MÉTODO DEPENDENCIA_CALC.....	65
FIGURA 20. RESULTADO DE LA APLICACIÓN DE LAS PRUEBAS INTERNAS.....	69

INTRODUCCIÓN

La disponibilidad de los recursos energéticos es uno de los factores en el desarrollo tecnológico de las naciones. A su vez, el desarrollo tecnológico determina la utilización de ciertos tipos de energía y, por lo tanto, la disponibilidad de ese recurso. La energía como recurso energético es utilizada para satisfacer algunas de las necesidades básicas en forma de calor y trabajo, esta dependencia energética ha acarreado un abuso de consumo de combustibles fósiles y recursos no renovables.

Dentro de los recursos no renovables se encuentran los combustibles fósiles, ya que no se reponen por procesos biológicos y en algún momento, se acabarán, y quizás sea necesario disponer de millones de años para que vuelvan a aparecer. El actual precio del combustible hace que la partida destinada a la compra del mismo suponga cada vez un desembolso para las empresas, por lo que un ahorro en su consumo significaría una reducción en sus gastos. (1)

Es por esta razón por la que muchas empresas deciden crear e implementar un sistema para la gestión de la información del combustible, ya que existen muchos factores de los que depende el consumo de este y hay que concienciar a la organización en todos sus niveles para conseguir buenos resultados. Por esto, la tarea de gestionar el consumo de combustible necesita ser estructurada y supervisada.

En la actualidad la presencia de las Tecnologías de la Información y las Comunicaciones (TIC) se hace imprescindible en las actividades de la vida cotidiana y a su vez está adquiriendo un lugar trascendental e incuestionable que está cambiando las formas de aprender y transmitir los conocimientos. Cuba además de estar inmersa en una verdadera revolución educacional se ha dado a la tarea de automatizar a la sociedad cubana. La Universidad de las Ciencias Informáticas (UCI) forma parte de esta gran misión, de informatizar empresas existentes en el país.

Actualmente en la UCI se desarrollan una serie de actividades que demandan el traslado diario de personas y de alimentos, ocasionando que se requiera distribuir y consumir combustible. En este proceso de distribución y consumo del combustible intervienen las áreas de: Dirección de Transporte, Grupo de Gestión Energética, Vicerrectoría Económica, el área de la Caja y el Departamento de Finanzas de la Dirección de Contabilidad y Finanzas, las cuales interactúan entre sí mediante el envío y respuestas de planificaciones y solicitudes. Por su parte, cada una de ellas desarrolla una serie de actividades para garantizar el correcto flujo y el control de la información.

Con el objetivo de diagnosticar el estado actual de la información asociada al proceso de distribución y consumo del combustible en la UCI, se realizaron entrevistas a las áreas que intervienen en este proceso, principalmente al área de la Vicerrectoría Económica, las cuales arrojaron los siguientes resultados:

La información relacionada con este proceso se gestiona de forma manual, en formato duro y a través de documentos Excel, ocasionando que se dificulte su búsqueda y se encuentre vulnerable al deterioro o alteraciones. Actualmente, generar los reportes solicitados por la Vicerrectoría Económica se entorpece debido a la gran cantidad de información que se genera por cada proceso. Los datos son manejados desde diferentes áreas de trabajo lo que provoca descentralización de la información y obstaculiza las supervisiones realizadas por el Vicerrector Económico.

Otro factor que influye de forma negativa es que los especialistas que trabajan en el Grupo Energético les resulta muy engorroso realizar el cierre mensual del consumo del combustible, ya que tienen que hacer muchos cálculos utilizando las fórmulas que brinda el Excel en función de obtener esta información. Por otra parte, las entregas de tarjetas magnéticas y las liquidaciones de consumo de combustible se ven afectadas, ya que para entregar la información contable actualizada, se registran los asientos de forma manual, provocando que se cometan errores de cálculo y que existan desajustes contables en los balances de comprobación. Esto sucede todos los meses propiciando que la distribución del combustible al parque automotor no posea la inmediatez necesaria y exista retraso en la circulación de las informaciones que se van aprobando por cada factor. Además de que existe el riesgo de que personas externas a la entidad tengan acceso a informaciones generadas en el proceso.

Estos inconvenientes no están en correspondencia con el nivel de control que se pretende lograr en este proceso.

De la problemática anteriormente descrita, se deriva el **problema de la investigación**: ¿Cómo mejorar la gestión de la información asociada al proceso de distribución y consumo de combustible en la UCI, en cuanto a la organización, supervisión, acceso y seguridad?

El **objeto de estudio de la investigación** se enmarca en el proceso de desarrollo de software para la distribución y consumo del combustible y el **campo de acción** es el proceso de desarrollo de software para la distribución y consumo del combustible en la UCI.

Para darle solución al problema antes descrito se tiene como **objetivo general**: Desarrollar un sistema informático, que contribuya a mejorar la gestión de la información asociada al proceso de distribución y consumo de combustible en la UCI en cuanto a la organización, supervisión, acceso y seguridad.

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

- Definir el marco teórico de la investigación mediante el estudio y el análisis de los principales referentes teóricos para el desarrollo de la solución.
- Realizar el análisis y diseño del sistema para la distribución y consumo del combustible en la UCI.
- Implementar el sistema para la distribución y consumo del combustible en la UCI.
- Validar la solución propuesta mediante pruebas de software para darle cumplimiento al objetivo de la investigación.
- Validar la variable dependiente mediante la técnica ladov.

Se defiende la idea: El desarrollo de un sistema informático permitirá mejorar la gestión de la información asociada al proceso de distribución y consumo de combustible en la UCI en cuanto a la organización, supervisión, acceso y seguridad.

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

1. Revisión de documentos y entrevista a los directivos de las áreas involucradas en el proceso.
2. Investigación de los sistemas de gestión para la distribución y consumo de combustible existente a nivel 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. Implementación de los componentes necesarios para desarrollar el Sistema de gestión para la distribución y consumo de combustible en la UCI.

8. Elaboración y realización de las pruebas para validar el comportamiento del sistema.

Con el propósito de desarrollar las tareas planteadas para la investigación se utilizarán 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 están 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 para 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.

Métodos Empíricos:

Entrevista: Se realiza una entrevista a los especialistas de todas las áreas involucradas en la distribución y consumo de combustible, con el objetivo de lograr una mayor comprensión de las particularidades y características del proceso de distribución del combustible. De la entrevista, se obtendrá 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:

1. El sistema debe ser capaz de brindar un conjunto de funcionalidades que permita obtener una correcta administración sobre el consumo y distribución del combustible en la UCI.
2. Proporcionar la información necesaria con rapidez y calidad entre las áreas involucradas en el proceso.

3. Garantizar la seguridad de las informaciones que se generen en el sistema.

El trabajo está estructurado en tres capítulos, descritos a continuación:

Capítulo I: Fundamentación teórica

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

Capítulo II: Análisis y Diseño del Sistema de Gestión para la distribución y consumo del combustible de la UCI

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 productos de trabajo derivados de la metodología de desarrollo de software que se seleccione. También 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 III: Implementación y pruebas del Sistema de Gestión para la distribución y consumo del combustible de la UCI

En este capítulo se abordan aspectos relacionados con la implementación del sistema en base a la arquitectura de desarrollo de software, se describen los estándares de codificación empleados durante la implementación, para propiciar legibilidad, claridad y una mayor comprensión del código por parte de los desarrolladores. Se efectúa el tratamiento de errores para brindar información oportuna ante eventualidades. Se emplean las pruebas de verificación y validación, haciendo uso de los métodos de pruebas de caja blanca y de caja negra, además se valida la variable de investigación y se realiza la técnica ladov para medir la satisfacción del cliente.

CAPÍTULO I: Fundamentación Teórica

1.1. Introducción

En el presente capítulo se define el marco teórico referencial de la investigación. Se realiza un análisis de los conceptos asociados al dominio del problema y se estudia el estado de arte de los procesos y sistemas de gestión existentes en la actualidad para el control del combustible. Se describen además, la metodología de software estudiada, las tecnologías y las herramientas que serán utilizadas en el desarrollo del sistema.

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

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. (2)

Combustible

Cualquier material capaz de liberar energía cuando se oxida de forma violenta con desprendimiento de calor. (3)

Otras bibliografías asocian el término como cuerpo o una sustancia que puede arder, sobre todo si con ello produce energía. (3)

Gestión de combustible

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

Una adecuada gestión de combustible está asociada:

- 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. (4)

Sistema de gestión

Es un conjunto de etapas unidas en un proceso continuo que permiten trabajar ordenadamente una idea hasta lograr mejoras y su continuidad. (5)

Sin embargo para autores como Delgado y Mendoza consiste en la gestión (planeamiento, organización, operaciones y control) de los recursos (humanos y físicos) que tienen que ver con el apoyo a sistemas (desarrollo, mejora y mantenimiento) y servicios (procesamiento, transformación, distribución, almacenamiento y recuperación) de la información (datos, textos, voz e imagen) para una empresa. (6)

1.3. Estudio del estado del arte

Con el objetivo de verificar si existen algunos sistemas que satisfagan las necesidades de la Vicerrectoría económica, se realizó un estudio de las tendencias actuales de los sistemas de gestión para la distribución y consumo del combustible a nivel nacional e internacional. La investigación se realizó teniendo en cuenta los siguientes indicadores: gestionar vehículos, planificar asignación de combustible, realizar cierre de consumo de combustible, entregar combustible por tarjetas magnéticas y realizar liquidación del consumo de combustible. A partir de la búsqueda efectuada y los indicadores seleccionados se realizó un análisis de los siguientes sistemas.

Sistemas de gestión para la distribución y consumo del combustible a nivel internacional:

Gestión de flota

El sistema de Gestión de flotas de Informática EUGCOM fue 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. Además 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. Este sistema evita que los informes de

combustible contengan información no válida después de editar o eliminar un movimiento. (7)

KreaFlota

Es un software diseñado y concebido para controlar los aspectos técnicos de cualquier flota vehicular, dispone de varios módulos para la correcta gestión de cada uno de los vehículos que disponen las empresas. 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 y pólizas de seguro. Además, se lleva un control del petróleo, llevando un registro diario de consumo de los vehículos, permitiendo obtener el rendimiento de los mismos (8).

FlotaWeb

FlotaWeb es una plataforma de Administración de Flotas basado en Internet, que permite gestionar todos los procesos relacionados con la flota vehicular. Funciona perfectamente para flotas de cualquier tamaño o sector, ya que habilita funcionalidades de acuerdo a sus necesidades. Puede controlar y supervisar los costos de combustible en la flota, analizar rendimientos, obtener informes estadísticos de los detalles de consumo por proveedores, ciudades, centros de costo y operaciones. Permite centralizar todos los reportes de los conductores y tareas pendientes de mantenimiento en una sola aplicación. Este sistema está compuesto por la configuración de las rutas, sus duraciones en kilómetros y días, además de la pre-configuración de los gastos de una ruta. (9)

Sistema de Gestión de Flotas de Odoo

Es un sistema que integra diferentes funciones de una empresa de transporte, como las cuentas, inventarios, nóminas, facturación, seguros, gestión de activos, mantenimiento de los vehículos y los datos del conductor. Además, el sistema permite una completa personalización según las necesidades específicas del cliente. Presenta un historial de la flota que incluye todos los detalles pertinentes de la flota como número de modelo, asientos, la matrícula, tipo de combustible y los detalles de inscripción, además mantiene un registro completo de los datos del conductor, como nombre y permiso de conducir. (10)

Sistemas de gestión para la distribución y consumo del combustible a nivel nacional:

Apolo

Es un producto nacional perteneciente al Ministerio de 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, donde la mayoría de la información que se brinda es directamente definible por un usuario de nivel administrativo del sistema (11).

Sistema de Gestión de mantenimiento vehicular (Orbita)

El Sistema de Gestión de mantenimiento vehicular realizado por el Centro de Informatización para la Gestión de Entidades (CEIGE) tiene como propósito gestionar los procesos de mantenimiento que se desarrollan en el área de Transporte de la UCI. Este sistema fue desarrollado basado en los principios de independencia tecnológica utilizando software libre. Mediante su utilización en la Dirección de Transporte se puede gestionar las asignaciones y recepciones de combustible, las inspecciones técnicas, las órdenes de trabajo e informes de baja de manera 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 y garantizando una correcta planificación, organización y control en la gestión del mantenimiento de los vehículos. (11)

Los sistemas nacionales e internacionales anteriormente mencionados se analizaron con el objetivo de conocer cómo gestionaban toda la información relacionada con la planificación, distribución y consumo del combustible y determinar si satisfacen en su totalidad a las necesidades existentes en la Vicerrectoría Económica.

El análisis realizado teniendo en cuenta los indicadores definidos arroja los siguientes resultados:

Leyenda:

- ∩ El sistema realiza el proceso que se define en la tabla.
- X El sistema no realiza el proceso que se define en la tabla.

Procesos del negocio.	Sistemas					
	KreaFlot a	FlotaWe b	Gestión de Flotas	Sistema Gestión de Flota de Odo	Apol o	Orbit a
Gestionar vehículos	∩	∩	∩	∩	∩	∩
Planificar asignación de combustible	∩	∩	∩	∩	∩	∩
Realizar cierre de consumo de combustible	∩	∩	∩	∩	∩	∩
Gestionar tarjetas magnéticas	X	X	X	X	X	X
Realizar entrega y liquidación de combustible por tarjetas.	X	X	X	X	X	X

Tabla 1.Comparación entre los sistemas

Por tanto, los resultados obtenidos a partir de los indicadores anteriormente definidos fueron:

Como se muestra en la tabla anterior existen sistemas que realizan procesos relacionados con el negocio, a pesar de que algunos de estos difieren del proceso que se ejecuta en la universidad, no se excluyen en su totalidad, ya que presentan varias funcionalidades que pueden ser reutilizadas en la propuesta de solución que se desea.

Resaltar que el estudio del Sistema de Gestión de mantenimiento vehicular, aportó varios elementos en cuanto a la asignación de combustible por vehículo y los cierres de consumo, ya que al pertenecer a la universidad conoce cómo se desarrollan estos procesos. En cuanto al Sistema de Gestión de Flota desarrollado sobre la plataforma Odo, se heredó la gestión vehicular que posee, adaptándola a las necesidades del negocio, además esta plataforma permitió reutilizar el sistema de control de acceso y la gestión de seguridad que tiene implementada, creando grupos y usuarios en dependencia de los roles y privilegios que existen.

Sin embargo estos sistemas no incluyen otros elementos significativos del negocio que se desean resolver, como son:

- No gestionan tarjetas magnéticas.
- No realizan la entrega y liquidación de combustible por tarjetas.

Por lo que se decidió desarrollar un Sistema de Gestión para la distribución y consumo del combustible en la UCI.

Luego de realizar un estudio sobre los sistemas de gestión para la distribución y consumo de combustible existentes, se procede a caracterizar la metodología de desarrollo de software seleccionada para la investigación.

1.4. Metodologías 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. (12)

Se encarga de estructurar, planificar y controlar el proceso de desarrollo del software, a través de un conjunto de procedimientos, técnicas, herramientas y soporte documental que ayuda a los desarrolladores a realizar un nuevo software. (12)

La UCI para estandarizar el proceso productivo decidió crear una nueva metodología, la metodología AUP-UCI (Metodología de desarrollo para la Actividad productiva de la Universidad de las Ciencias Informáticas), la cual se basa en una variación de la metodología AUP (Proceso Unificado Ágil) en unión con el modelo Capability Maturity Model Integration para el desarrollo (CMMI-DEV) v 1.3. (13)

A continuación se muestra las fases de la metodología AUP variación UCI:

FASES AUP VARIACIÓN UCI	OBJETIVOS DE LAS FASES(AUP-UCI)
INICIO	Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del

	alcance del proyecto, realizar estimaciones de tiempo, esfuerzo, costo y decidir si se ejecuta o no el proyecto.
EJECUCIÓN	En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto, considerando los requisitos y la arquitectura. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elabora la arquitectura y el diseño, se implementa y se libera el producto. Durante esta fase el producto es transferido al ambiente de los usuarios finales o entregado al cliente. Además en la transición se capacita a los usuarios finales sobre la utilización del software.
CIERRE	En esta fase se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

Tabla 2. Metodología de desarrollo para la actividad productiva de la Universidad de las Ciencias Informáticas.

1.5. Técnicas para la captura y validación de requisitos.

Para conocer las necesidades del cliente, es necesario realizar la captura de requisitos, en este proceso el equipo de desarrollo identifica de diferentes fuentes de información, los datos que son necesarios para conocer las funcionalidades a desarrollar por el sistema. Una vez realizada la captura de requisitos, se emplean técnicas de validación, para demostrar que la definición de los requisitos puntualiza realmente lo que el cliente desea y comprobar que los requisitos identificados no presenten ambigüedades, inconsistencias, omisiones o sea inalcanzables. (14):

Las técnicas para la captura de requisitos que se utilizan en la presente investigación son:

- **Entrevista:** Consiste en establecer una conversación entre personas de ambas partes para obtener información sobre el negocio y a partir de éstos se definen los requisitos.
- **Estudio de documentación:** Consiste en realizar una lectura basada en

documentos sobre el dominio del negocio de la organización o sus prácticas profesionales. Ejemplos de algunos documentos que se pueden consultar son: manuales de usuarios del sistema actual, normativas y legislaciones.

Las técnicas utilizadas en la presente investigación para validar los requisitos son:

- **Revisión técnica formal:** Para la revisión técnica formal debe existir un equipo de revisión que incluya ingenieros del sistema clientes, usuarios, y otros interesados que examinan la especificación. Estos buscan errores en el contenido o en la interpretación, áreas donde se necesitan aclaraciones, información incompleta, inconsistencias, requisitos contradictorios, o requisitos irreales o inalcanzables. La revisión es esencial para asegurarse que el cliente y el desarrollador tienen el mismo concepto del sistema. Es un proceso manual que involucra a personas tanto de la organización del cliente como de la del contratista, se verifica el documento de requisitos en cuanto a anomalías y omisiones. (15)
- **Construcción de prototipos:** En este enfoque de validación se muestra un modelo ejecutable del sistema a los usuarios finales y a los clientes. Estos pueden experimentar con este modelo para ver si cumple sus necesidades reales. El prototipo puede ser funcional o no. (16)

1.6. Patrones arquitectónicos

Los patrones arquitectónicos ofrecen soluciones a problemas de arquitectura en ingeniería de software. Dan una descripción de los elementos y el tipo de relación que tienen, junto a un grupo de restricciones sobre cómo pueden ser usados. Un patrón arquitectónico expresa un esquema de organización estructural esencial para un sistema de software, que consta de subsistemas, sus responsabilidades e interrelaciones. (17)

Patrón arquitectónico Modelo-Vista-Controlador (MVC)

Dentro de los patrones arquitectónicos existentes, se encuentra el patrón de arquitectura de las aplicaciones de software MVC, el cual resulta uno de los más empleados para el desarrollo de aplicaciones web. Este patrón está dividido en tres componentes de aplicación que separan la interfaz de usuario, el control de la lógica del negocio y el acceso a los datos en componentes diferentes. (18)

- **Modelo:** contiene la principal funcionalidad y la información con la que trabaja la aplicación. Integra las clases que contienen las consultas a la base de datos, así como la definición de las tablas, sus relaciones y atributos.
- **Vista:** transforma el modelo en una página web que permite al usuario interactuar con esta, mostrándole así la información. Además determina la interfaz que se muestra finalmente al cliente para interactuar con la aplicación.
- **Controlador:** se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista, dándole respuestas a las peticiones de los usuarios.

1.7. Patrones de diseño

Los patrones de diseño en la investigación permiten facilitar el trabajo, al emplear un conjunto de buenas prácticas, además proporcionan una estructura conocida por todos los programadores de manera que la forma de trabajar no resulte distinta entre los mismos. Los patrones de diseño son soluciones simples y elegantes a problemas específicos y comunes del diseño Orientado a Objetos, basadas en la experiencia, estos permiten reutilizar la experiencia de los desarrolladores, clasificar y describir formas de solucionar problemas que ocurren de forma frecuente en el desarrollo y están basados en la recopilación del conocimiento de los expertos en desarrollo de software. (12)

Patrones GRASP

Para el desarrollo del sistema se aplican los patrones de diseño GRASP, los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos. Entre los más conocidos y utilizados para la solución están: (12)

- **Experto:** cada clase dentro del sistema tiene la responsabilidad de utilizar únicamente la información que ella misma posee para realizar la labor para la que fue concebida. Tal es el caso de las clases del modelo que son las encargadas de toda la lógica del acceso a los datos.
- **Creador:** identifica quien debe ser el responsable de la creación de nuevos objetos o clases, donde la nueva clase deberá ser creada por la clase que tiene toda la información necesaria para realizar la acción, que usa

directamente las instancias creadas del objeto, almacena o maneja varias instancias de clase y contiene o agrega la clase.

- **Bajo acoplamiento:** Es la medida en que cada una de las clases realiza actividades independientes, además de poseer un conocimiento de las actividades que realizan las otras clases del sistema permitiendo la reutilización.
- **Alta cohesión:** Asigna una responsabilidad de modo que la cohesión siga siendo alta. Se define que existe alta cohesión funcional cuando los elementos de un componente colaboran para producir algún comportamiento bien definido. El nivel de cohesión no debe ser independiente de otras responsabilidades ni de otros principios como los patrones Experto y Bajo Acoplamiento.
- **Controlador:** Se encarga de asignar la responsabilidad a una clase en el momento de manejar mensajes correspondientes a eventos en un sistema, facilitando la centralización de actividades. Esta clase es la responsable de manejar la lógica de negocio, decidiendo así que clase es la encargada de ejecutar una tarea determinada.

La aplicación de estos patrones se ve reflejada con la asignación de las responsabilidades por clases para realizar solo las funcionalidades correspondientes a la información que las mismas contienen, evitando una sobrecarga. Además, se da la responsabilidad de crear instancias de otras clases solamente a aquellas que contengan a las mismas y se modelan clases controladoras, encargadas de controlar el flujo de las operaciones del sistema. El diseño de la aplicación permite la interacción entre las mismas propiciando un bajo acoplamiento y en consecuencia la reutilización.

Patrones Banda de los Cuatro (GoF por sus siglas en inglés)

Adaptados para solucionar un problema común en particular mediante la descripción de las clases y la comunicación entre objetos; se agrupan en tres categorías: creacionales, estructurales y de comportamiento. (19)

Creacionales

- **Singleton** (instancia única): garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Restringe la instanciación de una clase o valor de un tipo a un solo objeto.

Estructurales

- **Decorator** (Decorador): Añade funcionalidad a una clase dinámicamente.

Comportamiento

- **Observer** (Observador): Define una dependencia de uno-a-muchos entre objetos, de forma que cuando un objeto cambie de estado se notifique y actualicen automáticamente todos los objetos que dependen de él.
- **Template Method** (Método plantilla): Define en una operación el esqueleto de un algoritmo, delegando en las subclases algunos de sus pasos, esto permite que las subclases redefinan ciertos pasos de un algoritmo sin cambiar su estructura.

1.8. Métricas para la validación del diseño

Para comprobar la calidad y fiabilidad del Sistema de Gestión para la distribución y consumo del combustible en la UCI, se hizo necesario emplear métricas para la validación del diseño.

Una métrica es una medida efectuada sobre los programas, la documentación, su desarrollo y mantenimiento que permite medir de forma cuantitativa la calidad de los atributos internos del software.

Lorenz y Kidd proponen un conjunto de métricas para la validación del diseño del software siguiendo el paradigma orientado a objetos, para ello las clasifican en cuatro amplios grupos: tamaño, herencia, valores internos y valores externos. Las métricas orientadas al tamaño de las clases se centran en el recuento de atributos y operaciones para cada clase y los valores promedio para el sistema como un todo; las métricas basadas en la herencia se centran en la forma en que las operaciones se reutilizan en la jerarquía de clases; las métricas para valores internos examinan la cohesión y los aspectos orientados al código y las métricas orientadas a valores externos examinan el acoplamiento y reutilización entre las clases. Una de las métricas más empleadas es Tamaño Operacional de Clases (TOC), pues permite medir el total de atributos y operaciones encapsulados en una clase para valorar la sobrecarga de responsabilidades asignadas. (12) (20)

Chidamber y Kemerer también realizan una propuesta para medir la calidad del diseño, entre las cuales se encuentran: Métodos Ponderados por Clases (MPC), Árbol de Profundidad de Herencia (APH), Numero de Descendientes (ND), Acoplamiento entre Clases Objeto (ACO), Relaciones entre Clases (RC) y Carencia de Cohesión de los Métodos (CCM). (12) (21)

Por la relevancia de la investigación se considera oportuno aplicar de Lorenz y Kidd la métrica TOC, pues permite visualizar si se distribuyen correctamente las asignaciones de responsabilidades entre las clases, verificándose así la cohesión y armonía entre las mismas, y de Chidamber y Kemerer las métricas RC para evaluar el grado de acoplamiento entre las clases.

Métrica Tamaño Operacional de las Clases (TOC): La métrica TOC está dada por la cantidad de funcionalidades contenidas en las clases, a partir de las cuales se determina la afectación que ejerce en el diseño. La misma comprende los siguientes atributos de calidad. (22)

- **Responsabilidad:** Responsabilidad que posee una clase en un marco conceptual correspondiente al modelado de la solución propuesta.
- **Complejidad de implementación:** Grado de dificultad que tiene que implementar un diseño de clases determinado.
- **Reutilización:** Significa cuán reutilizada es una clase o estructura de clase dentro de un diseño de software.

Para un mejor entendimiento de la reutilización de esta métrica ver Anexos #10 y 11.

Métrica Relaciones entre Clases (RC): La métrica RC está dada por la cantidad de relaciones existentes entre las clases contenidas en el diseño, a partir de las cuales se determina la afectación que éstas ejercen dentro de la eficiencia del sistema. Los indicadores medidos por esta métrica son los siguientes. (22)

- **Acoplamiento:** Dependencia o interconexión de una clase o estructura de clase respecto a otras.
- **Cantidad de pruebas:** Número o grado de esfuerzo necesario para realizar las pruebas de calidad al producto (componente) diseñado.
- **Complejidad del mantenimiento:** Nivel de esfuerzo necesario para sustentar, mejorar o corregir el diseño de software propuesto. Puede influir significativamente en los costes y la planificación del proyecto.
- **Reutilización:** Significa cuán reutilizada es una clase o estructura de clase dentro de un diseño de software.

Para un mejor entendimiento de la reutilización de esta métrica ver Anexos #12 y 13.

1.9. Herramientas, tecnologías y lenguajes utilizados

Para definir las herramientas, tecnologías y lenguajes a utilizar durante el desarrollo de la propuesta de solución, se realizó un estudio sobre las herramientas que utiliza la plataforma de Odoo, ya que el sistema propuesto está desarrollado sobre la misma.

Odoo v8.0

Es un ERP (sistema de planificación de recursos empresariales) producido por la empresa belga Odoo S.A, publicado bajo licencia de código abierto y que funciona completamente online desde cualquier navegador web. Presenta una estructura modular ya que esta plataforma está formada por varias aplicaciones, donde cada una de ellas representa un módulo. La nomenclatura aplicación o módulo significa lo mismo, solamente que Odoo usa la nomenclatura de módulo en vez de aplicación.

Odoo se basa en una plataforma de desarrollo rápido, de aplicaciones modulares, escalables e intuitivas. Sus módulos se implementan utilizando archivos Python para la

lógica de datos de cada modelo, el lenguaje XML para fabricar las vistas y PostgreSQL como sistema gestor de base de datos. (23).

Python v2.7.8

Es un lenguaje de programación dinámico y orientado a objetos. El principal objetivo que persigue este lenguaje es la facilidad, tanto de lectura, como de diseño. Python es un lenguaje de programación multiparadigma, permite varios estilos: programación orientada a objetos, programación estructural y funcional. Se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation, presenta gran soporte e integración con otros lenguajes y herramientas. Tiene integradas varias bibliotecas estándar. (24).

Ventajas:

- Rápido de desarrollar.
- Sencillez y velocidad.
- Sus bibliotecas hacen gran parte del trabajo.
- Soporta varias bases de datos.

XML v1.6

Es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C) utilizado para almacenar datos en forma legible. Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes. A diferencia de otros lenguajes, XML da soporte a bases de datos, siendo útil cuando varias aplicaciones deben comunicarse entre sí o integrar información. XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

Ventajas:

- Transforma datos en información, le añade un significado concreto y se lo asocia a un contexto con el cual establece flexibilidad para estructurar documentos.
- Permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. (25)

PostgreSQL v9.4.1

En su versión 9.4.1, es uno de los motores de base de datos relacionales más potentes que existe 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. Mejor soporte que los proveedores comerciales: tiene una importante comunidad de profesionales. (26)

Ventajas:

- El código fuente está disponible para todos de manera gratuita.
- Multiplataforma: PostgreSQL está disponible en varias plataformas como son Linux y Windows.

Servidor Apache v2.2

Es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras. Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, además tiene amplia aceptación en la redes. La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones, o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache. Es el componente de servidor web en la popular plataforma de aplicaciones LAMP, junto a MySQL y los lenguajes de programación PHP/Perl/Python. (27).

Ventajas:

- La arquitectura del servidor Apache es muy modular. El servidor consta de una sección core y diversos módulos que aportan varias funcionalidades que podrían considerarse básicas para un servidor web.
- Código abierto.
- Multi-plataforma
- Extensible

PyCharm v4.5.4

Es un IDE o entorno de desarrollo integrado, multiplataforma, utilizado para desarrollar en el lenguaje de programación Python. Proporciona análisis de código, depuración gráfica, integración con VCS / DVCS y soporte para el desarrollo web con Django. Pycharm tiene cientos de funciones que lo puede ver como una herramienta muy pesada, pero que valen la pena, ya que ayuda con el desarrollo del día a día. (28).

Ventajas:

- Autocompletado, resaltador de sintaxis, herramienta de análisis y refactorización.

- Integración con frameworks web como: Django, Flask, Pyramid, Web2Py
- Integración con lenguajes de plantillas: Mako, Jinja2, DjangoTemplate
- Soporta entornos virtuales e intérpretes de Python 2.x, 3.x, PyPy, IronPython y Jython.

Unified Modeling Language (UML) v2.0

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 (29).

Ventajas:

- Es fácilmente entendible.
- Puede registrar diseños parciales independientes de los procesos.
- Permite seleccionar un proceso apropiado para proyectos independientes al lenguaje de modelado.

Notación de Modelado de Procesos de Negocio (BPMN¹)

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. Su principal objetivo es proveer una notación estándar que sea fácilmente legible y entendible por parte de todos los involucrados e interesados del negocio. BPMN sirve de lenguaje común para cerrar la brecha de comunicación que frecuentemente se presenta entre el diseño de los procesos de negocio y su implementación. (30)

Ventajas:

- Soporta diferentes niveles de detalle.

¹**BPMN:** Acrónimo en inglés de Business Process Modeling Notation.

- Brinda a los usuarios finales una notación simple y estándar.
- Proporciona mecanismos para generar procesos ejecutables.
- Proporciona una manera fácil de utilizar la notación de modelado de procesos, accesible a los usuarios y proporciona facilidades para traducir los modelos a una forma ejecutable.

Visual Paradigm v8.0

Es una herramienta multiplataforma, que soporta el ciclo de vida completo del desarrollo de software. Visual Paradigm permite dibujar todos los tipos de diagramas de clases y generar código inverso desde diagramas. Dentro de sus características fundamentales se encuentran que soporta la opción de usar BPMN. Posee como peculiaridad sobre el resto de las demás herramientas que cuenta con una potente funcionalidad para la creación de interfaces de usuarios de las aplicaciones. (31)

Ventajas:

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

1.10. Pruebas de software

Para verificar el correcto funcionamiento de la solución obtenida se hace necesario realizar pruebas al sistema. Un caso de prueba consiste en un conjunto de datos de entrada, condiciones de ejecución y resultados esperados, que son desarrollados para cumplir una función esperada del sistema. (12).

Pruebas definidas por la Metodología AUP-UCI

La metodología AUP-UCI define tres tipos de pruebas: Pruebas Internas, Pruebas de Liberación y Pruebas de Aceptación. (32)

- **Pruebas internas:** En esta prueba se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. El producto de trabajo de prueba desarrollado es el diseño de casos de pruebas.

Los diseños de casos de prueba son producto de trabajo que permite realizar las comprobaciones de los requisitos funcionales de software a partir de escenarios de pruebas. Puede constituir un elemento de seguimiento para la trazabilidad bidireccional de requisitos.

- **Pruebas de liberación:** Pruebas diseñadas y ejecutadas por una entidad certificadora de la calidad externa, a todos los entregables de los proyectos antes de ser entregados al cliente para su aceptación.
- **Pruebas de aceptación:** Si bien es importante comprobar que se desarrolla un producto correctamente, también es esencial corroborar que se desarrolló el producto correcto, para ello es aconsejable aplicar técnicas que permitan medir la satisfacción del cliente.

Métodos de prueba:

- **Pruebas de caja negra:** se centran en verificar el cumplimiento de los requisitos funcionales del software. Permiten encontrar:
 - Funciones incorrectas o ausentes.
 - Errores de interfaz.
 - Errores en estructuras de datos o en accesos a las bases de datos externas.
 - Errores de rendimiento.
 - Errores de inicialización y terminación.
- **Pruebas de caja blanca:** se centran en evaluar la ejecución por lo menos una vez de cada sentencia del programa. Estas pruebas deben garantizar como mínimo que:
 - Se ejerciten por lo menos una vez todos los caminos independientes para cada módulo.
 - Se ejerciten todas las decisiones lógicas en sus vertientes verdaderas y falsas.
 - Se ejerciten las estructuras internas de datos para asegurar su validez. (12)

Una de las técnicas que se aplica en las pruebas de caja blanca, es la del camino básico. Para aplicar esta técnica se debe introducir la notación para la representación del flujo de control, este puede representarse por un grafo de flujo en el cual (33):

- Cada nodo del grafo corresponde a una o más sentencias de código fuente.

- Todo segmento de código de cualquier programa se puede traducir a un grafo de flujo.
- Se calcula la complejidad ciclomática del grafo.

Un grafo de flujo está formado por tres componentes fundamentales que ayudan a su elaboración y comprensión, estos brindan información para confirmar que el trabajo se está haciendo adecuadamente. Componentes del grafo de flujo:

Nodo: Son los círculos representados en el grafo, el cual contiene una o más secuencias del procedimiento, donde un nodo corresponde a una secuencia de procesos o a una sentencia de decisión. Los nodos que no están asociados se utilizan al inicio y final del grafo.

Aristas: Son constituidas por las flechas del grafo, son iguales a las representadas en un diagrama de flujo y constituyen el flujo de control del procedimiento. Las aristas terminan en un nodo, aun cuando el nodo no representa la sentencia de un procedimiento.

Regiones: Son las áreas delimitadas por las aristas y nodos donde se incluye el área exterior del grafo, como una región más. Las regiones se enumeran siendo la cantidad de regiones equivalente a la cantidad de caminos independientes del conjunto básico de un procedimiento.

1.11. Técnica ladov

Para medir la satisfacción del cliente con el producto desarrollado se utiliza la técnica ladov. Esta técnica se compone de cinco preguntas claves: tres cerradas y dos abiertas, las cuales se reformulan en la investigación para valorar el grado de satisfacción de los clientes sobre un tema en específico.

Una vez establecidas las preguntas se conforma el cuadro lógico de ladov (ver Anexo #9) y el número resultante de la interrelación de las tres preguntas, indica la posición de los sujetos en la escala de satisfacción.

La escala de satisfacción está dada por los criterios. (34)

1. Máxima satisfacción.

2. Más satisfecho que insatisfecho.
3. No definida.
4. Más insatisfecho que satisfecho.
5. Máxima insatisfacción.
6. Contradictoria

Para obtener el índice de satisfacción grupal (ISG) se trabaja con los diferentes niveles de satisfacción que se expresan en la escala numérica que oscila entre +1 y - 1 de la siguiente forma (34):

Índice de satisfacción	Escala
Máxima satisfacción	+1
Más satisfecho que insatisfecho	0,5
No definido y contradictorio	0
Más insatisfecho que satisfecho	-0,5
Máxima insatisfacción	-1

Tabla 3. Índice de satisfacción de ladov

La satisfacción grupal (ISG) se calcula por la siguiente fórmula:

Donde:

- A representa el número de sujetos con índice individual 1
- B representa el número de sujetos con índice individual 2
- C representa el número de sujetos con índice individual 3 o 6
- D representa el número de sujetos con índice individual 4
- E representa el número de sujetos con índice individual 5
- N representa el número total de sujetos del grupo

1.12. Conclusiones parciales

Con el estudio de sistemas informáticos que gestionan los procesos relacionados con la distribución y consumo del combustible y el análisis de tecnologías, lenguajes y herramientas de desarrollo, se llegaron a las siguientes conclusiones:

- El análisis de las características y funcionamiento de los Sistemas de Gestión de distribución y consumo de combustible en el ámbito nacional e internacional, permitió comprobar que los sistemas estudiados no satisfacen en su totalidad a las necesidades de la Vicerrectoría Económica, lo que ratificó la necesidad de crear una solución para el problema planteado e incluir y adaptar algunas de las funcionalidades estudiadas de dichos sistemas.
- Para una correcta captura de requisitos se adoptó por utilizar las siguientes técnicas: la entrevista y el estudio de la documentación. En cuanto a las técnicas para validar los requisitos se decidió por la revisión técnica formal y la construcción de prototipos.

- Se determinó el empleo de tecnologías como: servidor web apache v2.2, Metodología de desarrollo AUP con versión para la UCI, PostgreSQL v9.4.1, Odoov8.0, XML v1.6, PyCharm v4.5.4, Python v2.7.8, Visual Paradigm 8.0 y UML v2.0.
- Se estableció aplicar 3 tipos de pruebas: pruebas internas, pruebas de liberación y de aceptación, que son los tipos de pruebas que define la metodología de desarrollo seleccionada. Así como los métodos de prueba caja negra y caja blanca para verificar el correcto funcionamiento de la solución obtenida y para medir el grado de satisfacción del cliente se determinó utilizar la técnica ladov.

CAPÍTULO II: Análisis y Diseño del sistema

2.1. Introducción

En el capítulo se exponen las principales características del sistema desarrollado, mediante el modelado del negocio y la identificación de los requisitos funcionales y no funcionales. Además se describe la arquitectura por la cual se rigió la solución obtenida y los patrones de diseño que fueron utilizados para dar solución al problema que dio origen a la presente investigación.

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. Además se identifican los objetos que forman parte del negocio. Para modelar los procesos de negocio identificados se utilizó la Notación BPMN. (30)

2.2.1. Descripción de los Procesos de Negocios en la UCI.

Planificar Asignación anual y mensual del Combustible

El proceso de planificar la asignación anual y mensual del combustible comienza cuando el especialista de energía consulta los resúmenes de consumo mensual del combustible de años anteriores y a partir de la consulta elabora una propuesta de planificación anual del combustible para el año entrante. Luego envía este plan al vicerrector económico para su aprobación, si no es aprobada se envía al especialista de energía para las correcciones. Una vez aprobada por el vicerrector económico, el especialista de energía envía la propuesta de planificación anual al Ministerio de Educación Superior (MES) para que sea analizada y aprobada. Si el MES aprueba la propuesta de planificación anual, el especialista de energía conforma la planificación

mensual del combustible, que en esta planificación mensual se refleja el plan operativo mensual, la desagregación por meses y la desagregación por tipo de actividad.

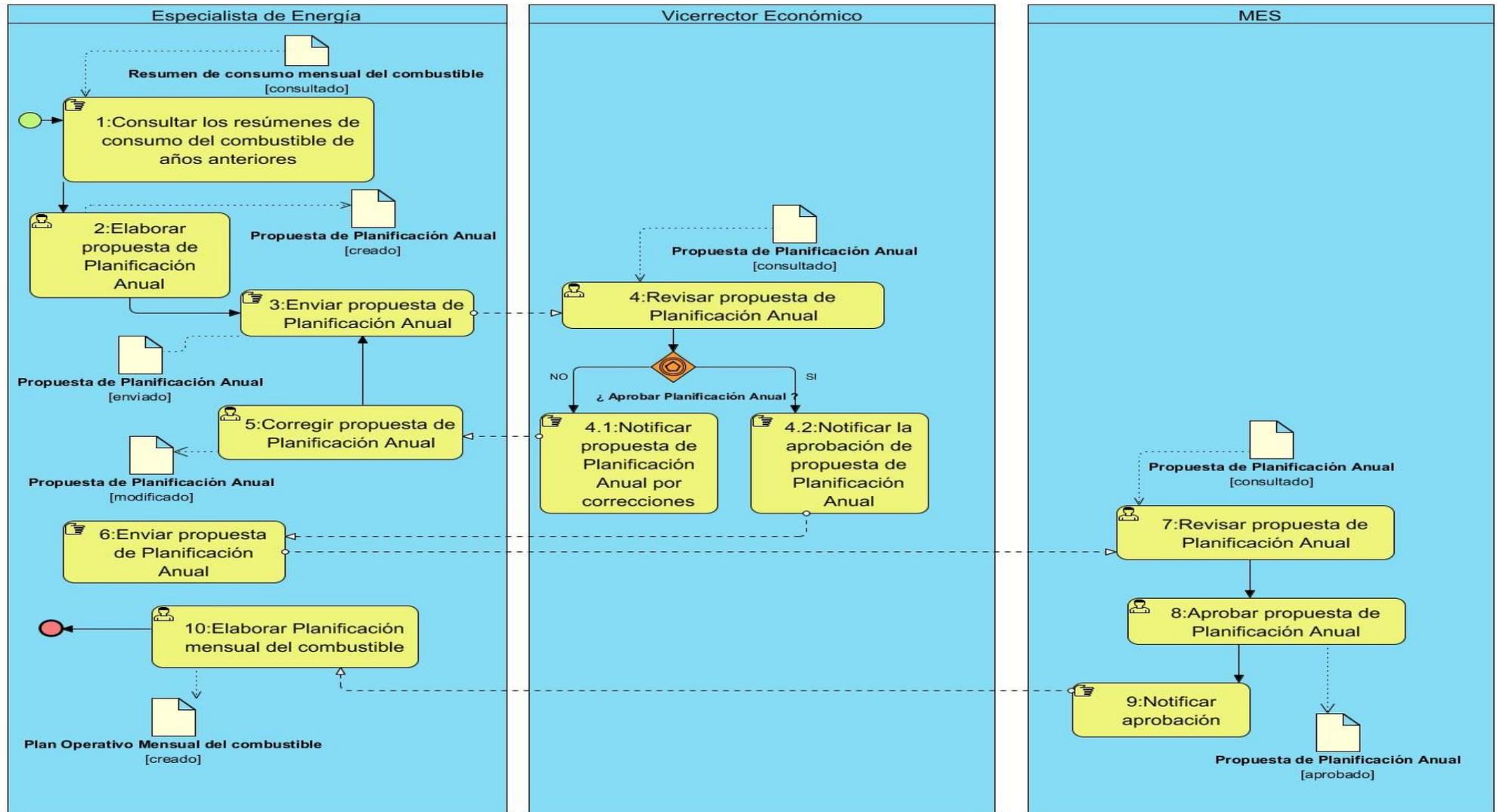


Figura 1. Diagrama de proceso de negocio: Planificar asignación anual y mensual de combustible.

Realizar asignación mensual de combustible

El proceso de realizar 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 para que la revise. El vicerrector la revisa y si es necesario hace modificaciones y luego este último se le envía a la Rectora para que la apruebe. La Rectora revisa y si se encuentra bien la aprueba, o la envía para la Dirección de transporte por correcciones. 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 terminando así este proceso. Para ver el Diagrama de proceso de negocio: Realizar asignación mensual de combustible, ir al Anexo #4.

Aprobar entrega de combustible adicional

El proceso de aprobar la entrega del combustible adicional comienza cuando el responsable del vehículo elabora y entrega la solicitud del combustible adicional al vicerrector asociado a su área para que este se lo solicite al vicerrector económico. El vicerrector económico puede no aprobar esta solicitud y así terminaría este proceso, pero si aprueba la solicitud, le entrega el modelo de autorizo al vicerrector asociado al área y este a su vez se lo hace llegar al responsable del vehículo, terminando así el proceso. Para ver el Diagrama de proceso de negocio: Aprobar entrega de combustible adicional, ir al Anexo #5.

Recogida y entrega de tarjetas magnéticas

Recogida de tarjetas magnéticas: La especialista de la caja se encarga de hacer una consulta de la asignación mensual de combustible a partir de los registros de consumo que posee en caja por cada tarjeta y los que posee la Dirección de transporte a partir de las liquidaciones mensuales de combustible por cada vehículo. Además, realiza una consulta en el submayor de consumo de combustible que poseen en los registros de la caja. Luego de hacer esto, se encargan de hacer la selección de las tarjetas por los tipos de combustible, los cuales pueden ser, diésel, gasolina especial y gasolina regular. Se elabora otro documento por parte de la caja, donde la cajera hace la

distribución de combustible por cantidades que posee cada tarjeta, este documento se hace entrega por parte de la cajera al grupo de finanzas, con el fin de revisar que la asignación que hizo la Dirección de transporte y que fue aprobada por varios factores, sea distribuida a partir de la existencia que posee cada tarjeta. Finanzas realiza la carga a cada tarjeta con el objetivo de ponerle lo asignado por la Dirección de transporte. Posteriormente finanzas revisa la carga realizada por un especialista de este grupo de trabajo, la cajera se encarga de recoger las tarjetas en soporte físico y de hacer entrega al responsable de cada vehículo del centro que le fue asignado combustible.

Entrega de tarjetas magnéticas: La caja de la UCI, a través de un especialista se encarga de hacer entrega a inicios de cada mes de la tarjeta que tiene asignada cada vehículo, para esto el chofer del vehículo debe presentarse en la caja y firmar varios documentos que la cajera elabora y se procede a entregar la tarjeta con el pin que posee, el pin no es más que el código de acceso al uso de la tarjeta. Para ver el Diagrama de proceso de negocio: Recogida y entrega de tarjetas magnéticas, ir al Anexo #6.

Cargar combustible en tarjetas magnéticas

Este proceso se lleva a cabo cuando el grupo de finanzas recibe del área de caja la tabla de cargue de tarjetas. Esta tabla posee un desglose por área de cada vehículo que tiene asignado y a su vez la cantidad de combustible que se le debe poner al vehículo en la tarjeta que tenga asignado para comenzar a ser usada. Luego de tenerse la tabla de carga, un especialista de este grupo se encarga de hacer las cargas asignadas a cada tarjeta magnética. Después de cargada cada tarjeta, se notifica a la caja de la UCI mediante un documento la carga realizada a cada tarjeta magnética, además de hacerle entrega a otro especialista de la caja de las tarjetas magnéticas en soporte duro y los ticket de notificación de carga, los ticket, no son más que el reporte que muestra el sistema de recarga de combustible de la empresa Financiera CIMEX S.A, los cuales notifican la carga recibida por cada tarjeta en valor contable y valor en consumo en litros. Para ver el Diagrama de proceso de negocio: Cargar combustible en tarjetas magnéticas, ir al Anexo #7.

Liquidar combustible

Este proceso se lleva a cabo tres veces en el mes. Cada responsable del vehículo debe ir el primer viernes del mes a entregar en la caja los ticket de consumo que haya llevado a cabo, luego lo hace el tercer viernes del mes y luego cualquier día de la última semana del mes, ya esta última vez, entrega la tarjeta para ser resguardada por la caja. Estas entregas de ticket permiten que la cajera pueda tener conocimiento del consumo que se ha llevado a cabo por cada tarjeta asignada a un vehículo, pues se debe tener el conocimiento de los consumos que se hacen, se debe tener registrado, donde se habilitó el vehículo, que cantidad se habilitó y que día lo hizo. De esto se debe hacer un reporte el cual se debe informar al grupo de finanzas para poder hacer el próximo cargue del mes entrante. Para ver el Diagrama de proceso de negocio: Liquidar combustible, ir al Anexo #8.

2.3. Modelo Conceptual

El modelo conceptual explica de forma global los aspectos lógicos significativos en el dominio del problema. Además estipula una especificación de los requisitos desde la perspectiva de la clasificación por objetos y desde el punto de vista de entender los términos empleados en el dominio. (35) A continuación se muestra la representación conceptual en el proceso de distribución y consumo de combustible en la UCI

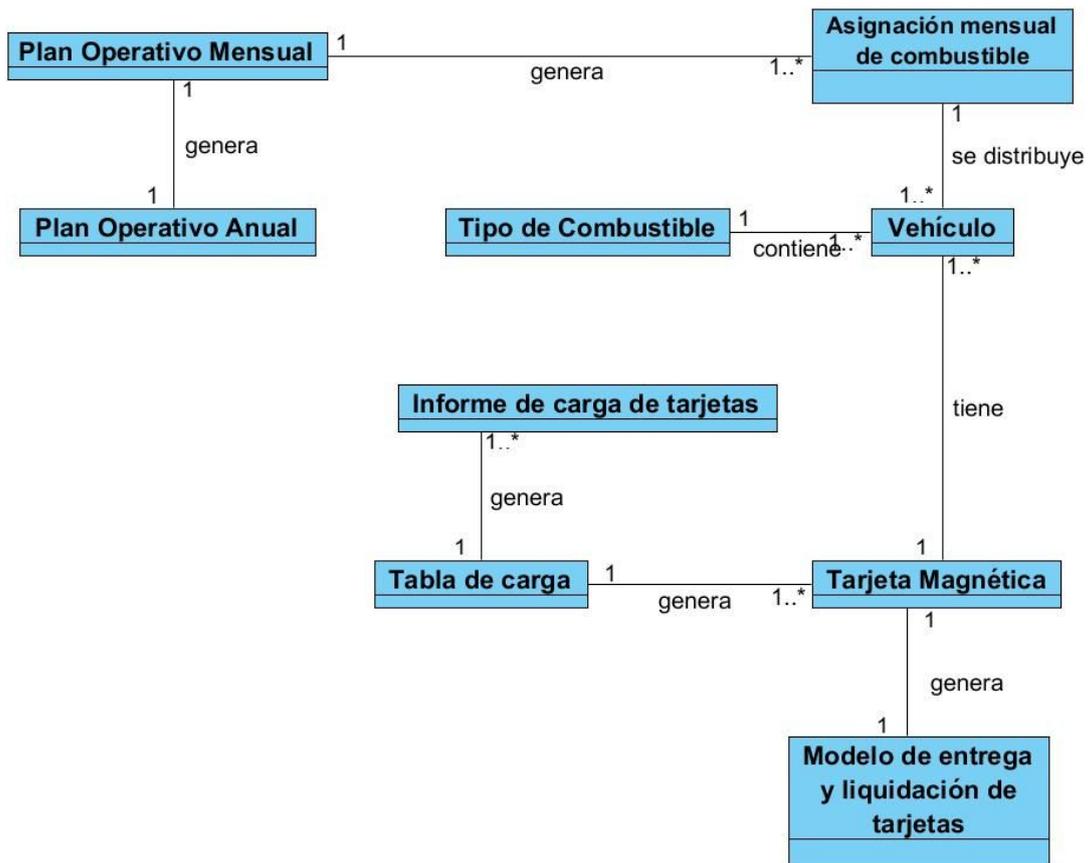


Figura 2. Modelo conceptual del proceso de distribución y consumo del combustible en la UCI.

2.4. Requisitos del software

Una vez aplicada la técnica de la entrevista para la captura de requisitos, se obtuvo un total de 124 requisitos funcionales, los cuales se encuentran agrupados por Áreas. A continuación se muestra un listado de los requisitos funcionales.

2.4.1. Listado de los Requisitos Funcionales

NO. RF	DESCRIPCIÓN DEL REQUISITO	PRIORIDAD
Grupo energético		
RF 1	Adicionar planificación anual de asignación del combustible	Alta
RF 2	Modificar planificación anual de asignación del combustible	Alta
RF 3	Mostrar planificación anual de asignación del combustible	Alta
RF 4	Eliminar planificación anual de asignación del combustible	Media
RF 5	Buscar planificación anual de asignación del combustible	Media
RF 6	Adicionar desagregación por meses del plan	Alta
RF 7	Modificar desagregación por meses del plan	Alta
RF 8	Mostrar desagregación por meses del plan	Alta
RF 9	Eliminar desagregación por meses del plan	Media
RF 10	Buscar desagregación por meses del plan	Media
RF 11	Adicionar desagregación por tipo de actividad	Alta
RF12	Modificar desagregación por tipo de actividad	Alta
RF13	Mostrar desagregación por tipo de actividad	Alta
RF14	Eliminar desagregación por tipo de actividad	Media
RF15	Buscar desagregación por tipo de actividad	Media
Vicerrectoría económica		
RF 16	Adicionar solicitud del combustible adicional	Alta
RF 17	Modificar solicitud del combustible adicional	Alta
RF 18	Mostrar solicitud del combustible adicional	Alta
RF 19	Eliminar solicitud del combustible adicional	Media
RF 20	Buscar solicitud del combustible adicional	Media
RF 21	Adicionar modelo de entrega de tarjetas magnéticas	Alta
RF 22	Modificar modelo de entrega de tarjetas magnéticas	Alta
RF 23	Mostrar modelo de entrega de tarjetas magnéticas	Alta
RF 24	Eliminar modelo de entrega de tarjetas magnéticas	Media
RF 25	Buscar modelo de entrega de tarjetas magnéticas	Media
RF 26	Aprobar solicitud del combustible adicional	Media
RF 27	Aprobar asignación mensual del combustible	Media
RF 28	Aprobar asignación anual del combustible	Media
Transporte		
RF 29	Adicionar cierre del combustible	Alta
RF 30	Modificar cierre del combustible	Alta

RF 31	Mostrar cierre del combustible	Alta
RF 32	Eliminar cierre del combustible	Media
RF 33	Buscar cierre del combustible	Media
RF 34	Adicionar vehículos al parque de equipos	Alta
RF 35	Modificar vehículos al parque de equipos	Alta
RF 36	Mostrar vehículos al parque de equipos	Alta
RF 37	Eliminar vehículos al parque de equipos	Media
RF 38	Buscar vehículos al parque de equipos	Media
RF 39	Adicionar asignación mensual del combustible	Alta
RF 40	Modificar asignación mensual del combustible	Alta
RF 41	Eliminar asignación mensual del combustible	Media
RF 42	Buscar asignación mensual del combustible	Media
RF 43	Adicionar plan base	Alta
RF 44	Modificar plan base	Alta
RF 45	Eliminar plan base	Media
RF 46	Buscar plan base	Media
RF 47	Adicionar recorridos diarios	Alta
RF 48	Modificar recorridos diarios	Alta
RF 49	Eliminar recorridos diarios	Media
RF 50	Buscar recorridos diarios	Media
RF 51	Adicionar recorridos de alimentos, camiones y extraplanes	Alta
RF 52	Modificar recorridos de alimentos, camiones y extraplanes	Alta
RF 53	Eliminar recorridos de alimentos, camiones y extraplanes	Media
RF 54	Buscar recorridos de alimentos, camiones y extraplanes	Media
RF 55	Adicionar trompos	Alta
RF 56	Modificar trompos	Alta
RF 57	Eliminar trompos	Media
RF 58	Buscar trompos	Media
Caja		
RF 59	Adicionar entrega de combustible por tarjeta.	Alta
RF 60	Modificar entrega de combustible por tarjeta.	Alta
RF 61	Eliminar entrega de combustible por tarjeta.	Alta
RF 62	Mostar entrega de combustible por tarjeta.	Alta
RF 63	Adicionar liquidación de combustible por tarjeta.	Alta
RF 64	Modificar liquidación de combustible por tarjeta.	Alta
RF 65	Eliminar liquidación de combustible por tarjeta.	Alta
RF 66	Mostrar liquidación de combustible por tarjeta.	Alta
RF 67	Buscar entrega por tarjeta.	Media

RF 68	Buscar liquidación por tarjeta.	Media
RF 69	Crear modelo de entrega y liquidación.	Alta
RF 70	Exportar el modelo de entrega y liquidación a pdf.	Alta
Contabilidad y Finanzas		
RF 71	Adicionar carga de combustibles en dinero y en litros.	Alta
RF 72	Modificar carga de combustibles en dinero y en litros.	Alta
RF 73	Eliminar carga de combustibles en dinero y en litros.	Alta
RF 74	Mostrar carga de combustible en dinero y en litros.	Alta
RF 75	Adicionar tarjeta.	Alta
RF 76	Modificar tarjeta.	Alta
RF 77	Eliminar tarjeta.	Alta
RF 78	Mostrar tarjeta.	Alta
RF 79	Buscar tarjeta.	Media
RF 80	Buscar carga por tarjeta.	Media
RF 81	Buscar Carga por Vehículo.	Media
RF 82	Realizar reporte de carga de combustible por mes.	Alta
RF 83	Realizar reporte de carga de combustible por tipo de combustible.	Alta
RF 84	Realizar reporte de carga de combustible por vehículo.	Alta
RF 85	Exportar reporte de carga de combustible por mes y por tipo de combustible a pdf.	Alta
Generales		
RF 86	Autenticar Usuario	Media
RF 87	Visualizar Ayuda	Baja
RF 88	Visualizar Descripción del sistema	Baja
Configuración		
RF89	Adicionar Usuario del sistema	Baja
RF 90	Modificar Usuario del sistema	Baja
RF 91	Mostrar Usuario del sistema	Baja
RF 92	Eliminar Usuario del sistema	Baja
RF 93	Adicionar estructura	Baja
RF 94	Modificar estructura	Baja
RF 95	Mostrar estructura	Baja
RF 96	Eliminar estructura	Baja
RF 97	Adicionar dependencia	Baja
RF 98	Modificar dependencia	Baja
RF 99	Mostrar dependencia	Baja
RF 100	Eliminar dependencia	Baja

RF 101	Adicionar tipo de combustible	Baja
RF 102	Modificar tipo de combustible	Baja
RF 103	Mostrar tipo de combustible	Baja
RF 104	Eliminar tipo de combustible	Baja
RF 105	Adicionar período	Baja
RF 106	Modificar período	Baja
RF 107	Mostrar período	Baja
RF 108	Eliminar período	Baja
RF 109	Adicionar recorridos	Baja
RF 110	Modificar recorridos	Baja
RF 111	Mostrar recorridos	Baja
RF 112	Eliminar recorridos	Baja
RF 113	Adicionar actividades	Baja
RF 114	Modificar actividades	Baja
RF 115	Mostrar actividades	Baja
RF 116	Eliminar actividades	Baja
RF 117	Adicionar tipo de vehículo	Baja
RF 118	Modificar tipo de vehículo	Baja
RF 119	Mostrar tipo de vehículo	Baja
RF 120	Eliminar tipo de vehículo	Baja
RF 121	Adicionar usuario del negocio	Baja
RF 122	Modificar usuario del negocio	Baja
RF 123	Mostrar usuario del negocio	Baja
RF 124	Eliminar usuario del negocio	Baja

Tabla 4. Listado de requisitos funcionales

2.4.2. Evaluación de requisitos.

Luego de haber identificado los requisitos funcionales del sistema, se realizó la evaluación de requisitos siguiendo los criterios de complejidad que establece la plantilla “Evaluación de requisitos” definida por el centro CEIGE. (36)

La clasificación de la complejidad en Alta, Media y Baja permite estimar el esfuerzo de implementación del requisito, estableciendo como requisitos de mayor prioridad aquellos que posean complejidad Alta.

La Tabla muestra los resultados obtenidos de la evaluación de requisitos de acuerdo a los criterios de complejidad a los que se hacían referencia anteriormente.

Resumen	Cantidad
Cantidad de requisitos con complejidad Alta	53
Cantidad de requisitos con complejidad Media	33
Cantidad de requisitos con complejidad Baja	38

Tabla 5.Resultados de la Complejidad

2.4.3.Descripción de Requisitos

A continuación se muestra la descripción del requisito Adicionar Desagregación por meses del plan.

Precondiciones	Se ha registrado la planificación anual del combustible
Flujo de eventos	
Flujo básico <<Adicionar Desagregación por meses del plan>>	
1.	El usuario selecciona la opción Planificación Mensual y después selecciona Desagregación por meses que se encuentra en el menú izquierdo del sistema
1.	Selecciona el botón Crear.
2.	El sistema presenta una nueva ventana donde el usuario introduce los datos: Planificación anual, Actividad, Consumo real y Nivel de actividad para cada mes
3.	El usuario selecciona la opción Guardar
4.	El sistema valida los datos introducidos.
5.	Si los datos son correctos el sistema confirma el registro de los datos.
6.	El sistema presenta los datos introducidos por el usuario.
7.	Concluye el requisito.
Pos-condiciones	
1.	Se muestra un listado con todas las desagregaciones por meses
Flujos alternativos	
Flujo alternativo 4.a Información Errónea	
1	El sistema muestra una alerta indicando que existen campos inválidos
1	El usuario corrige los datos.
2	Volver al paso 4 del flujo básico.
Pos-condiciones	
1	N/A
Flujo alternativo 4.b Información incompleta	
1	El sistema muestra una alerta indicando que existen campos inválidos
2	El usuario inserta los datos.

3	Volver al paso 4 del flujo básico.	
Pos-condiciones		
4	N/A	
Flujo alternativo *.a El usuario cancela la acción		
1	El usuario selecciona la opción descartar	
2	Concluye el requisito	
Pos-condiciones		
2	N/A	
Validaciones		
1	Se validan los datos según lo establecido en los diseños de caso de pruebas	
Conceptos	La desagregación por meses del plan consiste en realizar un desglose del consumo del combustible planificado para ese año	<p>Visibles en la interfaz: planificación anual, actividad, consumo real total, consumo planificado, nivel de actividad, consumo real e índice de consumo para cada mes</p> <p>Utilizados internamente: name, pom_id, pac_id, actividad_id, m01_nivel, m01_cons, m01_índice, m02_nivel, m02_cons, m02_indice, m03_nivel, m03_cons, m03_indice, m04_nivel, m04_cons, m04_indice, m05_nivel, m05_cons, m05_indice, m06_nivel, m06_cons, m06_indice, m07_nivel, m07_cons, m07_indice, m08_nivel, m08_cons, m08_indice, m09_nivel, m09_cons, m09_indice, m10_nivel, m10_cons, m10_indice, m11_nivel, m11_cons, m11_indice, m12_nivel, m12_cons, m12_indice, total_consumo, consumo_planificado</p>

Tabla 6. Descripción de requisito: Adicionar Desagregación por meses del plan.

2.4.4. Técnica para Validar los requisitos funcionales

Para la revisión técnica formal, se reunió el cliente con el equipo de desarrollo para verificar el documento de requisitos en cuanto a anomalías, omisiones, requisitos inalcanzables o contradictorios, con esta técnica se llegó a la conclusión que los requisitos están correctamente definidos y puntualiza realmente lo que el cliente necesita.

La construcción de prototipos resultó ser muy conveniente para los usuarios, se les mostró las futuras interfaces del sistema, permitiendo al cliente hacerse una idea de cómo sería el producto final una vez terminado, quedando el cliente totalmente complacido. En la Figura 3 y 4 se muestra el prototipo de interfaz de usuario del requisito Adicionar Desagregación por meses del plan.

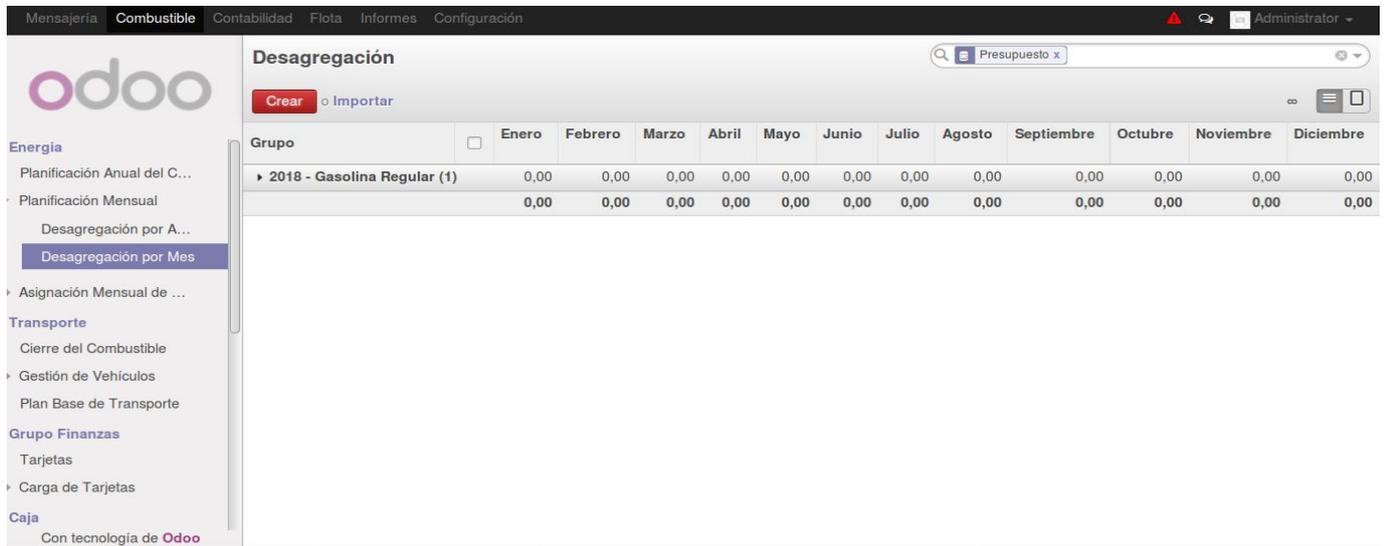


Figura 3. Prototipo de interfaz de usuario del requisito: Adicionar Desagregación por meses del plan.



Figura 4. Prototipo de interfaz de usuario del requisito: Adicionar Desagregación por meses del plan.

2.4.5. Listado de 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). Los RNF son aquellas características que el sistema debe cumplir para garantizar el despliegue y soporte del mismo. (12)

RNF 1	<p>Usabilidad</p> <ul style="list-style-type: none"> — El sistema debe poseer una distribución de la información 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. — El sistema debe permitirle a los usuarios con pocos conocimientos de informática poder interactuar con él. — El sistema debe poseer una interfaz intuitiva y fácil de navegar.
RNF 2	<p>Confiabilidad</p> <ul style="list-style-type: none"> — El sistema debe ser preciso con la información que maneja y le proporciona al usuario. Haciendo énfasis en los resultados de los análisis que ejecutará, para evitar errores que puedan incidir negativamente en la toma de decisiones. — La información no podrá ser modificada por ningún usuario no autorizado, protegiendo así la integridad de los datos.
RNF 3	<p>Seguridad</p> <ul style="list-style-type: none"> — El sistema le permitirá al usuario las funcionalidades en dependencia del rol que cumpla — 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. — Todo uso de las funcionalidades del sistema requiere la autenticación de los usuarios. — El sistema concederá acceso a cada usuario autenticado solo a las funciones que le estén permitidas, de acuerdo a la configuración del sistema. — El sistema maneja la seguridad de acceso y administración de usuarios mediante el otorgamiento de privilegios y roles.
RNF 4	<p>Disponibilidad</p> <ul style="list-style-type: none"> — Crear salvadas de seguridad en caso de fallo eléctrico — El sistema debe estar disponible las 24 horas del día los 7 días de la semana.
RNF 5	<p>Soporte</p> <ul style="list-style-type: none"> — El sistema debe dar la posibilidad de ser mejorado, así como de incorporarle nuevas funcionalidades, en caso de ser necesarias.

RNF 6	Software (Cliente) <ul style="list-style-type: none"> — Las PC clientes deben tener instalado un Navegador Web Mozilla Firefox o Google Chrome.
RNF 7	Software (Servidor) <ul style="list-style-type: none"> — Para un servidor de aplicación Web se requiere usar Apache versión 2.0 y un servidor de base de datos PostgreSQL 9.1 o superior.
RNF 8	Hardware (Cliente) <ul style="list-style-type: none"> — Las PC clientes deben poseer requisitos Mínimos como: Procesador 1ghz, con 512 de RAM. — Cada PC cliente debe tener Conexión de red.
RNF 9	Hardware (Servidor) <ul style="list-style-type: none"> — Se debe disponer de un servidor con requisitos Mínimos como: Procesador Intel Pentium Core-2-Duo, con 2GB de RAM. — Se debe disponer de un Disco duro de 160 GB o superior. — El servidor debe tener Conexión de red.
RNF 10	Portabilidad <ul style="list-style-type: none"> — El módulo permitirá la manipulación de dispositivos externos como las impresoras. — El sistema será multiplataforma, con el correcto funcionamiento en Linux y Windows.

Tabla 7.Listado de requisitos no funcionales.

2.5. Diseño de la solución

El diseño de un sistema se basa en su arquitectura, la cual está compuesta por varios estilos o patrones que posibilitan detallar un software con suficiente claridad, lo que facilita llevar a cabo la realización física del mismo. La elaboración de un correcto modelo de diseño influye de manera significativa en la realización de un software con alta calidad.

En el presente epígrafe se especifica la arquitectura y los patrones del diseño aplicados a la solución, el diagrama de clases y el modelo de datos obtenido.

2.5.1.Arquitectura de software del sistema

La arquitectura de software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.

Es la estructura del sistema, lo que comprende a los componentes del software, sus propiedades externas visibles y las relaciones entre ellas. Esta tiene gran importancia debido a que durante el desarrollo del sistema provee una excelente vista general del mismo, proporciona la relación de puntos de diseño a tratar, facilita el desarrollo simultáneo de componentes y permite detectar errores de diseño en fases tempranas. (37)

El sistema posee una arquitectura Cliente-Servidor, ya que está desarrollada sobre la plataforma Odoo. Como se muestra en la figura # 5, Odoo consiste en tres componentes básicos:

- Un servidor con un SGBD PostgreSQL que contiene todos los datos: datos de cada modelo de cada módulo instalado, el código de las vistas y datos de configuración del propio Odoo, etc.
- Un servidor que puede funcionar local o en cualquier proveedor de hosting online. Este componente es el que contiene el núcleo de Odoo. El núcleo de Odoo está programado en Python y es lo que hace que todo funcione. Una capa de este servidor está destinada a comunicarse con el SGBD PostgreSQL a través del motor del ORM que Odoo incorpora y la otra capa está destinada a comunicarse con el cliente web Odoo que se ejecuta en el navegador del usuario.
- El tercer componente consiste en una completa aplicación JavaScript que se ejecuta en el navegador del usuario para que el usuario pueda usar Odoo sin instalar ninguna aplicación.

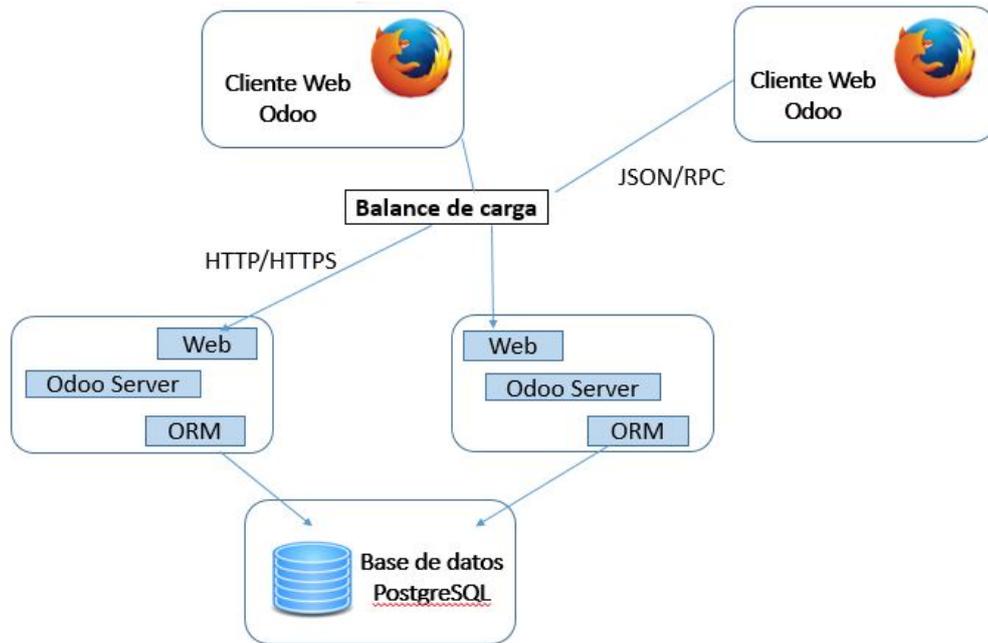


Figura 5. Funcionamiento Interno

Framework de desarrollo en Odoo, es el framework que utiliza el sistema, este proporciona todas las herramientas para construir la lógica de tratamiento de datos y su interfaz gráfica. Permite un desarrollo rápido de funcionalidades o conectividad con otras plataformas y su extensa documentación, facilita la integración y mejora del programa de manera muy ágil. (23)

2.5.2. Patrones de diseño y arquitectónico

El diseño de la solución fue elaborado siguiendo patrones que constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos.

Patrón arquitectónico

El sistema está construido de acuerdo al patrón MVC, sus modelos, vistas y controladores deben ser almacenados en sus respectivas carpetas. Todos los archivos estáticos y recursos web deben ser almacenados en las carpetas llamadas estáticas, en las que habrá carpetas específicas para cada tipo de recurso. Otros tipos de recursos también tienen sus propias carpetas, como los reportes y traducciones.

El modelo se define con objetos Python, donde se introducen los datos que se almacenan en la base de datos PostgreSQL. El mapeo de la base de datos se gestiona automáticamente por Odo, y el mecanismo responsable de esto es el modelo relacional de objetos (ORM).

Las vistas describen la interfaz de usuario. Estas vistas se definen mediante XML, que es usado por el framework cliente web para generar vistas HTML de reconocimiento de datos.

En cuanto al controlador son clases internas de Python que se generan a partir de los modelos que se definen.

A continuación se muestra la **iError! No se encuentra el origen de la referencia.6**, donde se ve relegado el patrón MVC.

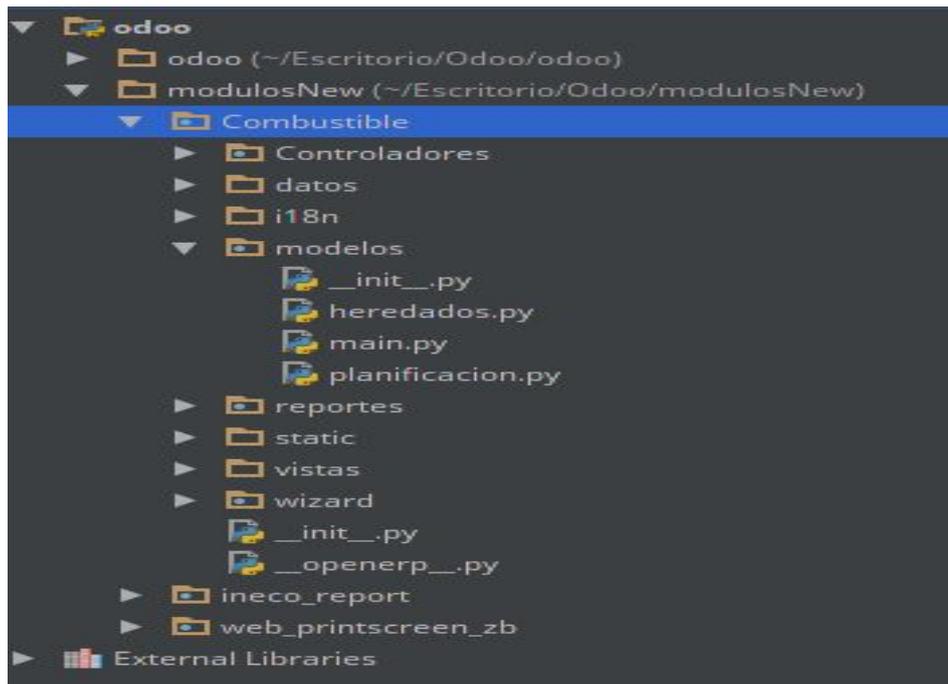


Figura 6. Estructura de carpetas.

Controladores: MVC controladores, incluidos para la web, que serán añadidos.

datos: Archivos con los datos necesarios para el módulo.

i18n: Traducción de archivos

modelos: MVC modelos que se añadirán a la ERP.

reportes: Las definiciones de informe, sus vista y códigos fuente.

static: Los archivos estáticos usados en el módulo.

Vistas: MVC vistas (formulario, listas), menús y acciones.

Wizard: Asistentes a las sesiones interactivas con el usuario.

Patrones de diseño GRASP

Para evitar la sobrecarga de las clases y contribuir a que el sistema sea más robusto y flexible se utilizaron los siguientes patrones GRASP:

- **Experto:** La utilización de este patrón se evidencia en las clases del modelo. Por ejemplo, la responsabilidad de crear tarjetas magnéticas se le asigna a la clase Tarjetas, la cual contiene toda la información necesaria para realizar dicha actividad.
- **Creador:** Se evidencia en la clase PlanBase ya que es el responsable de la creación de nuevos objetos o clases, como la creación de las clases BaseCamión, BaseExtraplan, BaseRecorrido, BaseTrompo y BaseAlimento, donde la nueva

clase deberá ser creada por la clase que tiene toda la información necesaria para realizar la acción.

- **Controlador:** La clase Controller, es ejemplo de la aplicación de este patrón, la misma tiene a cargo la responsabilidad de manejar la lógica de negocio.
- **Bajo acoplamiento:** La utilización de este patrón se evidencia en la clase Áreas, donde solo existen las relaciones necesarias entre las clases para mantener bajo el acoplamiento.
- **Alta cohesión:** Su empleo se evidencia en la clase AsignaciónMensual, tiene como responsabilidad fundamental realizar las labores que solo le competen a él y que no son desempeñadas por otros elementos del diseño.

Patrones Banda de los Cuatro (GoF por sus siglas en inglés): Adaptados para solucionar un problema común en particular mediante la descripción de las clases y la comunicación entre objetos; se agrupan en tres categorías: creacionales, estructurales y de comportamiento.

— Creacional

Singleton: La utilización de este patrón se evidencia en la clase Test02Mserialization, mediante la creación del método test_singleton_commands el cual asegura que solo se cree un único objeto de cada clase.

```
def test_singleton_commands(self):  
    results = self.partner.resolve_2many_commands(  
        self.cr, UID, 'child_ids', [DELETE_ALL()], ['name'])  
    self.assertEqual(results, [])
```

Figura 7. Código donde se evidencia patrón creacional.

— Comportamiento

Observer: La utilización de este patrón se evidencia en la clase Pac (plan anual del combustible), donde el plan anual del combustible presenta una dependencia de uno a mucho con la clase Elementos.

```

class Pac(models.Model):
    _name = 'combustible.pac'
    _description = 'Aspecto clave en la planificación del combustible'

    ano = fields.Many2one('account.fiscalyear', 'Año a Planificar', select=True, required=True)
    name = fields.Char(string='Código', size=5, required=True)
    cent = fields.Char(string='Centro', size=6, required=True, default='UCI')
    real_anno = fields.Integer('Real Año', compute='_dependecia_anno', store=True)
    est_anno = fields.Integer('Estimado', compute='_dependecia_anno', store=True)
    dem_anno = fields.Integer('Demanda', compute='_dependecia_anno', store=True)
    acum_anno = fields.Many2one('account.period', 'Acumulado hasta', select=True, required=True)

    tipocombustible_id = fields.Many2one('combustible.tipocombustible', 'Tipo Combustible', required=True)
    elemento_anual_ids = fields.One2many('combustible.elemento.anual', 'pac_id', 'Elementos')

```

Figura 8.Código donde se evidencia patrón comportamiento.

⤴ Estructurales

Decorator: La utilización de este patrón se evidencia en la clase cierre.

```

@api.depends('nivel', 'cons_plan', 'cons_total')
def _dependecia_calc(self):

```

Figura 9.Código donde se evidencia patrón decorator.

2.5.3.Diagrama de clases

Para un mayor entendimiento de cómo está estructurado el sistema se muestra el siguiente diagrama de clases, que es un tipo de diagrama de estructura estática ,que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos. (38)

2.5.4. Modelo de datos

El modelo de datos describe de forma abstracta las entidades y sus características, además de las relaciones entre estas dentro de la base de datos. Las entidades son objetos que guardan información necesaria para el sistema y los atributos son las características de una entidad. Las relaciones muestran la asociación entre dos entidades, representadas por una línea discontinua que une a las entidades involucradas. (39)

A continuación se muestra el modelo de datos del sistema:

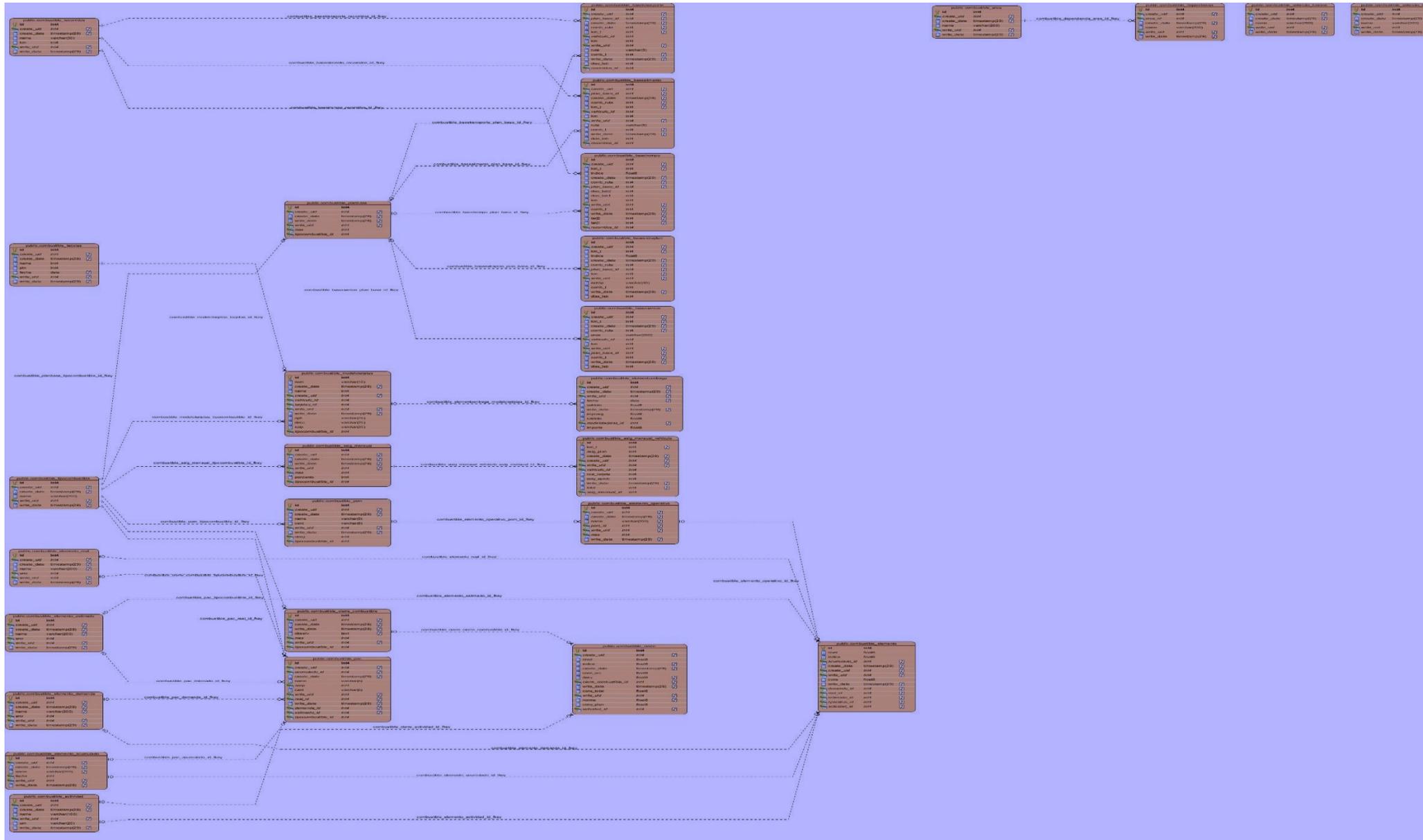


Figura 11. Diagrama de Entidad relación

2.5.5. Mapeo de objetos a Base de datos (ORM)

Odoo proporciona su propio ORM que permite manejar los datos de la base de datos con una API (Application Programming Interface) propia. Esta API muestra cómo hacer las operaciones sobre la base de datos, así como la manera de definir las columnas de cada tabla de la base de datos usando su propia sintaxis. Se utiliza esta API con el lenguaje de programación Python. El ORM de Odoo permitirá que se definan como atributos de sus clases, las columnas que formarán parte de la tabla correspondiente a ésta, así como las relaciones mantenidas con otras tablas de la base de datos. (23)

A continuación se describen algunos conceptos técnicos del ORM de Odoo.

Un modelo no es más que una clase programada en Python, que corresponde a una tabla de la base de datos, evidenciándose de la siguiente forma:

```
class Actividad(models.Model):
    _name = 'combustible.actividad'
    _description = 'Actividades del Centro'

    name = fields.Char(string='Nombre', size=50, required=True)
    code = fields.Char(string='Código', size=6, required=True)
    um = fields.Char(string='U.M N.Act', size=20, required=True)
```

Figura 12. Fragmento de la clase Actividad del Sistema de Gestión para la distribución y consumo del combustible en la UCI.

En el código anterior se definió la clase llamada “Actividad” que hereda de la clase `models.Model` para que sea mapeada en la base de datos en forma de tabla con el nombre “`combustible.actividad`”. El ORM de Odoo permite definir como atributos (formalmente llamados campos en Odoo) de esta clase, a las columnas que formarán parte de la tabla correspondiente, así como las relaciones mantenidas con otros modelos.

2.5.6. Validación del diseño

Con el fin de evaluar la calidad del diseño se utilizaron las métricas de software siguientes que permiten medir de forma cuantitativa la calidad de los atributos internos del sistema propuesto:

Aplicación de la métrica Tamaño Operacional de Clases (TOC)

Se le aplicó la métrica Tamaño Operacional de Clases a un total de 34 clases y 174 operaciones, para un promedio de procedimientos por clases de 5.11, aproximándolo al valor entero 5.

La mayoría de las clases tienen de 0 a 5 procedimientos, lo que demuestra que el funcionamiento general del sistema está distribuido equitativamente entre las diferentes clases. (Ver Anexo #1)

Figura 13. Representación en % de los resultados de la evaluación mediante la métrica TOC en los atributos: Reutilización, Complejidad de implementación y Responsabilidad.

Luego de la aplicación de la métrica los resultados obtenidos fueron satisfactorios pues más del 80% de las clases presentó baja responsabilidad y complejidad de implementación y solo el 5% presentó bajo grado de reutilización de un total de 34 clases.

Aplicación de la métrica Relaciones entre Clases (RC)

Se le aplicó la métrica Relaciones entre Clases a un total de 34 clases y 87 relaciones, para un promedio de relaciones entre clases de 2.55, aproximándolo al valor entero 3. La mayoría de las clases cuentan con dos (2) relaciones entre clases, por lo que las clases se encuentran dentro de los niveles aceptables e calidad. (Ver Anexo #2)

Figura 14. Representación en % de los resultados de la evaluación mediante la métrica RC en los atributos: Acoplamiento, Complejidad de mantenimiento, Cantidad de prueba y Reutilización.

Al analizar los resultados obtenidos de la métrica RC, se puede concluir que el diseño realizado tiene una calidad aceptable. Los atributos de calidad se encuentran en un nivel satisfactorio; en el 59% de las clases el nivel de Acoplamiento es mínimo. La Complejidad de mantenimiento y la Cantidad de pruebas son bajas a un 82%, mientras que el atributo Reutilización cuenta con igual por ciento en la categoría Alta. Esto implica que se reduzca el nivel de esfuerzo necesario para sustentar el diseño y se disminuyan la cantidad de pruebas realizadas para mejorar la calidad de software.

2.6. Conclusiones parciales

En el presente capítulo se describieron las características del sistema a partir de un modelado del negocio previo, que permitió conocer el proceso de distribución y consumo del combustible en la UCI, para ello se identificaron requisitos funcionales y no funcionales del sistema los cuales posibilitaron una mejor visión de la implementación y funcionamiento de la solución. También se evidenciaron los siguientes resultados:

- A partir de la aplicación de las técnicas para la validación de los requisitos funcionales se logró probar que los requisitos identificados se encuentran descritos de forma correcta, clara y consistente, además que se corresponden con las expectativas del cliente.
- Por otra parte, se realizó el modelado del diseño el cual fue verificado mediante la aplicación de las métricas TOC y RC, donde se evidenció que el diseño presenta

baja responsabilidad y complejidad, lo cual favorece la reutilización, la alta cohesión y el bajo acoplamiento, pudiéndose comprobar una correcta asignación de responsabilidades.

- El uso de patrones y el empleo de la plataforma Odoo, permitió obtener un sistema mantenible, escalable y flexible ante posibles cambios en la implementación de las funcionalidades.

CAPÍTULO III: Implementación y pruebas del Sistema de Gestión para la distribución y consumo del combustible de la UCI

3.1. Introducción

En el presente capítulo se explican a partir de los resultados del diseño, la implementación del Sistema de Gestión para la distribución y consumo del combustible en la UCI, mediante el diagrama de componentes, el diagrama de despliegue, los estándares de codificación y los tratamientos de errores. Finalizando con el análisis de los resultados obtenidos en la validación de la solución y las variables de la investigación.

3.2. Implementación.

La implementación constituye una de las disciplinas de la metodología AUP-UCI para el desarrollo del software. En ella se toman como punto de partida los resultados obtenidos en el análisis y diseño. A continuación se muestran aspectos relevantes de la implementación.

3.2.1. Diagrama de componentes.

Para el desarrollo del componente Combustible, se requirió consumir los servicios existentes en los componentes de Odoo. A continuación se muestran dichos componentes:

- **Base:** Se encarga de gestionar las acciones, a las que tendrán acceso los usuarios, según el rol que desempeñen en la entidad y el grupo al que pertenezca. Información que es consumida por el componente Combustible.
- **Account (Contabilidad):** Es un componente perteneciente de Odoo, que dentro de sus funciones tiene establecer los años fiscales y los períodos mensuales.
- **Fleet (Flota):** Es un componente de Odoo, que ayuda a gestionar todos los vehículos, los contratos asociados a los vehículos, así como las entradas de registro de combustible y los costos.

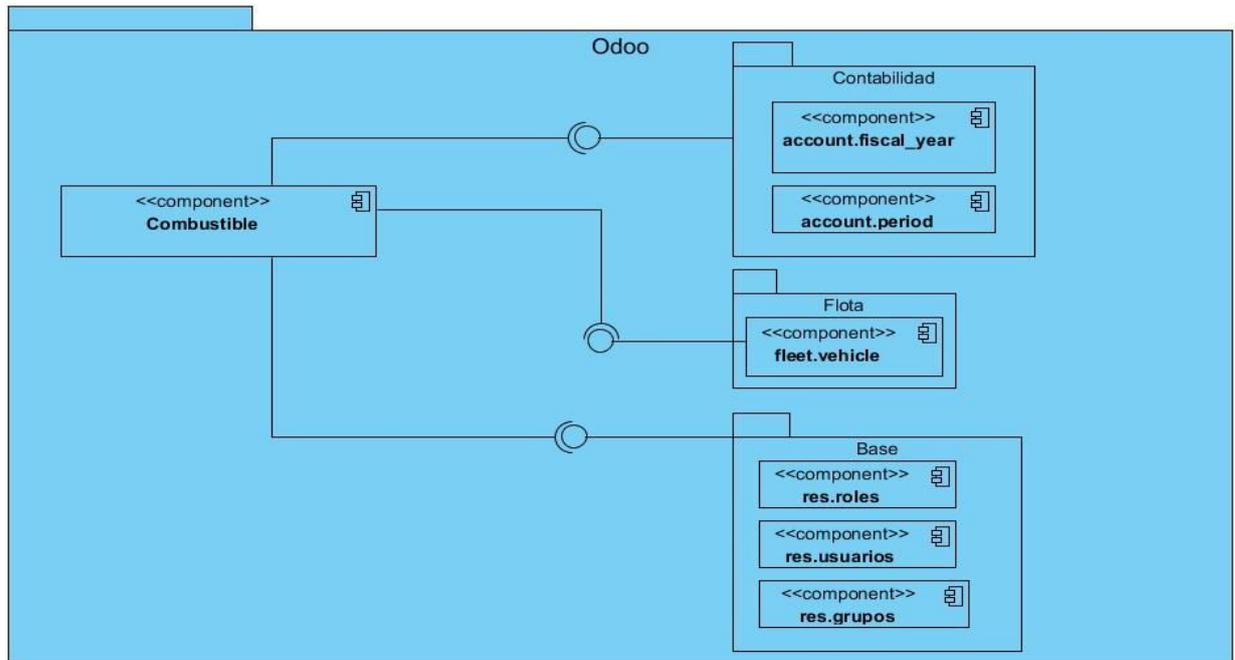


Figura 15. Diagrama de componentes del Sistema de Gestión para la distribución y consumo del combustible en la UCI.

3.2.2. Diagrama de despliegue.

Los diagramas de despliegue proveen una vista de cómo los componentes se despliegan en el transcurso de la infraestructura del sistema. Permite comprender la correspondencia entre la arquitectura de software y la de hardware.

El siguiente diagrama de despliegue muestra la forma en que se van a comunicar los componentes del sistema. La PC cliente (usuario) se comunica por el protocolo HTTPS (Protocolo seguro de transferencia de hipertexto) para utilizar las funcionalidades que se encuentran en el Servidor web. Este a su vez se comunica con el Servidor de Base de Datos por medio del protocolo TCP/IP (Protocolo de control de transmisión) para la obtención de la información que maneja el sistema. Además, para el despliegue del sistema, se tiene conectada la PC Cliente con una impresora con conexión USB donde se podrá obtener de forma física los reportes e informaciones generadas por el sistema.

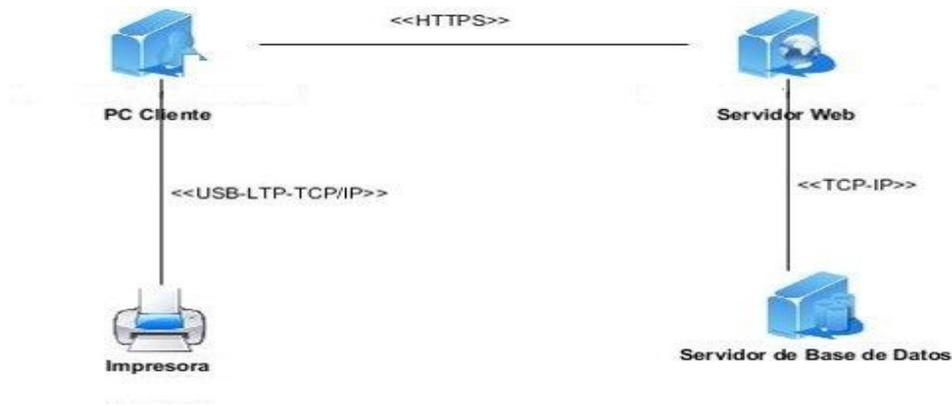


Figura 16. Diagrama de despliegue del Sistema de Gestión para la distribución y consumo del combustible en la UCI.

3.2.3. Estándares de codificación.

Un estándar de codificación comprende aspectos de la generación de código que los programadores deben implementar de forma prudente. El código fuente debe reflejar un estilo armonioso, como si un único programador hubiese escrito todo el código de una sola vez. Es importante establecer un estándar de codificación al comenzar un proyecto para propiciar el trabajo de forma coordinada. (40)

Odoo, como plataforma de desarrollo del Sistema de gestión para la distribución y consumo del combustible en la UCI, presenta estándares de codificación definidos en el documento PEP8 definiéndolos de la siguiente manera: (41)

- **Indentación:** Utilizar siempre 4 espacios y nunca mezclar tabuladores y espacios.
- **Tamaño máximo de línea:** Las líneas deben limitarse a un máximo de 79 caracteres.
- **Líneas en blanco:** Separar las definiciones de las clases y funciones con dos líneas en blanco. Los métodos dentro de clases se separan con una línea en blanco. Se recomienda utilizar líneas en blanco para separar partes del código, por ejemplo dentro de una función, que realizan tareas diferenciadas.
- **Imports:** Los imports de distintos módulos deben estar en líneas diferentes.
- Sí se pueden poner en una línea los elementos que se importan de un mismo módulo.

- Los imports deben ponerse siempre al principio del fichero, justo después de los comentarios y de la documentación y antes de la definición de las variables globales y las constantes.

Espacios en blanco en expresiones: Evitar espacios en blanco extra en:

- Dentro de paréntesis.
- Después de una coma.
- Antes del paréntesis de una llamada a una función.
- Antes del paréntesis de un índice.

```
@api.depends('cierre_combustible_id', 'actividad_id')
def _dependencia_cierre(self):...

@api.depends('cierre_combustible_id', 'cons_total', 'nivel')
def _dependencia_calc(self):

    if self.cierre_combustible_id and self.cons_total and self.nivel:
        self.norma = self.nivel / self.cons_plan

        if self.cons_total == 0:
            self.indice = 0
        else:
            self.indice = self.nivel / self.cons_total
            self.desv = 100 - (self.indice / self.norma * 100)
```

Figura 17. Ejemplo de estándar de codificación en Odoo.

3.2.4. Tratamiento de errores.

En Odoo existen diferentes formas para validar la información, estos tipos de validación son: (42)

Mediante funciones Python: Este tipo de restricción se definen mediante el atributo `_constraints`.

Mediante restricciones de la Base de Datos (SQL Constraint): Este tipo de restricción se define en la clase o en la tabla a la cual se está validando que los datos cumplan con los requisitos que se definen.

Además Odoo contiene una librería `exceptions` que valida la información que introduce el usuario contra la contenida en el sistema.

3.3. Validación de la Solución

A continuación se muestra los resultados obtenidos en la aplicación de las pruebas internas, de liberación y de aceptación. Haciendo uso de los tipos pruebas de caja blanca y caja negra para las pruebas internas.

3.3.1. Pruebas internas

Pruebas de Caja Blanca

Para aplicar el método de pruebas de caja blanca se empleó la técnica del camino básico. Esta permitió obtener una medida de la complejidad lógica para el diseño de los casos de pruebas y usar esta medida como guía para la definición de un conjunto básico de caminos de ejecución. Se tomó como ejemplo el método `dependencia_calc(self)`, perteneciente a la clase `cierres`. Para ello se procede a enumerar las sentencias del código y a partir del mismo se construye el grafo de flujo asociado.

```
@api.depends('cierre_combustible_id', 'cons_total', 'nivel')
def _dependencia_calc(self):
    1 if self.cierre_combustible_id and self.cons_total and self.nivel:
        2 self.norma = self.nivel / self.cons_plan
        3 if self.cons_total == 0:
            4 self.indice = 0
        5 else:
            6 self.indice = self.nivel / self.cons_total
        7 self.desv = 100 - (self.indice / self.norma * 100)
```

Figura 18. Método `dependencia_calc`.

A continuación se realiza el grafo de flujo partiendo del código tomado



Nodos Predicados

Ri: Regiones.

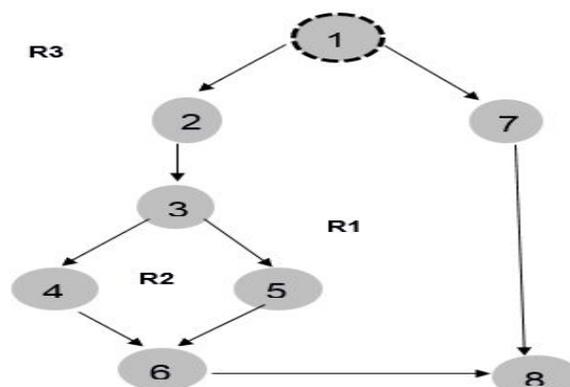


Figura 19. Grafo de flujo asociado al método dependencia_calc.

Luego de haber realizado la construcción del grafo de flujo se procede a calcular la complejidad ciclomática mediante las siguientes fórmulas que se describen a continuación. Donde:

- $V(G)$: Representa el valor del cálculo
- A : Representa la cantidad las aristas
- N : Representa la cantidad de Nodos
- P : Representa la cantidad de Nodos predicados
- R : Representa la cantidad de regiones

$$\begin{aligned} 1- V(G) &= (A-N)+2 \\ V(G) &= (9-8)+2 \\ V(G) &= 3 \end{aligned}$$

$$\begin{aligned} 2- V(G) &= P+1 \\ V(G) &= 2+1 = 3 \end{aligned}$$

$$\begin{aligned} 3- V(G) &= R \\ V(G) &= 3 \end{aligned}$$

El cálculo efectuado anteriormente dio como resultado que, la complejidad ciclomática es de 3, este valor indica que existen 3 posibles caminos por donde el flujo puede circular y determina el número de casos de prueba que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez. A continuación se representan los caminos básicos por los que puede transitar el flujo.

- Camino 1: 1, 2, 3, 4, 6, 8
- Camino 2: 1, 2, 3, 5, 6, 8
- Camino 3: 1, 2, 3

Después de haber extraído los caminos básicos del flujo, se procede a ejecutar los casos de pruebas para este procedimiento, se debe realizar al menos un caso de prueba por cada camino básico. Para realizarlos es necesario cumplir con las siguientes exigencias:

- ∪ Descripción: Se hace la entrada de datos necesaria, validando que ningún parámetro obligatorio pase nulo al procedimiento o no se entre algún dato erróneo.
- ∪ Condición de ejecución: Se especifica cada parámetro para que cumpla una condición deseada para ver el funcionamiento del procedimiento.
- ∪ Entrada: Se muestran los parámetros que entran al procedimiento.

- ✓ Resultados esperados: Se expone el resultado que se espera que devuelva el procedimiento.
- ✓ Resultados: Se muestra el resultado obtenido.
- ✓ Salida: Se presenta el valor final.

Camino básico # 1: 1, 2, 3, 4, 6, 8	
Descripción	Se calcula la norma de consumo, el índice de consumo y el porcentaje de desviaciones para cada actividad.
Condición de ejecución	Deben estar los valores de Nivel de Actividad y
Entrada	if self.cierre_combustible_id and self.cons_tota and
Resultados esperados	Se debe actualizar los datos de la norma de consumo, el índice de consumo y el porcentaje de desviaciones.
Resultados	Se actualizó los datos de la norma de consumo, índice de consumo y el porcentaje de desviaciones.
Salida	True.

Tabla 8. Caso de prueba para el camino #1.

Resultado de la ejecución de las pruebas de caja blanca

Una vez ejecutados todos los casos de pruebas obtenidos a través de la aplicación de la técnica camino básico, se concluye que los mismos fueron probados satisfactoriamente demostrando que el código generado no presenta ciclos infinitos y no existe código innecesario en el sistema desarrollado.

Pruebas de Caja Negra

Las pruebas de caja negra se basan en las especificaciones del sistema a ser probadas para elaborar los casos de prueba. Su comportamiento se evalúa mediante sus entradas y las salidas obtenidas a partir de ellas. Sin embargo, el estudio de todas las posibles entradas y salidas sería impracticable, es por ello que se selecciona un conjunto sobre las que se realizan las pruebas. (43)

Estas pruebas fueron desarrolladas por el Grupo de Aseguramiento de la Calidad del Centro de Informatización de Entidades (CEIGE), seleccionando la técnica de Particiones de Equivalencia que garantizó efectividad al examinar los valores válidos e inválidos de las entradas existentes en el software. A continuación se muestra el ejemplo del caso de prueba del requisito Adicionar vehículo pesado.

Leyenda: No es necesario proporcionar un valor en este campo (NA), Válido (V), Inválido (I).

Escenario	Descripción	Modelo	Chapa	Función	Índice Consumo	Fuera de circulación	Respuesta del sistema	Flujo central
EC 1.1: Crear vehículo pesado con todos los campos correctos.	Permite crear vehículo pesado con sus datos correspondientes.	V Marca44 / 44M	V HP5678	NA Recorrido	V 1	NA no seleccionado	El sistema inserta los datos introducidos por el usuario y crea un vehículo pesado.	El usuario selecciona la tarea Gestionar vehículos pesados, luego selecciona la opción de crear e inserta los datos correspondientes y después selecciona la opción de guardar.
		V Audi / 44p	V Y78	NA vacío	V 2,3	NA seleccionado		
EC 1.2: Crear vehículo pesado con los campos vacíos.	El sistema no permite el vehículo pesado porque existen campos obligatorios vacíos.	I vacío	V HP5678	NA Recorrido	V 5,3	NA no seleccionado	El sistema muestra los campos inválidos mediante un mensaje de error.	El usuario selecciona la tarea Gestionar vehículos pesados, luego selecciona la opción de crear e inserta algunos de los datos correspondientes y después selecciona la opción de guardar.
		V Audi / 44p	I vacío	NA vacío	V 2,3	NA seleccionado		
		V Audi / s4	V H59654	NA Trompo	I vacío	NA seleccionado		
EC 1.3: Crear vehículo pesado con los campos incorrectos.	El sistema no permite el vehículo pesado porque existen campos incorrectos.	V Marca44 / 44M	V HP5678	NA Distribuidor	I letra	NA no seleccionado	El sistema muestra los campos inválidos mediante un mensaje de error.	El usuario selecciona la tarea Gestionar vehículos pesados, luego selecciona la opción de crear e inserta los datos correspondientes y después selecciona la opción de guardar.
		I Marca 67	V H67899	NA Trompo	V 1,2	NA seleccionado		
		V Audi / 44U	I H9	NA Trompo	V 3,1	NA seleccionado		
EC 1.4: Crear vehículo pesado con los campos repetidos.	El sistema no permite el vehículo pesado porque existen campos repetidos.	V Marca44 / 44M	V HP5678	NA Distribuidor	V 1	V no seleccionado	El sistema muestra el siguiente mensaje: El vehículo ya ha sido insertado anteriormente.	El usuario selecciona la tarea Gestionar vehículos pesados, luego selecciona la opción de crear e inserta los datos correspondientes y

después selecciona la opción de guardar.

Tabla 9.Caso de prueba del requisito: Adicionar vehículo pesado

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Modelo	lista desplegable	NO	Campo que permite números y letras.
2	Chapa	campo de texto	NO	Campo que permite números y letras.
3	Función	lista desplegable	SI	Campo que permite números y letras.
4	Índice de consumo	campo de texto	NO	Campo que solo permite números
5	Fuera de circulación	checkbox	SI	atributo de selección

Tabla 10.Caso de prueba del requisito: Adicionar vehículo pesado.

Resultado de la ejecución de las pruebas de caja negra

Durante la aplicación de las pruebas se detectaron trece no conformidades en la primera iteración, que fueron solucionadas, en una segunda iteración se detectaron seis no conformidades, las cuales fueron corregidas y posteriormente en una tercera iteración no se identificaron no conformidades. Prueba de ello lo constituye el acta de liberación expedida por el grupo de calidad de CEIGE.

Figura 20.Resultado de la aplicación de las pruebas internas.

3.3.2. Pruebas de aceptación

Luego de la culminación de las pruebas de caja blanca y caja negra se procedió a realizar las pruebas de aceptación; estas últimas bajo un ambiente controlado son capaces de determinar si el software realiza lo que desea el cliente. Terminadas estas pruebas el cliente avaló la solución con la entrega del Acta de Aceptación del cliente. Para esta investigación, se tuvo como cliente al Vicerrector Económico José Manuel León Cano.

3.4. Aplicación de la técnica ladov

Para valorar el grado de satisfacción del cliente con la solución desarrollada respecto a mejorar la gestión de la información, se aplicó la técnica ladov que permite el estudio del grado de satisfacción del personal involucrado en un proceso objeto de análisis. Durante la valoración fungieron como clientes: el Vicerrector Económico, especialistas del Grupo Energético, de Caja, de la Dirección de Transporte y del Área de Contabilidad y Finanzas, para un total de siete personas. Como resultado se obtuvo:

¿Le gustaría hacer uso del sistema propuesto para desarrollar los procesos asociados a la distribución y consumo del combustible en la UCI?	¿Considera usted oportuno continuar ejecutando los procesos asociados a la distribución y consumo del combustible manualmente, a pesar del gran volumen de información que se genera?								
	No			No Sé			Si		
	¿El desarrollo del Sistema de Gestión para la Distribución y Consumo del combustible (SGDC) contribuye a mejorar la gestión de la información asociada a este proceso en cuanto a organización, supervisión, acceso y seguridad?								
	Si	No Sé	No	Si	No Sé	No	Si	No Sé	No
Me gustaría mucho	1	2	6	2	2	6	6	6	6
No me gustaría mucho	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgustaría más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gustaría nada	6	6	6	6	4	4	6	4	6
No sé qué decir	2	3	6	3	3	3	6	3	4

Tabla 11. Cuadro lógico de ladov.

De manera que el ISG = 0,86

Los resultados de la satisfacción individual según las categorías empleadas fueron los siguientes:

Nivel de satisfacción	Cantidad	%
Máxima satisfacción	6	85,71
Más satisfecho que insatisfecho	1	7,14
No definida	0	0

Tabla 12. Cuadro de satisfacción individual.

Al procesar las respuestas de la encuesta (ver Anexo #3) en el cuadro lógico de ladov, se obtiene un grado de satisfacción grupal de 0,86, lo cual se traduce en una clara satisfacción con el uso del Sistema de gestión para la Distribución y consumo del combustible en la UCI.

En el criterio respecto a mejorar la gestión de la información a través del uso de la solución propuesta, hubo una concordancia de un 100% en que contribuye a su mejora. De igual manera el 100% de los participantes manifestó que le gustaría mucho hacer uso de este sistema para desarrollar los procesos asociados a la distribución y consumo de combustible. Mientras que un 7,14% no sabe si sería oportuno continuar ejecutando estos procesos manualmente, a pesar del gran volumen de información que se genera.

Las preguntas abiertas que se formularon fueron:

- ¿Qué valoraciones le sugiere al sistema respecto a mejorar la gestión de la información asociada a estos procesos?
- ¿Qué elemento(s) usted adicionaría a la solución que se propone?
- Entre las valoraciones positivas obtenidas como respuestas a las preguntas abiertas, se recopilaron criterios como los siguientes:
- El sistema permite mejorar la gestión de la información asociado al proceso de distribución y consumo de combustible, en las diferentes áreas por las que transita.
- Facilita la obtención de reportes para una gestión proactiva desde la vicerrectoría económica.

- Permite establecer un mejor control de la ejecución del presupuesto y la visualización del mismo por parte de cada jefe de área.

La aplicación de la técnica de ladov aportó información significativa respecto al grado de satisfacción del cliente. Los resultados obtenidos y los criterios emitidos validan la fortaleza de la propuesta, reflejándose una valoración muy positiva del cliente con la solución.

3.5. Validación de las variables de la investigación

Para validar las variables de la investigación se aplicó una encuesta, donde se evidencia como se realiza el proceso en la actualidad y una vez desarrollada la propuesta de solución, con el objetivo de caracterizar la gestión de la información asociada al proceso de distribución y consumo del combustible en cuanto a organización, supervisión, acceso y seguridad.

Análisis de los resultados de la investigación

A continuación en la siguiente tabla se muestran los resultados obtenidos luego de haber tabulado la información recogida en la encuesta aplicado a 7 especialistas de las áreas de Vicerrectoría económica, Grupo de gestión energética, Dirección de transporte, Finanzas, Caja y Contabilidad de la UCI.

Indicadores.		Proceso Actual.	Proceso con la utilización del Sistema implementado.
Organización de la información		Los datos son manejados desde diferentes áreas de trabajo por separadas, lo que conlleva a una descentralización de la información.	Con el sistema desarrollado todos los procesos se podrán realizar de forma íntegra, garantizando la centralización de la información.
Seguridad	Integridad	Actualmente la información se encuentra en formato duro, y en documentos Excel por lo que se encuentra vulnerable al deterioro o alteraciones, que constituyen factores que atentan contra su integridad.	La propuesta de solución contribuye a la integridad de la información al contar con una base de datos que guarda todas las informaciones de las operaciones que se realizan.
		Las cifras con las que se trabajan en las diferentes áreas contienen referencias a otros documentos, por lo que en ocasiones la información	Las cifras se actualizan a medida que se van introduciendo datos en el sistema, validándose que no existan errores de cálculo.

		puede sufrir alteraciones debido a errores que se cometen.	
	Confidencialidad	Existe el riesgo de que personas externas tengan acceso a informaciones generadas en el proceso.	La información solo puede ser conocida por el personal autorizado en el sistema.
Acceso a la información		La documentación se encuentra en formato duro, almacenada en estantes, lo que dificulta la búsqueda para consultar la información.	La documentación se encuentra en formato digital y está almacenada en base de datos, por lo que se puede acceder durante la ejecución de los procesos como una vez culminado los mismos.
		El acceso a la información se realiza a través del personal responsable de cada área.	El acceso a la información puede ser realizada por los usuarios según el permiso que tengan en el sistema en cualquier momento.
Supervisión		Es engorroso supervisar eficientemente a las áreas subordinadas considerando que son varias y las supervisiones se realizan personalmente.	Con el sistema desarrollado se facilita la supervisión a la información gestionada por las áreas de un modo más ágil.
		Generar los reportes solicitados por la Vicerrectoría económica se dificultan, debido a la gran cantidad de información que controlan.	Los reportes son generados por el sistema mediante llamadas a la base de datos que permiten obtener la información necesaria.

Tabla 13. Validación de la variable de investigación

3.6. Conclusiones parciales

Al finalizar el presente capítulo se puede arribar a las siguientes conclusiones:

- Se realizó el diagrama de componente y de despliegue para mostrar una vista de cómo quedó el sistema a nivel de componentes y su distribución. En cuanto los estándares de codificación empleados en la implementación permitieron ganar en legibilidad, claridad y mayor entendimiento del código por parte de los desarrolladores. El tratamiento de errores posibilitó evitar que datos incorrectos fuesen introducidos en los diferentes campos de los formularios mediante el uso de determinados métodos de validación.
- Se aplicaron pruebas de caja negra para verificar el correcto funcionamiento del software donde se arrojaron los siguientes resultados: en la primera iteración se detectaron trece no conformidades que fueron solucionadas, en una segunda iteración se detectaron seis no conformidades, las cuales fueron corregidas y posteriormente en una tercera iteración no se identificaron no conformidades.
- La validación de las variables de la investigación y la aplicación de la técnica ladov, permitieron medir la satisfacción del cliente respecto a mejorar la gestión de la información en cuanto a organización, supervisión, acceso y seguridad, haciendo uso del sistema desarrollado, donde los resultados obtenidos se traducen en un elevado agrado por parte del cliente con la solución.

CONCLUSIONES GENERALES

- El estudio de sistemas que manejan el proceso de distribución y consumo del combustible, evidenció que los mismos no brindan una solución completa a los problemas existentes, lo cual constituyó motivación para el desarrollo de la presente investigación, que utiliza como framework OpenObject, empleando la metodología de desarrollo aprobada en el plan de mejoras de la UCI, garantizando un ambiente visual ameno y organizado con el empleo de la plataforma Odoon, pyCharm 4.5.4, PostgreSQL 9.1, Apache 2.2, python 2.7 ,entre otras herramientas.
- Se describieron las características del sistema a partir de un modelado del negocio previo, para ello se identificaron requisitos funcionales y no funcionales, los cuales fueron validados y el resultado de dicha validación constató que los mismos en su totalidad no son ambiguos, son correctos, comprensibles y posibles de tracear. En cuanto a la verificación del diseño realizado, mediante la aplicación de las métricas TOC y RC, se evidenció una baja responsabilidad y complejidad de implementación, la cual favoreció la reutilización corroborándose así una correcta asignación de responsabilidades.
- La aplicación de los estándares de codificación permitió una implementación responsable y organizada facilitando la obtención del sistema de gestión para la distribución y consumo del combustible en la UCI.
- Para verificar el correcto funcionamiento de la solución obtenida se aplicaron pruebas internas, de liberación y de aceptación, en cuanto a las pruebas de caja negra en una primera iteración se detectaron trece no conformidades que fueron solucionadas, en una segunda iteración se detectaron seis no conformidades, las cuales fueron corregidas y posteriormente en una tercera iteración no se identificaron no conformidades.
- La aplicación de la técnica ladov y la validación de las variables de la investigación, aportaron información significativa respecto al grado de satisfacción del cliente. Los resultados obtenidos y los criterios emitidos validan la fortaleza de la propuesta, reflejándose una valoración positiva del cliente con la solución.

RECOMENDACIONES

- Que el sistema sea capaz de generar automáticamente las planificaciones anuales, a partir de un estudio de los historiales de años anteriores.
- Que el sistema sea capaz de gestionar las notificaciones para informar al usuario las tarjetas que están por vencer y los choferes que faltan por liquidar en un período determinado.
- Vincular el sistema con el módulo de Contabilidad del centro CEIGE y el módulo de Presupuesto que se está desarrollando en la Vicerrectoría Económica.

REFERENCIAS

1. **La gestión energética en el contexto empresarial cubano. . Pérez, Osvaldo. s.l. : Revista caribeña de Ciencias Sociales, 2013.**
2. **Recursos Energéticos. [En línea] DEMO E-DUCATIVA CATEDU, 2010. http://e-educativa.catedu.es/44700165/aula/archivos/repositorio//3500/3669/html/3_recursos_energicos.html..**
3. **Definición de combustible. *Definición abc.* [En línea] <http://www.definicionabc.com/general/combustible.php>..**
4. **Centro de innovación para la logística y el transporte por carrera. [En línea] <http://www.definicionabc.com/general/combustible.php>..**
5. **Laguna, Lic. Yosmicel Mendoza. yosmicelmlarrobault.edu.cu, Cuba : Centro Universitario Las Tunas.**
6. **Delgado, Lic. Omar Gómez. omargdarrobault.edu.cu, Cuba : Centro Universitario Las Tunas.**
7. **EUGCOM. *Descripción del Producto ADMINISTRACIÓN Y CONTROL DE FLOTAS.* Chile : s.n., 2009.**
8. **KreaSoft. KreaSoft Software y Tecnología. [En línea] 2010. <http://www.kreasoft.cl/kreaflota.html>..**
9. **Incremental, Software. FlotaWeb. [En línea] 2010. <http://flotaweb.com/>..**
10. **Odoo. SA, Odoo. España : s.n., 2014.**
11. **Saavedra, A., Rodríguez, R. SAAVEDRA. *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.* [En línea] 2013. http://bibliodoc.uci.cu/RDigitales/2013/noviembre/26/TD_06861_13.pdf..**
12. ***Software Engineering: A Practitioner's Approach.* Pressman, Roger. 2009.**
13. **Sánchez, Tamara Rodríguez. *Metodología de desarrollo para la actividad productiva de la UCI.* La Habana : s.n., 2015.**

14. Gracia, L. **Un poco de Java. Técnicas para la captura de Requisitos.** [En línea] 2013. [http://unpocodejava.wordpress.com/2013/05/09/tecnicas-para-la-captura-de-requisitos/..](http://unpocodejava.wordpress.com/2013/05/09/tecnicas-para-la-captura-de-requisitos/)
15. Sommerville, Ian. **Software Engineering. 8va.** Boston : s.n., 2011.
16. Andalucía. **Marco de Desarrollo de la Junta de Andalucía.** [En línea] 2013. <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/415>.
17. Buschmann, F., y otros. **Pattern Oriented Software Architecture: A System of Patterns. s.l. : John Wiley & Sons, 1996.** 1996.
18. **Patrón Modelo-Vista-Controlador.** . Díaz González, Yanette y Fernández Romero, Yenisleidy. 1, La Habana : CUJAE : s.n., 2012, Vol. 11. 1729-3804.
19. Gamma, Dr. Erich, y otros. **Design Patterns: Elements of Reusable Object-Oriented Software.** 1994.
20. Lorenz, Mark y Kidd, Jeff. **Object-Oriented Software Metrics.** 1994.
21. Chidamber, Shyam R. y Kemerer, Chris F. **A Metrics Suite for Object Oriented Design. s.l. : Software Engineering, IEEE Transactions.** 1994.
22. García Ramírez, Christian y Pavón Reyes, Raydel. **Componente dinámico para la interoperabilidad de procesos.** La Habana : s.n., 2013.
23. Reis, Daniel. **Odoo Development Essentials.** 2015.
24. DUQUE, Raúl González. **Python para todos.** España : s.n., 2011.
25. Lapuente, María Jesús Lamarca. **De Marc21 a RDF.** [En línea] 2013. <http://www.bn.gov.ar/media/page/mostacciom-blancon-gattafonis-uviniar-durantib-yedidn-ponencia.pdf>.
26. **PostgreSQL: Características , limitaciones y ventajas.** [En línea] 2016. <http://postgresql-dbms.blogspot.com/p/limitaciones-puntos-de-recuperacion.html>.
27. KABIR, Mohammed J. **Servidor Apache 2.** 2003.
28. ISLAM, Quazi Nafiul. **Mastering PyCharm. Packt Publishing.** 2015.
29. Orallo, Enrique Hernadez. **El lenguaje Unificado de Modelado (UML).**

30. White, Stephen A. *Introduction to BPMN*. 2014.
31. Visual Paradigm. [En línea] 2010. <https://www.visual-paradigm.com/>.
32. CEIGE. Gestor de Documentos Administrativos. [En línea] 2015. <http://excriba.prod.uci.cu/page/context/shared/document-details?nodeRef=workspace://SpacesStore/a6002b5f-d3cb-4217-b90d-ddd55621c271..>
33. Pruebas de caja blanca - EcuRed. [En línea] http://www.ecured.cu/Pruebas_de_caja_blanca.
34. Kuzmina. *Metódicas Investigativas de la actividad pedagógica*. Leningrado. 1970.
35. LARMAN, CRAIG. Modelo Conceptual. [En línea] <http://dspace.ucbscz.edu.bo/dspace/bitstream/123456789/13197/1/2409.pdf>.
36. Baez Fernández, Liannet. *Expediente de proyecto 3.4 del centro CEIGE*. Universidad de las Ciencias Informáticas. La Habana, Cuba: : s.n., 2015.
37. IEEE, Arquitectura -. *Arquitectura - IEEE 1471*. 2000.
38. Tutorial de UML - Modelo de Clases. [En línea] 2016. <http://users.dcc.uchile.cl/~psalinas/uml/modelo.html>.
39. Definición de un modelo de datos. [En línea] 2016. <https://tombasededatos.wordpress.com/2010/08/28/2-1-definicion-de-un-modelo-de-datos/>.
40. Microsoft. Microsoft Developer Network. Revisiones de código y estándares de codificación. [En línea] 2015. [https://msdn.microsoft.com/es-es/library/aa291591\(v=vs.71\).aspx..](https://msdn.microsoft.com/es-es/library/aa291591(v=vs.71).aspx..)
41. Nick Coghlan, Barry Warsaw. Style Guide for Python Code. [En línea] 2011. <https://www.python.org/dev/peps/pep-0008/>.
42. Restricciones en OpenERP mediante PostgreSQL ~ Programando con Odoo (OpenERP) y Python. [En línea] 2016. <http://poncesoft.blogspot.com/2014/01/restricciones-en-openerp-mediante.html>.

43. Pruebas de caja negra - EcuRed. [En línea]
http://www.ecured.cu/Pruebas_de_caja_negra

ANEXOS

Clases	Atributos	Métodos	Responsabilidad	Complejidad	Reutilización
VehiculoTipo	4	5	Baja	Baja	Alta
BaseAlimento	10	6	Baja	Baja	Alta
BaseTrompo	12	6	Baja	Baja	Alta
Pom	5	7	Media	Media	Media
CierreCombustible	4	5	Baja	Baja	Alta
PlanBase	3	5	Baja	Baja	Alta
BaseCamión	9	6	Baja	Baja	Alta
TipoCombustible	4	5	Baja	Baja	Alta
AsigMensualVehiculo	8	6	Baja	Baja	Alta
BaseExtraplan	10	6	Baja	Baja	Alta
AsigMensual	4	7	Media	Media	Media
FleetVehicle	21	6	Baja	Baja	Alta
VehiculoFunción	4	5	Baja	Baja	Alta
Dependencia	4	5	Baja	Baja	Alta
Área	4	5	Baja	Baja	Alta
FleetVehicletag	4	5	Baja	Baja	Alta
FleetServiceType	5	5	Baja	Baja	Alta
FleetContractState	4	5	Baja	Baja	Alta
ElementoOperativo	4	6	Baja	Baja	Alta
Elemento	6	6	Baja	Baja	Alta
Actividad	5	5	Baja	Baja	Alta
Cierre	10	7	Media	Media	Media
ElementoReal	4	6	Baja	Baja	Alta
Recorridos	5	5	Baja	Baja	Alta
BaseTransporte	9	6	Baja	Baja	Alta
Tarjetas	6	5	Baja	Baja	Alta
ElementoEntrega	8	5	Baja	Baja	Alta
ModeloTarjetas	8	6	Baja	Baja	Alta
FleetVehicleModelBrand	7	5	Baja	Baja	Alta
ElementoEstimado	4	6	Baja	Baja	Alta
ElementoAcumulado	4	6	Baja	Baja	Alta
Pac	5	7	Media	Media	Media
ElementoDemanda	4	6	Baja	Baja	Alta
FleetVehicleModel	7	5	Baja	Baja	Alta

Anexo #1. Resultado de aplicación de la métrica TOC

Clases	Cantidad de Relaciones entre clases	Acoplamiento	Complejidad de mantenimiento	reutilización	Cantidad de pruebas
Clases	1	Bajo	Baja	Alta	Baja
VehiculoTipo	3	Alto	Media	Media	Media
BaseAlimento	2	Media	Baja	Alta	Baja
BaseTrompo	2	Media	Baja	Alta	Baja
Pom	2	Media	Baja	Alta	Baja
CierreCombustible	6	Alta	Alta	Baja	Alta
PlanBase	2	Media	Baja	Alta	Baja
BaseCamión	7	Alta	Alta	Baja	Alta
TipoCombustible	2	Media	Baja	Alta	Baja
AsigMensualVehiculo	1	Baja	Baja	Alta	Baja
BaseExtraplan	2	Media	Baja	Alta	Baja
AsigMensual	10	Alta	Alta	Baja	Alta
FleetVehicle	1	Baja	Baja	Alta	Baja
VehiculoFunción	2	Media	Baja	Alta	Baja
Dependencia	1	Baja	Baja	Alta	Baja
Área	0	No definido	No definido	No definido	No definido
FleetVehicletag	0	No definido	No definido	No definido	No definido
FleetServiceType	0	No definido	No definido	No definido	No definido

FleetContractState	2	Media	Baja	Alta	Baja
ElementoOperativo	6	Alta	Alta	Baja	Alta
Elemento	2	Media	Baja	Alta	Baja
Actividad	2	Media	Baja	Alta	Baja
Cierre	2	Media	Baja	Alta	Baja
ElementoReal	3	Alta	Media	Media	Media
Recorridos	3	Alta	Media	Media	Media
BaseTransporte	1	Baja	Baja	Alta	Baja
Tarjetas	1	Baja	Baja	Alta	Baja
ElementoEntrega	4	Alta	media	Media	media
ModeloTarjetas	1	Baja	Baja	Alta	Baja
FleetVehicleModelBrand	2	Media	Baja	Alta	Baja
ElementoEstimado	2	Media	Baja	Alta	Baja
ElementoAcumulado	5	Alta	Alta	Baja	Alta
Pac	2	Media	Baja	Alta	Baja
ElementoDemanda	2	Media	Baja	Alta	Baja

Anexo #2. Resultado de aplicación de la métrica RC

Anexo#3. Encuesta

Encuesta:

La siguiente encuesta tiene como objetivo caracterizar la gestión de los procesos asociados a la distribución y consumo del combustible en la Universidad de las Ciencias Informáticas (UCI). Entiéndase como procesos:

- ⤵ La planificación anual y mensual del combustible.
- ⤵ La asignación y distribución del combustible.
- ⤵ Eficiencia de consumo de combustible de los vehículos.
- ⤵ Entrega de combustible adicional.
- ⤵ Carga, Entrega, Liquidación y Control de tarjetas magnéticas para combustible.

Por ello le pedimos su contribución y que sea lo más sincero posible en sus planteamientos. Le garantizamos confidencialidad y anonimato. Gracias de antemano por su colaboración.

Datos de interés:

Área a la que pertenece: _____ Rol: _____

Preguntas:

1. ¿Los procesos asociados a la Distribución y Consumo del combustible se realizan en áreas diferentes?

Sí ___ No ___

a. En caso de ser positiva la respuesta especificar en qué área se realizan estos procesos por separado.

1.1. ¿Las áreas se relacionan unas con otras?

Sí ___ No ___

2. ¿Qué mecanismos emplean para garantizar la seguridad de la información?

3. ¿En qué formato se registra la información relacionada con estos procesos?

Excel ___ Word ___ txt ___ PDF ___

3.1 ¿Se genera mucha documentación en la realización de dichos procesos?

Sí ___ No ___

a. En caso de ser positiva la respuesta mencionar algunos inconvenientes que esto puede causar.

4. ¿Considera usted oportuno continuar ejecutando los procesos asociados a la distribución y consumo del combustible manualmente a pesar del gran volumen de información que se genera?

Sí ___ No ___ No Sé ___

5. ¿Qué ventajas a su consideración cree que le puede propiciar el desarrollo de un software que informatice los procesos asociados a la distribución y consumo del combustible?

6. ¿El desarrollo del Sistema de Gestión para la Distribución y Consumo del combustible (SGDC) contribuye a mejorar el control de la información asociada a este proceso en la UCI?

Sí__ No__ No Sé__

7. ¿Le gustaría hacer uso del sistema propuesto para desarrollar los procesos asociados a la distribución y consumo del combustible en la UCI?

Me gustaría mucho

No me gustaría mucho

Me da lo mismo

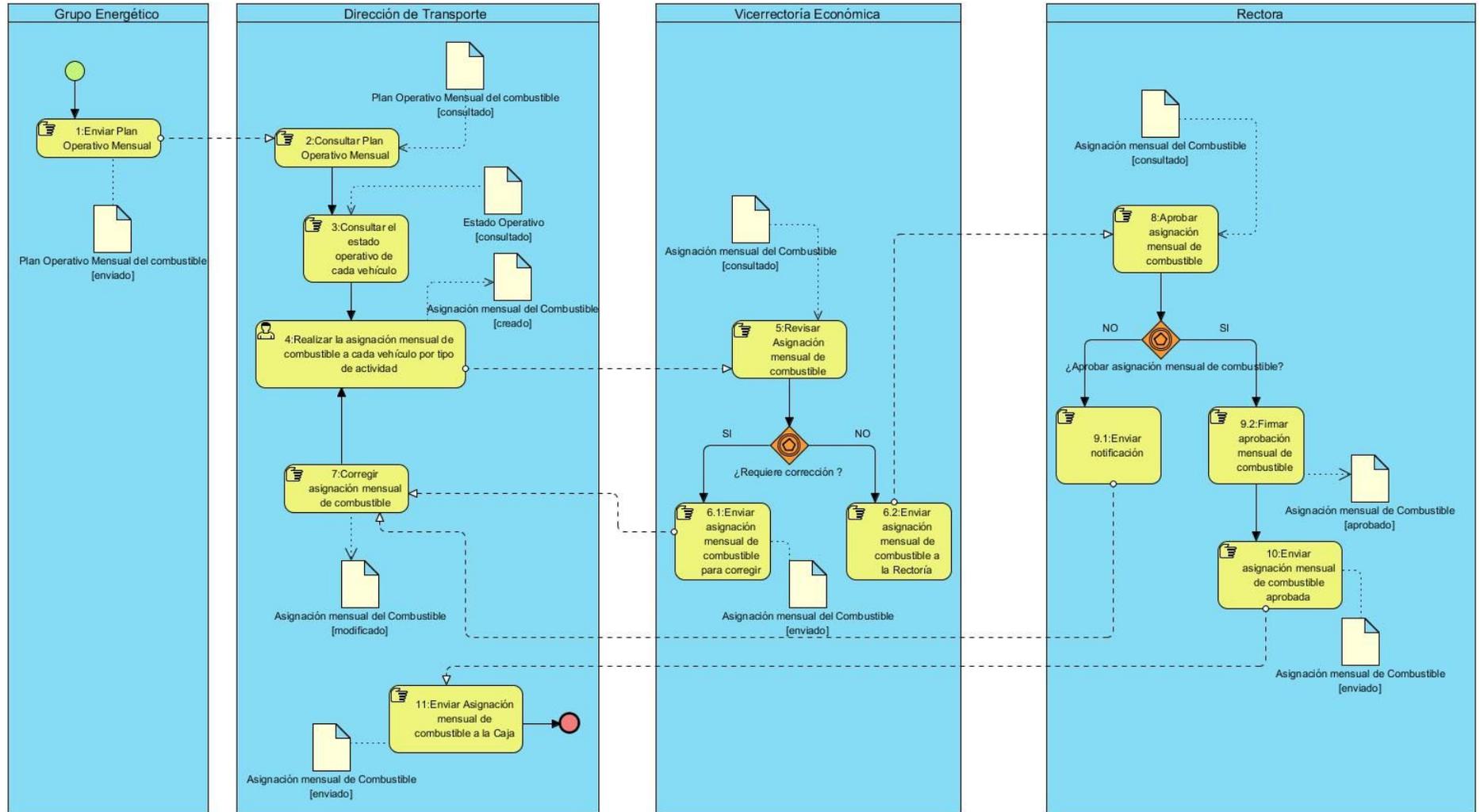
Me disgustaría más de lo que me gusta

No me gustaría nada

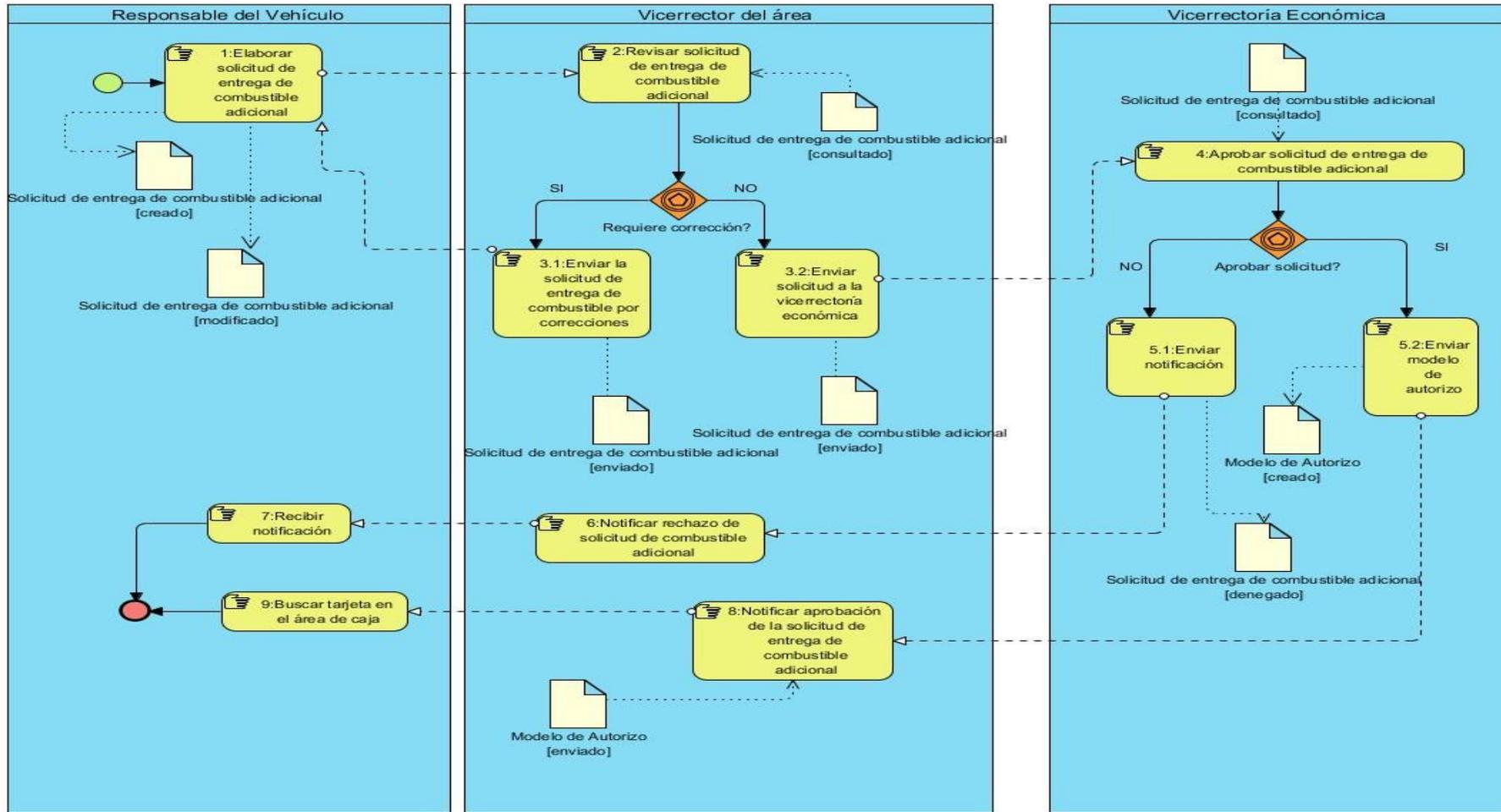
No sé qué decir

8. ¿Qué valoraciones le surgiere el sistema respecto al control de la información asociada a estos procesos?

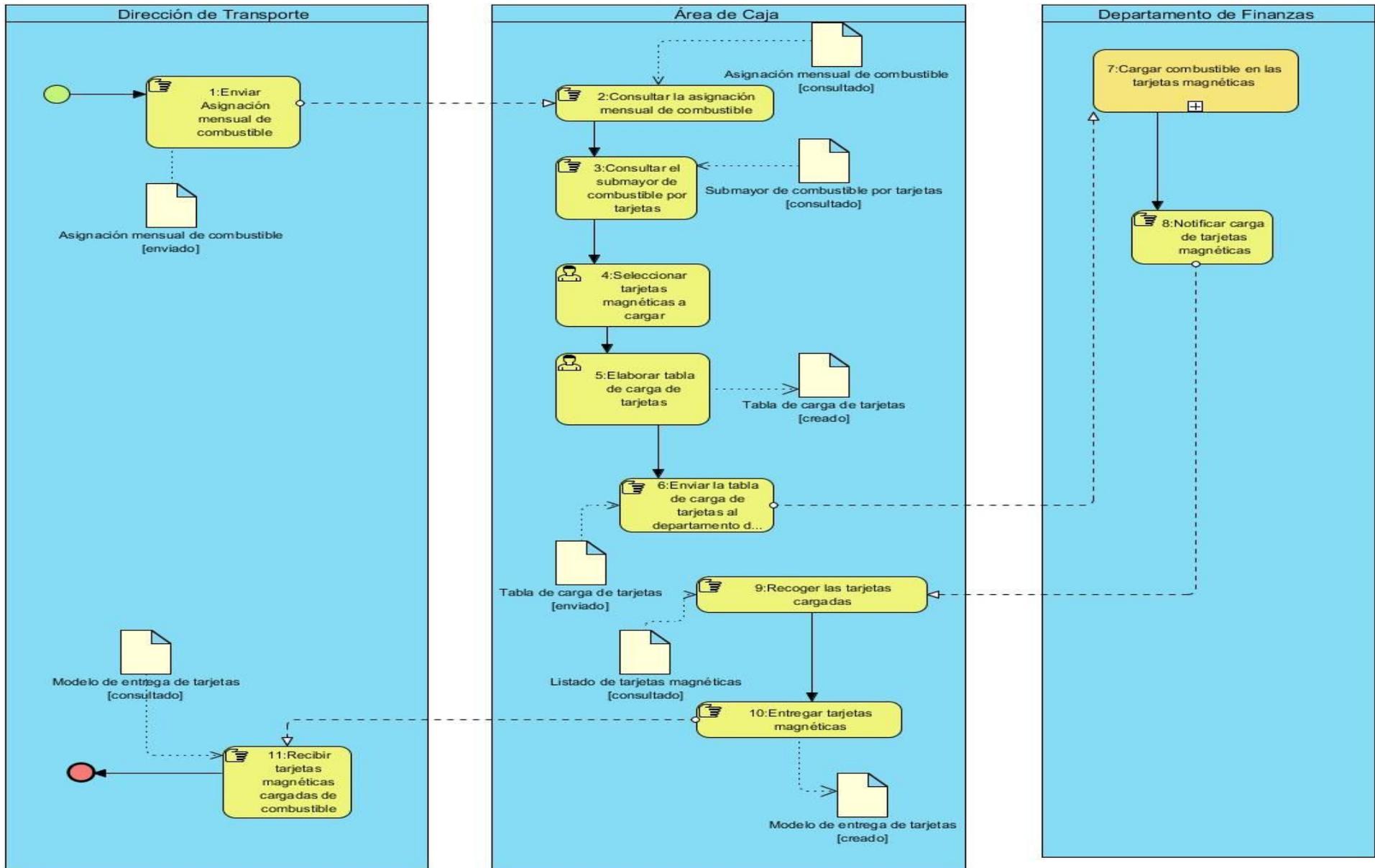
9. ¿Qué elemento(s) usted adicionaría a la solución que se propone?



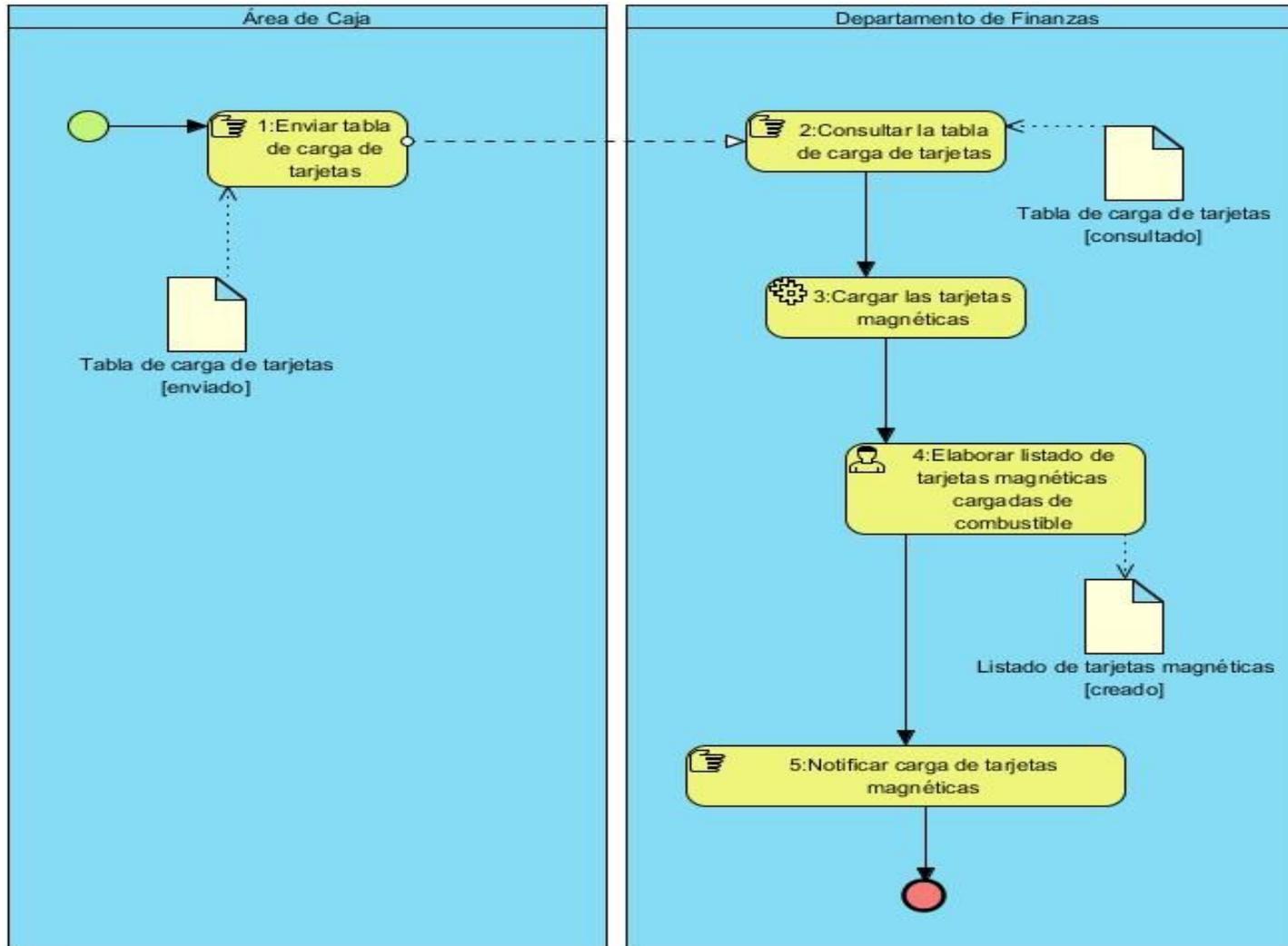
Anexo #4: Diagrama de proceso de negocio: Realizar asignación mensual de combustible



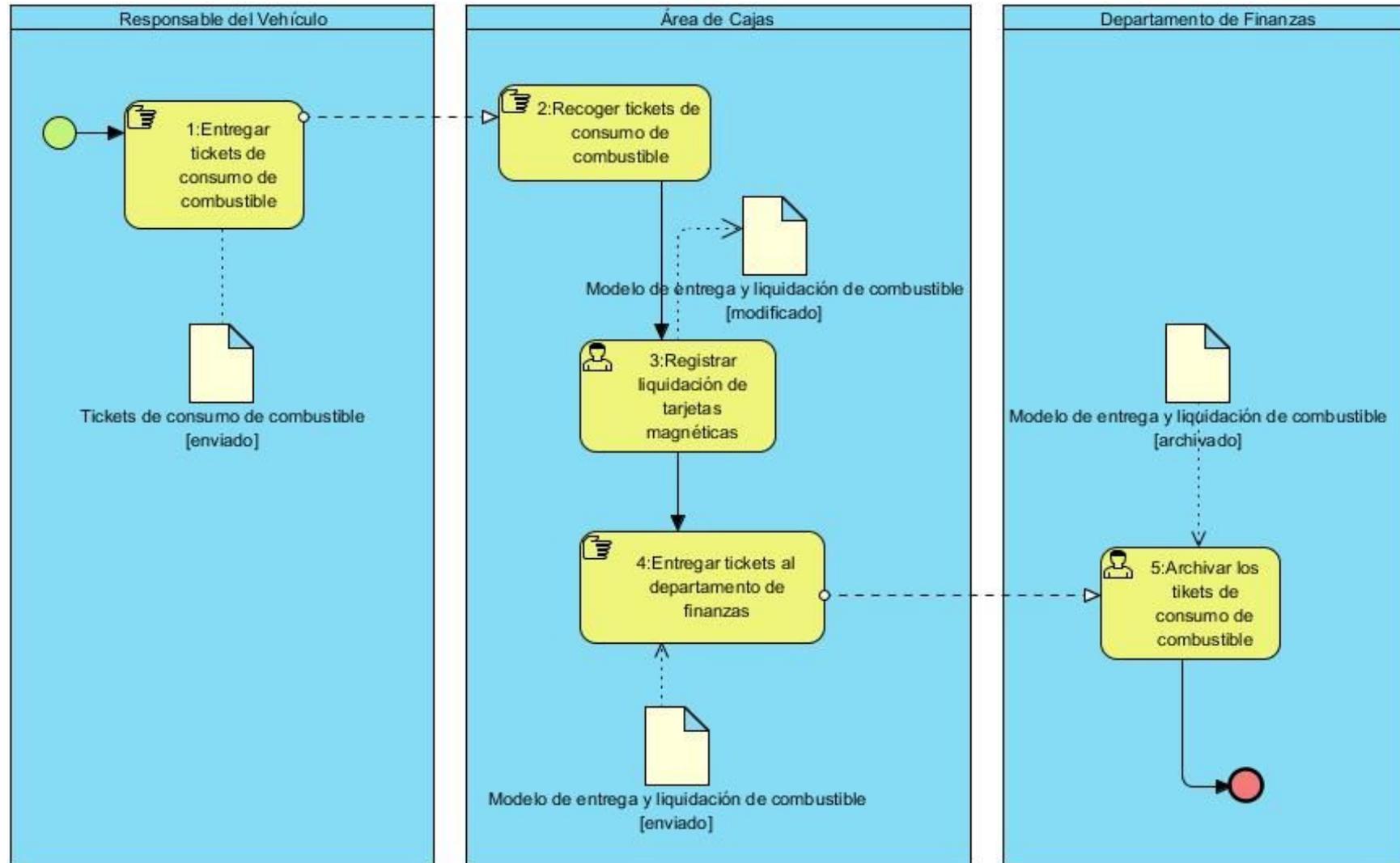
Anexo #5: Diagrama de proceso de negocio: Aprobar entrega de combustible adicional.



Anexo #6: Diagrama de proceso de negocio: Recogida y entrega de tarjetas magnéticas



Anexo #7: Diagrama de proceso de negocio: Cargar combustible en tarjetas magnéticas



Anexo #8: Diagrama de proceso de negocio: Liquidar combustible.

Pregunta cerrada 3	Pregunta cerrada 1								
	No			No Sé			Si		
	Pregunta cerrada 2								
	Si	No Sé	No	Si	No Sé	No	Si	No Sé	No
Me gusta mucho	1	2	6	2	2	6	6	6	6
No me gusta mucho	2	2	3	2	3	3	6	3	6
Me da lo mismo	3	3	3	3	3	3	3	3	3
Me disgusta más de lo que me gusta	6	3	6	3	4	4	3	4	4
No me gusta nada	6	6	6	6	4	4	6	4	6
No sé qué decir	2	3	6	3	3	3	6	3	4

Anexo #9: Cuadro lógico de ladov.

Atributo que afecta	Modo en que lo afecta
Responsabilidad	Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
Complejidad de Implementación	Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.
Reutilización	Un aumento del TOC implica una disminución en el grado de reutilización de la clase.

Anexo #10: Métrica Tamaño Operacional de Clase.

Atributo	Categoría	Criterio
Responsabilidad	Baja	\leq Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$> 2 \times$ Promedio
Complejidad de Implementación	Baja	\leq Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	$> 2 \times$ Promedio
Reutilización	Baja	$> 2 \times$ Promedio
	Media	Entre Promedio y $2 \times$ Promedio
	Alta	\leq Promedio

Anexo #11: Rango de valores para la métrica TOC.

Atributo que afecta	Modo en que lo afecta
Acoplamiento	Un aumento de la RC implica un aumento del acoplamiento de la clase.
Complejidad de	Un aumento de la RC implica un aumento de la complejidad del mantenimiento de la clase.
Reutilización	Un aumento de la RC implica una disminución en el grado de reutilización de la clase.
Cantidad de Pruebas	Un aumento del RC implica un aumento de la cantidad de pruebas de unidad necesarias para probar una clase.

Anexo #12: Relaciones entre clases.

Atributo	Categoría	Criterio
Acoplamiento	Baja	1
	Media	2
	Alta	> 2
Complejidad de mantenimiento	Baja	\leq Promedio
	Media	Entre Promedio y $2 * \text{Promedio}$
	Alta	$> 2 * \text{Promedio}$
Reutilización	Baja	$> 2 * \text{Promedio}$
	Media	Entre Promedio y $2 * \text{Promedio}$
	Alta	\leq Promedio
Cantidad de Pruebas	Baja	\leq Promedio
	Media	Entre Promedio y $2 * \text{Promedio}$
	Alta	$> 2 * \text{Promedio}$

Anexo #13: Rango de valores para la métrica RC.