



**Universidad de las Ciencias Informáticas**

**Facultad 1**

**“Aplicación Web para la Gestión de Información del Combustible en la  
Universidad de las Ciencias Informáticas”**

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

**Autores**

Leidy Pérez González.

Yuniel Alvarez Peña.

**Tutores**

Ing. Alexander Rodríguez Mompíe.

Ing. Yariel Ramos Negrín.

*Ciudad de la Habana, junio, 2010*

*“Año 52 de la Revolución”*

## Declaración de autoría

Por este medio declaramos que Leidy Pérez González y Yuniel Alvarez Peña somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) para que haga el uso que estime pertinente del mismo.

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

---

*Leidy Pérez González.*

**Firma del Autor**

---

*Yuniel Alvarez Peña.*

**Firma del Autor**

---

*Ing. Alexander Rodríguez Mompíe*

**Firma del Tutor**

---

*Ing. Yariel Ramos Negrín*

**Firma del Tutor**

### Resumen

La creciente necesidad de ahorrar y controlar el uso de los combustibles fósiles en el mundo, no sería posible sin el aporte de cada país e institución. Para lo cual la Universidad de las Ciencias Informáticas (UCI), creada en Cuba al fulgor de la Batalla de Ideas, provee la realización de una aplicación web que permita la gestión de información de los procesos desarrollados en los departamentos de: Energía, Transporte, Mantenimiento y Cajero Central; para el control del combustible, el cual hasta el momento se venía realizando a través de planillas y modelos Excel, dificultando la búsqueda, el manejo de la información y la generación de reportes.

La realización de la “Aplicación Web para la Gestión de la Información del Combustible en la Universidad de las Ciencias Informática”, se basó en el desarrollo de las funcionalidades provistas por cada uno de los departamentos mencionados con el fin de: facilitar el acceso a la información así como la generación de reportes, reducir el tiempo empleado en la elaboración de tareas, estructurar el almacenamiento de datos y garantizar niveles de seguridad. En su ejecución se utilizó como: metodología de desarrollo RUP, lenguajes: UML (modelado), PHP (programación), el *framework*\* Symfony, la librería ExtJS, las herramientas: Visual Paradigm, PostgreSQL, NetBeans y el servidor web Apache.

**Palabras Claves:** Gestión, Combustible, Aplicación Web, Información.

**Glosario de Términos:** Las palabras que se señalarán con (\*).

## Índice

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA.....</b>	<b>5</b>
Introducción .....	5
1.1 Las Tecnologías de la Información y la Comunicación (TIC) .....	5
1.2 La gestión del combustible como ente fundamental en el ahorro .....	6
1.2.1 Gestión y administración del combustible en la UCI .....	7
1.2.2 Conceptos asociados al dominio del problema .....	9
1.3 Sistemas automatizados existentes vinculados al campo de acción .....	10
1.3.1 Sistemas Internacionales .....	10
1.3.2 Sistemas Nacionales .....	12
1.4 Proceso de Desarrollo de Software .....	12
1.4.1 Metodologías de Desarrollo de Software .....	12
1.4.2 Metodología de Desarrollo de Software seleccionada (RUP).....	14
1.4.3 Lenguaje de Modelación Visual (UML) .....	15
1.4.4 Lenguaje de programación web (PHP) .....	15
1.5 Tendencias y Tecnologías .....	16
1.5.1 Patrón de Arquitectura Modelo Vista Controlador (MVC).....	16
1.5.2 Entornos de desarrollo.....	16
1.5.3 Servidor Web Apache 2.2.6 .....	18
1.5.4 Librería EXTJS.....	18
1.6 Herramientas Utilizadas .....	18
1.6.1 Sistema Gestor de Base de Datos (SGBD).....	18
1.6.2 NetBeans 6.5 .....	20
1.6.3 Visual Paradigm for UML 6.4 Release Notes .....	21
Conclusiones Parciales .....	21
<b>CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA .....</b>	<b>22</b>
Introducción .....	22
2.1 Modelo de negocio .....	22
2.1.1 Procesos del Negocio.....	22
2.1.2 Entidades del Negocio.....	23
2.1.3 Actores del Negocio.....	24
2.1.4 Trabajadores del Negocio .....	25
2.1.5 Diagrama de casos de uso del negocio .....	25
2.1.6 Descripción de Casos de Uso del Negocio .....	26
2.1.7 Modelo de Objetos del Negocio.....	30

2.1.8 Reglas del Negocio.....	31
2.2 Requisitos de Software.....	31
2.2.1 Requerimientos Funcionales.....	31
2.2.2 Requerimientos no Funcionales.....	33
2.3 Descripción del Sistema Propuesto.....	34
2.3.1 Actores del sistema.....	35
2.3.2 Diagrama de Casos de Uso del Sistema.....	36
2.3.3 Descripciones abreviadas de los casos de uso del sistema.....	38
Conclusiones Parciales.....	40
<b>CAPÍTULO III: ANÁLISIS Y DISEÑO.....</b>	<b>41</b>
Introducción.....	41
3.1 Análisis.....	41
3.1.1 Modelo de Análisis.....	41
3.1.2 Diagramas de Clases del Análisis.....	42
3.2 Diseño.....	43
3.2.1 Modelo del diseño.....	44
3.2.2 Diagramas de Clases del Diseño.....	44
3.2.3 Diagramas de Interacción.....	51
3.2.4 Descripción de clases del diseño.....	51
3.3 Diseño de la Base de Datos.....	55
Conclusiones.....	55
<b>CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBA.....</b>	<b>56</b>
Introducción.....	56
4.1 Modelo de Implementación.....	56
4.1.1 Diagrama de Despliegue.....	56
4.1.2 Diagrama de Componentes.....	56
4.2 Modelo de Prueba.....	61
4.2.1 Pruebas de Caja Negra.....	61
Conclusiones Parciales.....	67
<b>CONCLUSIONES GENERALES.....</b>	<b>68</b>
<b>RECOMENDACIONES.....</b>	<b>69</b>
<b>BIBLIOGRAFÍA REFERENCIADA.....</b>	<b>70</b>
<b>BIBLIOGRAFÍA CONSULTADA.....</b>	<b>72</b>
<b>GLOSARIO DE TÉRMINOS.....</b>	<b>73</b>

### **INTRODUCCIÓN**

A través de la historia, el ser humano ha hecho uso de disímiles materiales capaces de liberar energía durante su combustión, con el fin de satisfacer las necesidades propias de su entorno. Destacándose el petróleo como combustible fósil, el cual marcó el inicio de una nueva etapa de desarrollo para la humanidad y una nueva técnica para la generación de energía. Sus características de ser un recurso agotable y no renovable, su distribución desigual en el planeta, la dependencia que ocasionan para el desarrollo de la sociedad, los productos derivados del mismo y sus altos precios en el mercado internacional, lo convierten en una fuente de conflictos entre las naciones.

En la actualidad el límite de sus reservas está cada vez más cercano. Los nuevos descubrimientos de yacimientos se han reducido drásticamente en las últimas décadas haciendo insostenible por mucho tiempo los elevados niveles de extracción, se espera que incluso esos yacimientos entren en declive hacia el 2010 - 2011, lo que provocaría que toda la producción mundial disminuya irremediablemente, conduciendo a la mayor crisis energética que haya sufrido el mundo industrializado.

Cuba no está exenta de esta situación, menos ahora, en medio de la crisis económica de carácter mundial, de consecuencias y duración prácticamente impredecibles, donde los precios del crudo y sus derivados alcanzan valores astronómicos, sumado al férreo bloqueo llevado a cabo por el poder hegemónico del imperio. Por tales razones se han ido tomando un número considerable de medidas en aras de ahorrar la mayor cantidad de energía posible, dando pasos de avance en la búsqueda de soluciones propias que permitan alcanzar la invulnerabilidad energética.

En la Universidad de las Ciencias Informáticas (UCI), con el fin de aplicar los grandes avances de las tecnologías de la información y las telecomunicaciones, en la búsqueda de soluciones que le permitan informatizar los procesos de gestión de la información del combustible, se llevará a cabo este proyecto; con el propósito de integrar a los departamentos de: Transporte, Energía, Mantenimiento y Cajero Central; en los cuales hasta el momento el control del combustible se lleva a cabo mediante planillas y modelos Excel, conllevando a la creación de grandes volúmenes de información dificultando, la búsqueda, el manejo de datos y la generación de reportes. Para dar solución a este problema se opta por establecer un intercambio de información entre los distintos departamentos, y así eliminar el uso del correo electrónico como medio de comunicación, por lo que se decide elaborar una aplicación web, sostenida bajo la licencia de software libre y en la cual tenga acceso solamente el personal autorizado.

Tras el estudio de un trabajo de diploma desarrollado en cursos anteriores que abarca el análisis y diseño de los procesos desarrollados en el Departamento de Transporte se decide, por el surgimiento de nuevas reestructuraciones en el manejo de la información dentro de los modelos y planillas y la necesidad de integrar a los procesos desarrollados en los demás departamentos, realizar esta aplicación para la cual se propone como **problema científico**: ¿Cómo informatizar los procesos de gestión de información del combustible en la Universidad de las Ciencias Informáticas?

Comprende como **objeto de estudio**: los procesos de gestión de información del combustible en la Universidad de las Ciencias Informáticas (UCI). Abarcando como **campo de acción** la informatización de los métodos y modelos utilizados en la gestión de información del combustible en la Universidad de las Ciencias Informáticas.

### **Objetivo general:**

Desarrollar una aplicación web que permita la gestión de información del combustible en la Universidad de las Ciencias Informáticas.

### **Objetivos específicos:**

- Determinar las tecnologías, lenguajes y herramientas que serán utilizadas.
- Definir las funcionalidades de la aplicación a través del Modelo de Negocio y levantamiento de requisitos.
- Realizar el análisis y diseño de acuerdo con las funcionalidades definidas.
- Desarrollar la implementación de las funcionalidades de la aplicación.
- Comprobar el funcionamiento de la aplicación mediante la realización de pruebas.

Partiendo de la **hipótesis**: si se realiza la “Aplicación Web para la Gestión de Información del Combustible en la Universidad de las Ciencias Informáticas”, se logrará una mayor eficiencia y un correcto manejo de la información.

**Variable Independiente**: Aplicación Web para la Gestión de Información del Combustible en la Universidad del las Ciencias Informáticas.

### **Variable Dependiente:**

- Mayor eficiencia.
- Correcto manejo de información.

La operacionalización de variables se encuentra en el Anexo I.

## Tareas de la Investigación:

- Caracterizar el estado actual de los procesos de gestión de información del combustible, así como su identificación.
- Seleccionar la metodología, lenguajes, tecnologías y herramientas, para el desarrollo de la aplicación web.
- Realizar el estudio de la factibilidad (ver Anexo II).
- Modelación del ciclo de vida del software.
- Informatizar el proceso de almacenamiento de la información, facilitando: la búsqueda y generación de reportes, optimizando así el control estadístico del combustible.
- Realizar las pruebas pertinentes al sistema garantizando el buen funcionamiento de la aplicación.

## Posibles resultados:

- Una Aplicación Web para la Gestión de la Información del Combustible en la Universidad de las Ciencias Informáticas.
- Una base sólida de fundamentos teóricos que permita relacionar el análisis documental y el estado del arte de los procesos de gestión de información del combustible en la Universidad de las Ciencias Informáticas.

El desarrollo del sistema traerá como **beneficios**: una reducción del tiempo empleado en el control y asignación de combustible, como la eficiencia en la búsqueda de información manejada por el usuario, para la creación de informes y reportes dentro de los departamentos que integran los procesos de gestión del combustible en la UCI.

Para llevar a cabo estas tareas y garantizar un desarrollo óptimo de la investigación se emplearán los métodos de investigación científica:

## Métodos Empíricos:

- **Entrevista:** Se utiliza para obtener a través del personal que labora en los distintos departamentos la información referente a cómo se llevan a cabo estos procesos, garantizando así el perfeccionamiento de la solución propuesta.

## Métodos Teóricos:

- **Análisis y la Síntesis:** Permite mentalmente la ramificación del problema a solucionar en sus múltiples relaciones y componentes para luego sintetizar sus características generales y los vínculos que existen entre ellas.

- **Histórico Lógico:** Es utilizado para analizar el comportamiento del problema de gestión de la información del combustible a lo largo de la historia y así subsanar los errores antes cometidos llegando a un mayor perfeccionamiento.
- **Modelación:** Muestra al cliente un estimado de la solución del problema y conocer si cumple con sus expectativas.
- **Deducción e Inducción:** Se utiliza para obtener a través de una panorámica general, deducciones sobre el proceso de gestión de información del combustible en específico, lo cual a través de observaciones y reflexiones de la realidad ayudan a concretar la hipótesis que de solución a dicho problema.

El presente documento se encuentra estructurado en: introducción, cuatro capítulos, conclusiones, recomendaciones, referencias bibliográficas, bibliografía, glosario de términos y anexos, donde quedan expuestos y cumplidos los objetivos de la investigación. A continuación, una breve descripción del contenido de cada capítulo:

- **Capítulo I. “Fundamentación Teórica”:**

En este capítulo se abordarán los diferentes conceptos y definiciones manipulados en el transcurso de la investigación; se realizará un estudio de los sistemas relacionados con el control del combustible tanto en el ámbito internacional como nacional. Serán seleccionadas las tecnologías, metodologías, herramientas y lenguajes que se utilizarán para la realización de la aplicación.

- **Capítulo II. “Características del Sistema”:**

En este capítulo se describirán las características del negocio y el flujo actual de los procesos involucrados en el campo de acción. Se confeccionará el modelo de negocio del sistema, se realizará el levantamiento de requisitos y la especificación de los casos de uso del sistema.

- **Capítulo III. “Análisis y Diseño del Sistema”:**

En este capítulo se reunirá lo referente al análisis y diseño del software. Estará compuesto por los artefactos que se generan como son: los diagramas de clases del análisis, los diagramas de clases del diseño, los diagramas de secuencia del diseño, la descripción de clases del diseño y el diagrama de base de datos.

- **Capítulo IV. “Implementación y Pruebas”:**

En este capítulo se hará referencia a los flujos de trabajos de implementación y prueba de la aplicación. Estará compuesto por: los diagramas de despliegue y de componentes. Abarcando también todo lo referente al proceso de pruebas a las que el software será sometido para comprobar su correcto funcionamiento en específico las pruebas de caja negra.

# CAPÍTULO I: FUNDAMENTACIÓN TEÓRICA

## Introducción

El presente capítulo recoge una serie de conceptos importantes a los cuales se hará referencia a lo largo del documento. Presentando además una panorámica de la trascendencia e impacto del tema a través del estudio de los sistemas relacionados con el control del combustible tanto internacional como nacionalmente. También abarcará las metodologías, tecnologías, lenguajes y herramientas que serán utilizadas en el diseño e implementación de la aplicación.

## 1.1 Las Tecnologías de la Información y la Comunicación (TIC)

### 1.1.1 Gestión de Información

Es el proceso que se encarga de suministrar los recursos necesarios para la toma de decisiones, así como para mejorar los procesos, productos y servicios de la organización. (1)

### 1.1.2 Software de Gestión

Comprende las aplicaciones diseñadas para sustituir procedimientos comerciales o administrativos; desarrolladas especialmente para adecuarse a los diferentes requerimientos de la organización.

### 1.1.3 Internet

Internet, sus orígenes datan de la década de 1960 a 1970 específicamente en 1962, desde aquel momento ha sido un foco fundamental, para las redes intercomunicadas, donde se han generado diferentes servicios que actualmente se usan y tienen gran futuro en la esfera de la informática, en la actualidad todas las comunidades, compañías y empresas del mundo buscan como punto clave de desarrollo el acceso a Internet. (2)

### 1.1.4 Aplicación web

Se denomina aplicación web a aquellas aplicaciones que los usuarios pueden utilizar accediendo a un servidor web a través de Internet o de una intranet mediante un navegador web. Es una aplicación que se codifica en un lenguaje soportado por los navegadores web (HTML\*, JavaScript\*, Java, asp.net, php, etc.) en la que se confía la ejecución al navegador. Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, así como a la facilidad para actualizar y mantener sin distribuir e instalar software a miles de usuarios potenciales.

## Conceptos fundamentales para aplicaciones web:

La programación web parte de las siglas *WWW\**. Para confeccionar una página web se debe tener en cuenta tres conceptos fundamentales:

**URL\*** (Localizador Uniforme de Recursos).

**HTTP\*** (*Protocolo de Transferencia de Hipertexto*).

**HTML\*** (Lenguaje de Marcado de Hipertexto).

### 1.1.5 Seguridad en Aplicaciones web

La seguridad, se ha hecho importante desde los inicios de la computación, mucho más desde el surgimiento de Internet, el manejo de la información es cada vez mayor y mantener la misma al margen, más que un derecho viene siendo un deber de cada aplicación web. Actualmente se ha incrementado el número de ataques en internet por año a aplicaciones web, y no tan sólo a éstas sino también a computadoras donde están montados los servidores. La mayoría de los ataques son a través de inyección SQL\*, XSS\* y CSRF\*, pero la utilización del *framework\** *Symfony*, invalida la posibilidad de ataques de este tipo.

### 1.1.6 Software Libre

Es el programa desarrollado y distribuido según la filosofía de dar al usuario las siguientes libertades:

**Libertad 0:** ejecutar el programa sea cual sea su propósito.

**Libertad 1:** estudiar el funcionamiento del programa y adaptarlo a sus necesidades, el acceso al código fuente es condición indispensable para esto.

**Libertad 2:** redistribuir copias.

**Libertad 3:** mejorar el programa y luego publicarlo para bien de toda la comunidad, el acceso al código fuente es condición indispensable para esto. (3)

## 1.2 La gestión del combustible como ente fundamental en el ahorro

La gran dependencia del mundo al consumo de energía como factor primario para el desarrollo de la economía de los países radica fundamentalmente en el uso de los combustibles fósiles. Es preciso destacar que el 87% de la energía consumida en el planeta proviene del carbón, petróleo y gas natural (combustibles fósiles); que los países industrializados aunque representan solo el 20% de la población mundial, consumen el 75% de la producción total de energía del planeta. Como factor de vital importancia a la hora de valorar la necesidad de realizar un óptimo control del combustible están sus características de: ser un producto no renovable y el alto contenido de

contaminación que expulsan mediante la combustión a la atmósfera, propiciando así los cambios climáticos que están ocurriendo en el planeta.

La demanda cada vez mayor de energía, unida a la necesidad de cumplir con las exigencias expuestas en el protocolo de Kyoto suscrito en 1997, que entró en vigor en 2005 sobre una reducción del 12% en las principales fuentes de energía (los combustibles fósiles) para el año 2010 y así disminuir el riesgo de cambio climático; se hace imperante la necesidad de lograr la reducción del consumo mediante el uso de energías alternativas y el aumento de la eficiencia energética.

En Cuba se han tomado una serie de medidas con el fin de lograr una mayor eficiencia y control del combustible, mediante la planificación de las actividades y fundamentalmente el ahorro. La universidad no ha quedado exenta en este proceso y aunque ha tomado medidas para contribuir al ahorro, también adoptó como iniciativa la creación de una aplicación web que permita informatizar los procesos de gestión de combustible que en ésta se llevan a cabo, para lograr así una mejor administración de estos recursos y una mayor eficiencia.

### **1.2.1 Gestión y administración del combustible en la UCI**

El proceso de control de combustible en la UCI integra a diferentes departamentos como: Energía, Transporte, Mantenimiento y Cajero Central. La gestión del combustible en la Universidad se realiza en diferentes niveles, un nivel externo que incluye la solicitud y entrega del combustible a la institución y un proceso interno que tiene el objetivo de distribuir el combustible. También se controla el combustible utilizado para el mantenimiento de los grupos electrógenos, el cual se realiza de manera diferente del proceso antes mencionado. Para una mayor organización cada uno de los departamentos pasará a ser un módulo del sistema. A continuación se describe la interacción que existe entre cada uno de ellos y posteriormente el papel que desarrollan vistos individualmente.

#### **Proceso de solicitud y entrega del combustible a la UCI:**

Este proceso se inicia cuando la Universidad y en especial el Departamento de Energía antes de finalizar el año envía a Cupet (Cubapetroleo) la solicitud del plan del próximo año mediante el documento (Plan de Combustible) el cual comprende el consumo real del año anterior, un estimado del año presente y la solicitud del año a planificar; para los diferentes tipos combustibles consumidos; Cupet autoriza o no la demanda del combustible solicitado, en caso de ser aceptada envía un documento a la UCI y otro a Fincimex (Financiera CIMEX) donde especifica la cantidad de combustible otorgado a la institución. Posteriormente la UCI planifica mensualmente el

combustible que va a necesitar, envía a Cupet un documento llamado CDA\*-001 por tipo de combustible a consumir, siendo Cupet el encargado de autorizar el consumo. Con posterioridad Cupet le envía un documento con dicho autorizo a Fincimex y a la Universidad, esta última es la encargada de realizar el pago por el concepto del combustible a consumir para el próximo mes a la institución de Fincimex, en un plazo de 72 horas, en el cual la universidad debe realizar el recargo de las tarjetas.

### **Distribución del combustible dentro de la UCI:**

El proceso de distribución del combustible en la Universidad se inicia cuando el Departamento de Transporte envía la distribución mensual por vehículos al Departamento de Energía, este departamento autoriza el consumo de los vehículos y envía dicha aprobación a Cajero Central, el cual es el encargado de recargar las tarjetas.

### **Distribución del combustible en los Grupos Electrógenos:**

El proceso de asignación del combustible a los grupos electrógenos presenta características peculiares ya que dicho combustible se utiliza de la reserva existente para casos de afectaciones en el circuito eléctrico de la institución, el control de los grupos electrógenos se realiza diariamente, la Universidad envía a la empresa eléctrica los grupos electrógenos que fueron activados y el consumo que estos ocasionaron al finalizar el día, posteriormente la universidad envía un documento mensual que comprende estas afectaciones, la Organización Básica Eléctrica lo comprueba dados los informes anteriores y envía a Cupet la solicitud, para reponer el combustible a la UCI, Cupet autoriza dicho combustible y le envía a la Universidad un documento avalando el uso del combustible demandado, además de hacerle llegar a Fincimex un documento que autorice a la UCI el uso del mismo. Con posterioridad la universidad realiza el pago a Fincimex dando un plazo de 3 días para el recargo de las tarjetas.

### **Procesos desarrollados por módulos:**

**Módulo de Transporte:** En este módulo se llevan a cabo todos los procesos que se desarrollan en el Departamento de Transporte, relacionados tanto, con los datos de los vehículos existentes en la universidad como son: chapa, marca, modelo, índice de consumo, tipo de carro, clasificación; como los datos de los choferes, certificados técnicos, solicitudes de combustible y tipos de equipos. También se abarcan las asignaciones de: propietario a equipo, actividad a carro, combustible a equipo (pesado, ligero) y combustible a ruta.

**Módulo de Energía:** En este módulo se llevan a cabo todos los procesos desarrollados en el Departamento de Energía, siendo este departamento el encargado de gestionar la solicitud y entrega del combustible a la universidad, en lo cual intervienen una serie de modelos como son: plan de combustible, modelos CDA\*\_001 y CDA\*\_002, redistribución de Cupet, estos de vital importancia en la administración de los valores de consumo del combustible dentro del centro.

**Módulo de Cajero Central:** En este módulo se llevan a cabo todos los procesos desarrollados en el Departamento de Cajero Central, los cuales tienen el propósito de administrar las tarjetas de combustible, así como los datos de cada una de ellas, ya sean las otorgadas a transporte o a grupos electrógenos; realizándose también asignaciones de combustible, liquidaciones y entregas de tarjetas.

**Módulo de Mantenimiento:** Este módulo comprende todos los procesos desarrollados en el área de mantenimiento de grupos electrógenos dentro de la universidad, para lo cual es necesario registrar todos los datos referentes a cada uno de ellos, su ubicación y un historial individual que contiene el estado en que se encuentran si fueron activados o puestos en mantenimiento.

Actualmente en la universidad y específicamente en los departamentos encargados de realizar la planificación, control y distribución del combustible, se realiza mediante planillas y modelos Excel, los documentos se envían entre los departamentos mediante correo electrónico; la búsqueda de la información y el acceso a ella es un procedimiento lento y en muchos casos tedioso por el gran volumen de información almacenada. La necesidad de realizar su informatización mediante una aplicación web que permita vincular el trabajo de los departamentos, facilitar la búsqueda y generación de reportes con la información necesaria, unido a la organización y eficiencia que traería consigo para la gestión de información de estos recursos no renovables fueron factores determinantes a la hora de emprender el presente proyecto.

### 1.2.2 Conceptos asociados al dominio del problema

#### **Combustible:**

Elemento susceptible a quemarse en contacto con el oxígeno y alta temperatura. Que culmina su combustión obteniéndose una fuente de calor y un residuo gaseoso.

#### **Combustible Fósil:**

Carbón, petróleo y gas natural. Estos combustibles se derivan de los restos de organismos vivos que existieron en el pasado remoto los cuales mineralizados se extraen del subsuelo con el

objetivo de producir energía mediante la combustión. Los combustibles fósiles son denominados recursos no renovables.

### **Gestión:**

Es el proceso de "planear, organizar, dirigir, evaluar y controlar" (4)

Utilizado en esta aplicación; la Gestión es el proceso de planificación, organización y control del combustible dentro de la universidad.

### **1.3 Sistemas automatizados existentes vinculados al campo de acción**

En el mundo el desarrollo de las tecnologías va aparejado al incremento de sistemas para la automatización de los procesos en las distintas esferas. La necesidad de comprender y analizar la trascendencia de los sistemas informáticos relacionados con la gestión del combustible, en el ámbito internacional y nacional es un factor clave para la realización de un nuevo sistema.

#### **1.3.1 Sistemas Internacionales**

**Software para administrar Flotillas (SoftFlot):** Es un Software para administrar flotillas de cualquier clase de vehículos, como puede ser Transporte Público, Transporte Pesado, Transporte de Carga, AutoTanques Utilitarios, Mudanzas, Automóviles en Arrendamiento, Motocicletas, Maquinaria para Construcción, Montacargas, Retroexcavadoras etc. Este sistema crea una base de datos con toda la información de sus vehículos, llevando un control detallado de costos y presupuestos, llantas, bitácora de combustible, refacciones, control de mantenimientos preventivos y fallas, trámites de gestoría, depreciaciones, logística, liquidaciones, pago a proveedores y administración de contratos. (5)

**Software de gestión de flota y taller (AsFleet):** Permite realizar un seguimiento y control de una flota de vehículos y de los gastos ocasionados por los vehículos, incluye además la gestión del taller propio y externo. Tiene diferentes módulos integrados, como el control de flotilla de barcos, control del sueldo de los trabajadores, entre otros. Entre algunas funcionalidades se encuentra: mantenimientos preventivos o periódicos, compras e intercambios de piezas, taller propio y externo, seguimiento de neumáticos, gastos externos, facturación del cliente, consumo de carburantes, comunicación con petroleras, previsiones de inversión y anomalías. (6)

**Software de gestión de transporte y mercancías (AsTrans):** Permite gestionar todas las áreas estratégicas de una empresa de transporte como: Administración, Dirección y Tráfico. Permite realizar un seguimiento y control de los gastos generados por la flota de vehículos, incluye además la gestión del taller propio y externo. Engloba todo el proceso de transporte de

mercancías en el que se ven involucrados los diferentes departamentos de la empresa, desde la introducción de pedidos o expediciones hasta la transferencia a contabilidad, pasando por las fases de planificación y facturación. Concluido el proceso, el módulo de estadísticas ayuda a la toma de decisiones. Entre sus funcionalidades se encuentran: seguimiento y gestión, cálculo de distancias, coste Km en vacío/carga, hoja de ruta e informes asociados, facturas de ventas, facturas de compras, exportación contable compatible y parametrizable a productos líderes del sector, liquidación de conductores, incidencias, litigios, control de calidad y atención al cliente. (7)

**Gestión de Flotas y Combustible (PetrolCap):** Es un completo sistema de software para la gestión y el control de gasto de combustible y mantenimiento de sucesos periódicos de los vehículos de su flota. Permite multitanque, pedidos automáticos de recargas. Incluye además control y registro de Seguros, Siniestros, Sanciones, Revisión y ajuste de costes, historial de intervenciones a los vehículos, vistas configurables, multitud de informes y listados, exportación de apuntes, etc. Además, avisa de los sucesos periódicos basándose en los gastos de combustible de los vehículos. Cuando estos se anotan, se especifica también los kilómetros con los que se reposta, las horas de trabajo que realizan, y basándose en ellos y su evolución PetrolCap avisará automáticamente cuando vence un suceso relacionado con alguna máquina en cuestión. (8)

**Mantenimiento de vehículos y cargas de combustibles (Motor0):** Es una aplicación web, para llevar de forma online la gestión del mantenimiento de vehículos y el control de las cargas de combustible; también cuenta con una versión para móviles; como idiomas además del español está disponible en alemán e inglés. (8)

Los sistemas antes mencionados tienen entre sus principales funcionalidades gestionar los recursos referentes al transporte con sus características peculiares según el organismo o empresa que lo necesite. De los cuales se tuvieron en cuenta distintas funcionalidades como:

- El control de todos los datos del vehículo (marca, chapa, modelo, índice de consumo propio).
- Llevar un historial del consumo de combustible por vehículo.
- Según los Kms recorridos conocer cuando un carro necesita ser reparado.

En Cuba su utilización se ve imposibilitada por diferentes factores entre los que se puede encontrar que la mayoría de dichos productos deben ser adquiridos mediante la compra donde se obtiene la licencia y los permisos del propietario; también que dada las características de la economía cubana y con esta el manejo de la información; para la gestión de los medios

relacionados con el transporte, así como la necesidad de abarcar otros sectores vinculados al consumo de combustible es necesario el desarrollo de un producto que se adapte a las necesidades del país en este ámbito.

### **1.3.2 Sistemas Nacionales**

En Cuba se han desarrollado sistemas informáticos relacionados fundamentalmente con el control estadístico y financiero, que en algún momento pueden llegar a ser útiles para administrar las finanzas en sectores como el transporte; pero específicamente realizados para lograr la gestión del combustible en las diferentes entidades no existen hasta el momento.

En la universidad en años anteriores se optó por la realización de un software que permitiera automatizar estos procesos mediante una aplicación de escritorio, pero de esta sólo se realizó el análisis y diseño, el cual quedó comprendido en el trabajo de diploma : "Análisis y Diseño de los módulos Tráfico, Taller, Medios Técnicos y Explotación y Control de Tarjetas y Combustible para la gestión de los recursos de la Dirección de Transporte de la Universidad de las Ciencias Informáticas (UCI)", pero debido a modificaciones actuales en cuanto a la administración de estos recursos y la ventaja de integrar a otros sectores consumidores como son los grupos electrógenos, se aprobó la realización de una aplicación web que gestione toda la información relacionada con el control del combustible y vincule a los departamentos que influyen en este proceso, facilitando así su comunicación y trabajo.

### **1.4 Proceso de Desarrollo de Software**

El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo". (9)

#### **1.4.1 Metodologías de Desarrollo de Software**

##### **1.4.1.1 Metodologías Ágiles**

Se consideran metodologías ágiles a las que tienen la capacidad de proveer respuestas rápidas y adaptabilidad a cambios. Estas hacen la entrega del proyecto menos complicada y más satisfactoria tanto para los clientes como para los equipos de entrega.

Entre las metodologías ágiles más destacadas hasta el momento se encuentran:

- XP
- Scrum
- Crystal Clear

**Programación Extrema (XP):** Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico. Esta utiliza una técnica denominada Historias de Usuario para especificar los requisitos del software donde se recogen las principales características que el sistema debe poseer, sean requisitos funcionales o no funcionales. (10)

**SCRUM:** Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Sus principales características se pueden resumir en dos: el desarrollo de software se realiza mediante iteraciones, denominadas *sprints*, con una duración de 30 días, donde el resultado de cada *sprint* es un incremento ejecutable que se muestra al cliente y las reuniones a lo largo del proyecto, entre ellas se destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración. (10)

**Crystal Methodologies:** Se trata de un conjunto de metodologías para el desarrollo de software caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. Donde el desarrollo de software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar. El equipo de desarrollo es un factor clave, por lo que se deben invertir esfuerzos en mejorar sus habilidades y destrezas, así como tener políticas de trabajo definidas en el equipo. (11)

### 1.4.1.2 Metodologías Tradicionales

Se basan en normas provenientes de estándares seguidos por el entorno de desarrollo y presentan cierta resistencia a cambios. Son aquellas con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos y modelado. Algunos ejemplos de estas son:

- RUP (Rational Unified Procces)
- MSF (Microsoft Solution *Framework*\*)

**Proceso Unificado de Desarrollo (RUP):** El Proceso Unificado de Desarrollo fue creado por el mismo grupo de expertos que crearon *UML*, en 1998. El objetivo que se perseguía con esta

metodología era producir software de alta calidad, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuestos establecidos. Esta metodología concibió desde sus inicios el uso de *UML* como lenguaje de modelado. Caracterizándose por ser un proceso dirigido por casos de uso, que avanza a través de una serie de flujos de trabajo (requisitos, análisis, diseño, implementación, prueba), estar centrado en la arquitectura y ser iterativo e incremental. (12)

Además, cubre el ciclo de vida de desarrollo de un proyecto y toma en cuenta las mejores prácticas a utilizar en el modelo de desarrollo de software como son:

- Desarrollo de software en forma iterativa.
- Manejo de requerimientos.
- Utiliza arquitectura basada en componentes.
- Modela el software visualmente.
- Verifica la calidad del software.
- Controla los cambios.

### **Microsoft Solution *Framework*\* (MSF):**

MSF es una metodología desarrollada por Microsoft Consulting Services en conjunto con varios grupos de negocios de Microsoft y otras fuentes de la industria. MSF provee los principios, modelos y disciplinas para un correcto desarrollo de proyectos en cualquier plataforma (Linux, Citrix, Microsoft, Unix). Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas. Teniendo como principales fases: visión y alcances, planificación, desarrollo, estabilización e implantación. Presentando como características su: adaptabilidad, escalabilidad, flexibilidad y tecnología agnóstica. (13)

### **1.4.2 Metodología de Desarrollo de Software seleccionada (RUP)**

Para la realización de este proyecto se selecciona RUP como la metodología de desarrollo de software a utilizar debido a que garantizan la eficiencia en el desarrollo de un software minimizando los riesgos, garantizando la predicción de los resultados, propiciando así la calidad de entrega. RUP es una metodología flexible de procesos de desarrollo de Software, adaptándose con facilidad a las peculiaridades del sistema, brindando guías conscientes y personalizadas. Es una guía de cómo utilizar de manera efectiva UML. Presenta un fácil acceso a una base de conocimiento con guías, plantillas y herramientas para todas las actividades críticas

de desarrollo. Permite seleccionar fácilmente el conjunto de componentes de proceso que se ajustan a las necesidades específicas del proyecto, organizándolos en términos de disciplinas y fases, consistiendo cada una, en una o más iteraciones; provee un entorno de proceso de desarrollo configurable basado en estándares; permite tener claro y accesible el proceso de desarrollo que se sigue y que este sea configurado a las necesidades de la organización y del proyecto. La aproximación iterativa ayuda a mitigar los riesgos en forma temprana y continúa con un progreso demostrable y frecuentes releases ejecutables. Esta metodología ha sido una de las más usadas en el mundo y en particular en la universidad por su organización y las facilidades que brinda para la creación de un proyecto con calidad. Las características más significativas de RUP se muestran en el Anexo III.

### 1.4.3 Lenguaje de Modelación Visual (UML)

Lenguaje Unificado de Modelado fue originalmente concebido por la Corporación Rational Software y tres de los más prominentes metodologistas en la industria de la tecnología y sistemas de información: Grady Booch, James Rumbaugh, e Ivar Jacobson. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OGM (*Object Management Group*). Es un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema. (14)

### 1.4.4 Lenguaje de programación web (PHP)

**Hypertext Preprocessor (PHP):** Para la implementación del sistema se usará PHP versión 5.2, el mismo posee una comunidad de desarrollo muy grande, la cual implementa constantemente mejoras en su código, en la Universidad de las Ciencias Informáticas este lenguaje es utilizado por muchos programadores. PHP es un lenguaje del lado del servidor (esto significa que funciona en un servidor remoto que procesa la página web antes de que sea abierta por el navegador del usuario) especialmente creado para el desarrollo de páginas web dinámicas. Puede ser incluido con facilidad dentro del código HTML\*, y permite una serie de funcionalidades tan extraordinarias que se ha convertido en el favorito de millones de programadores en todo el mundo.

Entre sus características se encuentran: gratuito, de gran popularidad, posee una enorme eficiencia, se integra de forma sencilla con múltiples bases de datos, es versátil y tiene un gran número de funciones predefinidas.

## 1.5 Tendencias y Tecnologías

Para la realización de este proyecto se han realizado estudios e investigaciones acerca de las principales tendencias y tecnologías usadas en la actualidad a lo largo del mundo, seleccionando las que se adaptan a las características de la aplicación a desarrollar y a las capacidades tanto del país como de la universidad proporcionando así un correcto manejo y control de la información del combustible dentro de la UCI.

### 1.5.1 Patrón de Arquitectura Modelo Vista Controlador (MVC)

Modelo Vista Controlador (MVC) es un patrón que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos, aparta esencialmente la lógica del negocio de la lógica de la presentación, factor que posibilita la simplificación del trabajo y el mantenimiento de los sistemas. La correcta implementación de este patrón dispone de tres entidades básicas: el Modelo, la Vista y el Controlador.

**Modelo:** Encapsula los datos y las funcionalidades. Es independiente de cualquier representación de salida y comportamiento de entrada.

**Vista:** Muestra la información al usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.

**Controlador:** Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsación de botones del ratón, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicio para el modelo o la vista.

Este modelo de arquitectura presenta varias ventajas:

- Hay una clara separación entre los componentes de un programa; lo cual permite implementarlos por separado.
- Hay un API\* muy bien definido; cualquiera que use el API\*, podrá reemplazar el Modelo, la Vista o el Controlador, sin aparente dificultad.
- La conexión entre el Modelo y sus Vistas es dinámica; se produce en tiempo de ejecución, no en tiempo de compilación. (15)

## 1.5.2 Entornos de desarrollo

### 1.5.2.1 Framework\* Symfony

El *Framework\* Symfony* presenta como objetivo fundamental la automatización de los patrones más utilizados en la elaboración de aplicaciones web; también se destaca por orientar a los programadores hacia la clasificación de código, establece un estándar de código legible, encapsula operaciones complejas en simples líneas de código, lo que trae consigo un ahorro de

tiempo a la hora de mostrar datos directamente de la base de datos (BD). Este *framework*\* está implementado bajo el lenguaje PHP 5, es compatible con la mayoría de los SGBD existentes como *My SQL*, *PostgreSQL*, *Oracles* y el *SQL\* Server de Microsoft*; otorga facilidades para diferentes plataformas; está basado en el patrón MVC el cual en *Symfony* provee una abstracción de la base de datos permitiendo una mayor facilidad en la obtención de datos, haciendo a la vista y las acciones independientes al SGBD; en la vista posee *helpers* que facilitan el trabajo para los diseñadores en el código HTML\* de la aplicación; aunque es necesario crear una nueva vista, mantiene al modelo y al controlador original; encargándose de mantener alejados a la vista y al modelo de los detalles del protocolo utilizado. Características de *Symfony* Anexo IV. (16)

### 1.5.2.2 Tecnologías Utilizadas en el Desarrollo del Sistema

Para el transcurso de la elaboración del sistema desde el *framework*\* *Symfony* se utilizan diferentes tecnologías que favorecen y facilitan el uso del *Symfony* en la elaboración de cualquier Aplicación web. En entre ellas se destacan:

**Prototype:** *Framework*\* escrito en *Javascript*\* que se orienta a lo sencillo y dinámico de las aplicaciones web, amplía las posibilidades de programar en ese lenguaje, ofrece muy buenos mecanismos para la manipulación del DOM\* (Modelo de Objetos del Documento) optimizando la obtención de objetos, clases CSS\* (Hojas de Estilo en Cascada) permitiendo un lenguaje tan conciso como el PHP. (16)

**AJAX** (*Asynchronous Javascript\* and XML\**): Un nuevo avance de la tecnología con el objetivo de incrementar el intercambio de datos con el servidor, con la ayuda de *Prototype* se hace de una forma más simple el poder actualizar datos de la aplicación sin necesidad de refrescarla y con pocas líneas de código se puede hacer esto posible. (16)

**Scriptaculous:** Librería de efectos visuales que implementado en AJAX ofreciendo un ambiente visual más amigable con el usuario de forma que cada vez más se vincule a efectos visuales más complejos, y puede ser utilizado con pocas líneas de código. (16)

**Propel:** Herramienta elaborada en *software* (programa) libre y utilizado para el mapeo de los objetos en la base de datos más conocido como ORM (Mapeo Objeto-Relacional), ofrece una de las mejores abstracciones de objetos/relaciones. Permite que los datos del modelo se generen automáticamente, en casos de restricciones de llave foránea crea métodos o funcionalidades especiales para modificar o acceder a los mismos, también una persistencia para los objetos y un servicio de consultas. Todas estas ventajas favorecen la vinculación de un programador a la hora de programar en *Symfony*. (16)

### 1.5.3 Servidor Web Apache 2.2.6

Apache es considerado el más popular servidor HTTP\* en uso, su primera versión fue creada por Robert McCool, en la actualidad es desarrollada por la comunidad de desarrolladores auspiciada por *Apache Software Foundation*. Esta aplicación es compatible con sistemas operativos como Unix, GNU, FreeBSD, Linux, Solaris, Novell NetWare, Mac OS X, Microsoft Windows, OS/2, TPF, y eComStation. Se caracteriza por ser un programa gratuito de código abierto. Para la realización de este proyecto se utilizará Apache 2.2.6 caracterizado por: la flexibilidad que le otorga su estructura modular, cuenta con filtros inteligentes, soporte para archivos de gran tamaño, una mayor seguridad, estabilidad, etc.

### 1.5.4 Librería EXTJS

La programación con la librería ExtJS, que extiende la librería YUI e integra AJAX, *Prototype* y *Scriptaculous*. Con el tiempo se convirtió en un *framework*\* independiente y a principio de 2007 se crea una compañía para comercializar y dar soporte al ExtJS.

ExtJS es neutral al lenguaje que se use en el servidor. Siempre que el resultado se envíe a la página en el formato adecuado, ExtJS no se preocupará de lo que pase en el servidor. Hay docenas de *widgets* (pequeña aplicación o programa) a escoger en ExtJS, incluyendo composiciones automáticas de páginas, pestañas, menús, barras de herramientas, diálogos, vistas en árbol. Proporciona un selector de nodos DOM\* extremadamente poderoso llamado *DomQuery* (puede usarse como una librería independiente, pero en el contexto de ExtJS se usará para seleccionar elementos para poder interactuar con ellos a través de la interfaz *Element*, contiene mucho de los métodos y propiedades de DOM\* que se necesitará proporcionando una interfaz conveniente, unificada y multinavegador).

La versión 3.0 de esta librería incluye varias características que la hacen única, como son: el uso de *scrolling tabs* (Pestañas desplazables) y árboles cuyos nodos pueden contener columnas y la posibilidad de crear fácilmente gráficas de barras, lineares y de pastel a partir de un *store*. ExtJS tiene dos tipos de licencias, LGPL\* y comercial. (17)

## 1.6 Herramientas Utilizadas

### 1.6.1 Sistema Gestor de Base de Datos (SGBD)

Se puede definir como todo el conjunto de herramientas que acompañan al motor de base de datos\* (BD) y permiten a los diferentes tipos de usuarios interactuar con la BD. El motor de BD es la arquitectura lógica o mejor dicho el algoritmo que permite acceder a los datos. Los SGBD

también se conocen como sistemas administradores de BD con su sigla en inglés DBMS (*Data Base Management System*).

### 1.6.1.1 MySQL

Quizás el SGBD más popular hoy en día para la web, su principal característica es la velocidad aunque carece de grandes posibilidades de administración. Tiene como interfaces SQL92, ODBC, C, Perl, JAVA, C++, Python. Sistemas Operativos BSDOS, SunOS, Solaris, Linux, IRIX, AIX, OSF1, BSD/OS, FreeBSD, Windows. Es un sistema multihilo y multiusuario con más de seis millones de instalaciones en todo el mundo. Fue desarrollado sobre la filosofía de software libre. Fue implementado por la empresa MySQL AB, que también le brinda soporte. Principales ventajas de MySQL: (18)

- Diseñado con el objetivo de aumentar la velocidad.
- Consume muy pocos recursos de CPU y memoria lo que proporciona un alto rendimiento.
- Tamaño del registro sin límite.
- Buena integración con PHP.
- Utilidades de administración (PhpMyAdmin).
- Buen control de acceso usuarios-tablas-permisos.

**Desventajas:** Con el propósito de lograr un bajo consumo de CPU y memoria le fueron eliminadas algunas funcionalidades.

### 1.6.1.2 Oracle

Es un sistema de gestión de bases de datos relacionales, surge a finales de los 70 bajo el nombre de *Relational Software* (Programa Relacional) a partir de un estudio sobre *Sistemas Gestores de Bases de Datos*. La tecnología Oracle se encuentra prácticamente en casi el 98% de las empresas líderes del mundo, debido a que su sistema es 100% online en todos sus productos. Las principales ventajas que presenta Oracle son:

**Soporte de transacciones:** Consiste en la interacción de varios procesos que se han de aplicar uno después que el otro, para así lograr la equivalencia a una interacción atómica, es decir, que la acción y reacción sea observable al mismo tiempo.

**Estabilidad:** Posee un nivel de fallos bastante bajo o incluso nulo lo que involucra mayor seguridad.

**Escalabilidad:** Su rendimiento es mejorado después de haberle añadido más capacidad, proporcionalmente a la capacidad añadida.

**Es multiplataforma:** Es soportado por múltiples plataformas, tal como Windows, Linux.

Una de las **desventajas** de Oracle es su precio elevado en versiones *Standard* (Estándar). (19)

### 1.6.1.3 PostgreSQL

Un SGBD objeto-relacional orientado principalmente a bases de datos grandes, tiene muy buen soporte de administración y auditoría. Tiene como interfaces SQL\*, C API\*, C++ API\*, Tcl API\*, Perl5 API\*, Python API\*, WWW\* Gateway, JDBC driver, X11. Sistemas Operativos Linux, Solaris, Digital Unix, BSD, Windows. Es distribuido bajo licencia BSD\* y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Es multiplataforma, de libre copia y distribución bajo licencia BSD\*.

### 1.6.1.4 Sistema Gestor de Base de Datos Seleccionado

Para la realización de este proyecto se seleccionó como SGBD PostgreSQL debido a sus características y fundamentalmente por ser distribuido bajo la licencia BSD\* de *software* libre, ser el sistema gestor de base de datos de código abierto más potente del mercado, siendo estos aspectos significativos e imprescindibles para el desarrollo y despliegue de un *software* en Cuba debido a sus peculiaridades. También es preciso destacar que presenta prestaciones muy similares a las ofrecidas por Oracle, pero con un consumo de recursos inferior (espacio en disco y memoria), lo que hace que el rendimiento de la aplicación no se vea afectado en forma considerable en caso de implementar el SGBD y la aplicación en la misma máquina. A continuación se muestran algunos rasgos distintivos de este SGBD que confirman su utilidad en la realización del sistema de propuesto.

Se utilizará la versión PostgreSQL 8.4, esta versión se caracteriza por obtener mejoras en cuanto a la administración, consulta y programación en PostgreSQL mucho más fácil que nunca, presenta 293 funcionalidades nuevas o mejoradas de la versión 8.4; la mayoría de estas funcionalidades son herramientas y órdenes de administración y monitoreo. (20)

### 1.6.2 NetBeans 6.5

Esta versión tiene nuevos soportes para lenguajes tanto web como escritorio. Importante recalcar que es un *software* libre, multiplataforma, esta versión es más rápida a la hora de compilar,

permite un completamiento de código más extenso al estar utilizando el *Symfony* como *framework\**, ofrece ventajas técnicas y económicas para todo aquel que desee usarlo, presenta un *plugin* para el desarrollo de la tecnología AJAX, presenta una interfaz amigable y sobre todo permite depurar el código realizado.

### 1.6.3 Visual Paradigm for UML 6.4 Release Notes

Visual Paradigm es una herramienta de modelado que utiliza el lenguaje de modelado estándar UML, permite la generación de códigos e ingeniería inversa. Con una clase de diseño bien especificada, Visual Paradigm puede generar código hasta en 15 lenguajes de programación entre ellos PHP que es el que se utilizará en el desarrollo del sistema. Visual Paradigm actualmente cumple con las políticas de migración a Software Libre en Cuba, ya que es una herramienta multiplataforma que se puede utilizar tanto en Linux como en Windows. 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 además 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. Visual Paradigm para UML facilita a los desarrolladores de *software* una herramienta vanguardia para crear aplicaciones de elevada calidad, rápidas y más baratas. Brinda una mejor interfaz gráfica al usuario. (21)

### Conclusiones Parciales

- Se afirmó a través de un análisis exhaustivo de sistemas utilizados para el control del combustible dentro del ámbito internacional y nacional, la necesidad de crear una aplicación web que gestione la información del combustible en la Universidad de las Ciencias Informáticas (UCI).
- Se seleccionaron para el desarrollo de la aplicación como: metodología de desarrollo de software RUP, UML como lenguaje de modelado, lenguaje de programación: del lado del servidor PHP, del lado del cliente JS, de base de datos\* PostgreSQL, el *framework\** Symfony, la librería EXTJS y las herramientas: Visual Paradigm6.4 (modelado), PostgreSQL 8.4 (SGBD), NetBeans 6.5 y Apache 2.2.6.

# CAPÍTULO II: CARACTERÍSTICAS DEL SISTEMA

## Introducción

El presente capítulo abarcará una descripción detallada de los procesos del negocio, donde serán identificados: actores, casos de uso, trabajadores y reglas del negocio, elaborados: los diagramas de casos de uso del negocio y el modelo de objetos. Lo cual sentará las bases para la definición de los requerimientos funcionales y no funcionales del sistema, indispensables para el posterior desarrollo de los casos de uso y el modelo de casos de uso del sistema.

### 2.1 Modelo de negocio

El Modelamiento del Negocio constituye el primer flujo de trabajo a desarrollar según la metodología de desarrollo de software (RUP). Este “describe los procesos de negocio de una entidad en términos de casos de uso y actores del negocio que se corresponden con los procesos del negocio y los clientes respectivamente.” (22).

#### 2.1.1 Procesos del Negocio

“Un proceso de desarrollo de software es una definición del conjunto completo de actividades necesarias para convertir los requisitos de usuario en un conjunto consistente de artefactos que conforman un producto software, y para convertir los cambios sobre esos requisitos en un nuevo conjunto consistente de artefactos.” (22)

Proceso	Descripción
Realizar el Plan de Combustible	Este proceso es realizado por el Departamento de Energía con el fin de enviar a Cupet la solicitud de combustible de la UCI para un próximo año. Se confecciona a partir de un modelo que contiene: el consumo real del año anterior, un estimado del año presente y la propuesta de plan para el próximo año.
Realizar Modelo CDA*_001	Este proceso se realiza en el Departamento de Energía, su objetivo es obtener mediante un modelo las relaciones existentes entre las actividades y los valores de consumo de cada una de ellas, tanto acumulado de meses anteriores, como un estimado del mes actual y la demanda del próximo mes; generando así el índice de consumo de la demanda acumulada y estimada.
Realizar Redistribución de Cupet	Este proceso es llevado a cabo en el Departamento de Energía para seleccionar los posibles medios de pagos de combustible, la cantidad por tipos de combustible y los diferentes destinos que pueden tener estos en cada entidad. Este modelo es enviado finalmente a Cupet.

Realizar Solicitud de Combustible.	Este proceso se lleva a cabo en el Departamento de Transporte con el fin de realizar las solicitudes de combustible mensual ya sea por: equipos (pesados o ligeros), ruta y extra al Departamento de Energía.
Realizar Historial de Grupo Electrónico	Este proceso se desarrolla en el Departamento de Mantenimiento de Grupos Electrónicos con el fin de controlar el estado de los grupos electrónicos, destacándose: el tiempo que fueron utilizados, su consumo y si están en mantenimiento o necesitan de este.
Realizar Entrega de Tarjetas.	Este proceso se desarrolla en el Departamento de Cajero Central con el objetivo de realizar la entrega de tarjetas al Departamento de Transporte ya sea para equipos (pesados o ligeros), rutas y extras, asignándole una cantidad de combustible mediante una tarjeta a un vehículo determinado.

Tabla 1: Procesos del Negocio

### 2.1.2 Entidades del Negocio

Las entidades de negocio representan a los objetos que los trabajadores del negocio toman, inspeccionan, manipulan, producen o utilizan durante la realización de los casos de uso de negocio.

Entidades del Negocio	Descripción
Plan de Combustible por Actividades	Es un modelo mediante el cual se solicita el combustible anual a consumir por la universidad en relación con las actividades que en la misma se desarrollan, contiene los datos de consumo real del año anterior, un estimado del año presente y la propuesta del plan para el próximo año, mediante los valores obtenidos en cada año se calcula el índice de real de demanda acumulada y el índice real de demanda estimada.
Modelo CDA*_001	Este modelo se realiza mensualmente, para cada producto en específico ya sea gasolina o diesel, contiene las relaciones existentes entre las actividades y el nivel de consumo del mes anterior, un estimado del presente mes, y la demanda para el próximo; Este modelo es elaborado por el Departamento de Energía para llevar un balance del consumo de combustible mensual.
Redistribución de Cupet	Es un modelo estándar para todas las empresas de Cuba el cual contiene los posibles medios de pago de combustible, en este caso la universidad compra a Fincimex con moneda nacional, la cantidad por tipos de combustible (gasolina, diesel, GLP) solicitada y su destino

	que puede ser: privado, compensado y estatal. También muestra el combustible distribuido por tiro directo a las diferentes provincias.
Solicitud de Combustible	Es un modelo que se realiza en el Departamento de Transporte donde solicita la cantidad de combustible que necesita por actividades, organizándolas según sus equipos (pesados o ligeros), rutas o extra.
Modelo de Entrega de Tarjetas	Es un modelo que se realiza en el Departamento de Cajero Central con el propósito de recoger los datos de los equipos (pesados o ligeros), rutas o extra a los cuales le serán asignados las tarjetas de combustible, sus datos y la cantidad de las disponen.
Historial de Grupo Electrónico	Cada grupo electrónico presenta un historial el cual reúne todos sus datos, la fecha en que fue utilizado o puesto en mantenimiento, así como las horas en que estuvo activo, la demanda liberada y la energía producida.

Tabla 2: Entidades del Negocio

### 2.1.3 Actores del Negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externo; con los que el negocio interactúa.

Actor del Negocio	Descripción
Cupet	Es la compañía relacionada con la extracción de los depósitos de petróleo, refinado y la distribución del petróleo y sus derivados en el país. Cupet es el encargado de solicitar el plan de combustible a la universidad y autorizarlo.
Departamento de Transporte	Es el departamento al cual se hace entrega del combustible asignado a la universidad y se encarga de distribuirlo según los vehículos y las actividades que desarrolla.
Departamento de Energía	Es el departamento en el cual se realiza el proceso de solicitud y entrega de combustible. Él solicita a Transporte el combustible que va a utilizar cada mes según las actividades y vehículos del que dispone.
Organización Básica Eléctrica(OBE)	La organización básica eléctrica es la encargada de solicitar los historiales de grupos electrónicos, así como los datos y ubicación de cada uno de ellos para llevar los registros de consumo y mantenimiento de cada uno de ellos en la universidad.

Tabla 3: Actores del Negocio

### 2.1.4 Trabajadores del Negocio

Un trabajador del negocio es una abstracción de una persona (o grupo de personas), una máquina o un sistema automatizado; que actúa en el negocio realizando una o varias actividades, interactuando con otros trabajadores del negocio y manipulando entidades del negocio. Representa un rol.

Trabajador del Negocio	Descripción
Especialista de Energía	Es el encargado de confeccionar los modelos de Plan de Combustible, CDA*_001 y Redistribución de Cupet, para la planificación del combustible dentro del Departamento de Energía.
Director de Gestión Energética	Es el encargado de revisar los modelos de Plan de Combustible, Redistribución de Cupet y CDA*_001, elaborados por el Especialista de Energía.
Vicerrector de Economía	Es el encargado de revisar y aprobar los modelos elaborados para la solicitud y asignación del combustible dentro de la universidad.
Especialista de Combustible	Es el encargado de confeccionar los modelos de Solicitud de Combustible que comprende el consumo de combustible por las actividades a solicitar en cada mes.
Director de Transporte	Es el encargado de revisar todos los documentos generados en el Departamento de Transporte como: el modelo Solicitud de Combustible.
Especialista de Cajero Central.	Es el encargado de realizar las entregas de Tarjetas de Combustible a Transporte y administrar los datos relacionados con ellas.
Director de Mantenimiento de Grupos Electrógenos	Es el encargado de administrar los datos de los grupos electrógenos, su ubicación y la actualización de los historiales por cada uno de estos.

Tabla 4. Trabajadores del Negocio

### 2.1.5 Diagrama de casos de uso del negocio

Un diagrama de casos de uso del negocio representa gráficamente los procesos del negocio y su interacción con los actores del negocio.

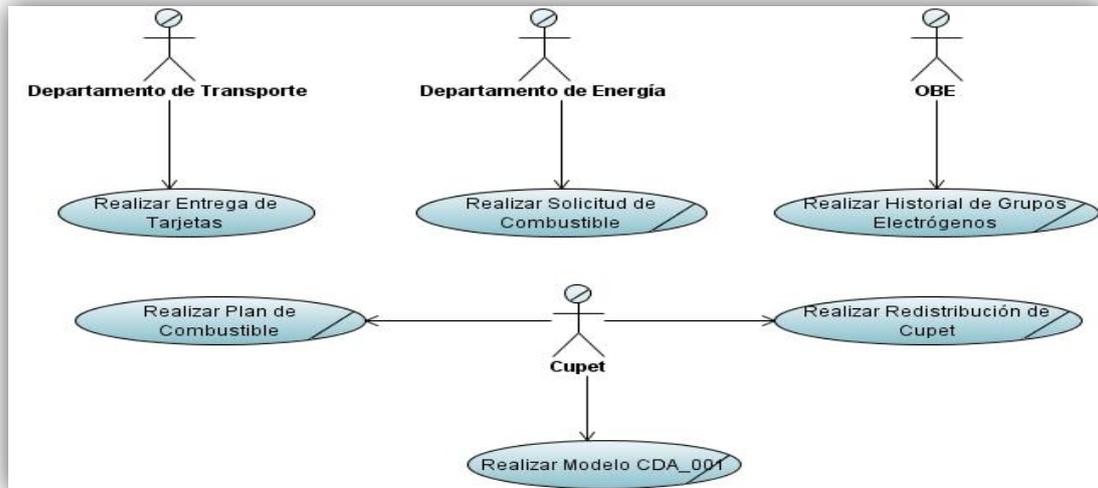


Figura 1: Diagrama de Casos de Uso del Negocio

### 2.1.6 Descripción de Casos de Uso del Negocio

Caso de Uso del Negocio	
<b>Caso de Uso</b>	Realizar Plan de Combustible.
<b>Actor</b>	Cupet
<b>Trabajador</b>	Especialista de Energía, Director de Gestión Energética, Vicerrector de Economía.
<b>Resumen</b>	El caso de uso se inicia cuando Cupet solicita la demanda de combustible del próximo año. El Especialista de Energía es el encargado de crear el plan de combustible anual, el Director de Gestión Energética es el responsable de revisarlo y el vicerrector de economía es quien aprueba este antes de ser enviado a Cupet. Este caso de uso termina cuando Cupet aprueba el plan de combustible.
<b>Precondiciones</b>	
Flujo Normal de Eventos	
Sección 1	
Acción del Actor	Respuesta del Negocio
1. Cupet solicita la creación del plan de combustible anual.     5. Cupet recibe el modelo del plan de combustible anual y puede aprobar o denegar este.	2. El Especialista de Energía confecciona el modelo del plan de combustible anual teniendo en cuenta: las actividades, el índice de consumo del año anterior, el estimado de año presente y la demanda del próximo año. 3. Este modelo es enviado al Director de Gestión Energética el cual lo revisa y envía a vicerrector de economía. 4. El vicerrector de economía aprueba este modelo y autoriza su envío a Cupet.

<b>Postcondiciones</b>	En caso de que Cupet deniegue el plan de combustible solicita la creación de un nuevo plan. Si lo aprueba continúa el proceso de demanda de combustible.
------------------------	--

Tabla 5. CUN\_ Realizar Plan de Combustible

<b>Caso de Uso del Negocio</b>	
<b>Caso de Uso</b>	Realizar Modelo CDA*_001
<b>Actor</b>	Departamento de Energía
<b>Trabajador</b>	Especialista de Energía, Director de Gestión Energética, Vicerrector económico
<b>Resumen</b>	El caso de uso se inicia cuando Cupet solicita al Departamento de Energía el Modelo CDA*_001. El Especialista de Energía es el encargado de confeccionar el modelo, enviarlo al Director de Gestión Energética para que lo revise y luego al Vicerrector de Economía para que lo apruebe antes de ser enviado a Cupet.
<b>Precondiciones</b>	Tiene que estar aprobado el plan de combustible anual.
<b>Flujo Normal de Eventos</b>	
<b>Sección 1</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
1. Cupet solicita al Departamento de Energía el modelo CDA*_001.	2. El Especialista de Energía confecciona el modelo CDA*_001, donde se registran las actividades y el consumo mensual por cada una de ellas, tanto el acumulado de meses anterior, como un estimado del mes presente y la demanda del próximo mes, obteniendo así el índice de consumo real y estimado. Terminado el modelo lo envía al Director de Gestión Energética. 3. El Director de Gestión Energética es el encargado de revisar el modelo y enviarlo al Vicerrector de Economía para su aprobación. 4. El Vicerrector de Economía es la persona autorizada a aprobar el modelo CDA*_001 y enviarlo al Departamento de Energía para los

5. Cupet recibe el modelo CDA*_001.	trámites de solicitud de combustible a Cupet.
<b>Postcondiciones</b>	

Tabla 6. CUN\_Realizar Modelo CDA\*\_001

<b>Caso de Uso del Negocio</b>	
<b>Caso de Uso</b>	Realizar Redistribución de Cupet
<b>Actor</b>	Cupet
<b>Trabajador</b>	Especialista de Energía, Director de Gestión Energética, Vicerrector de Economía.
<b>Resumen</b>	El caso de uso comienza cuando Cupet solicita al Departamento de Energía de la UCI, el modelo de redistribución de Cupet. El Especialista de Energía confecciona el modelo y se lo envía al Director de Gestión Energética, el cual lo revisa y se lo envía al vicerrector de economía el cual es el encargado de aprobarlo para ser enviado posteriormente a Cupet.
<b>Precondiciones</b>	Para la realización de este modelo debe estar aprobado el plan de combustible anual al igual que el CDA*_001 mensual.
<b>Flujo Normal de Eventos</b>	
<b>Sección 1</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
1. Cupet solicita la creación del modelo redistribución de Cupet.	2. El Especialista de Energía confecciona el modelo de redistribución de Cupet solicitado donde se registran: los medios de pago, la cantidad por tipos de combustible y el destino del mismo. Después de confeccionado el modelo este es enviado al Director de Gestión Energética.
	3. El Director de Gestión Energética es el encargado de revisar el modelo de redistribución de Cupet y lo envía al vicerrector de economía.
5. Cupet recibe el modelo de redistribución y autoriza la entrega de combustible a la UCI.	4. El vicerrector de economía es la persona autorizada para aprobar este modelo y permitir su envío a Cupet.
<b>Postcondiciones</b>	

Tabla 7. CUN\_Realizar Redistribución de Cupet

<b>Caso de Uso del Negocio</b>	
<b>Caso de Uso</b>	Realizar Solicitud de Combustible
<b>Actor</b>	Departamento de Energía
<b>Trabajador</b>	Especialista de Combustible
<b>Resumen</b>	El caso de uso comienza cuando el Departamento de Energía solicita el estimado del consumo de combustible para el próximo mes al Especialista de Transporte el cual le envía un modelo con la cantidad que necesita por actividades ya sean realizadas por equipos (pesados o ligeros), rutas o extras. El caso de uso termina cuando el

	Departamento de Energía recibe el modelo.
<b>Precondiciones</b>	
<b>Flujo Normal de Eventos</b>	
<b>Sección 1</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
1. El Departamento de Energía solicita el Modelo de Solicitud de Combustible.	2. El Especialista de Combustible elabora el Modelo de Solicitud de Combustible y se lo envía al Director de Transporte.
	3. El Director de Transporte lo revisa y aprueba, luego lo envía al Vicerrector de Economía.
	4. El Vicerrector de Economía es el encargado de revisarlo para su aprobación final antes de ser enviado al Departamento de Energía.
5. Recibe el Modelo de Solicitud de Combustible.	
<b>Postcondiciones</b>	

Tabla 8: CUN\_ Realizar Solicitud de Combustible

<b>Caso de Uso del Negocio</b>	
<b>Caso de Uso</b>	Realizar Entrega de Tarjetas
<b>Actor</b>	Departamento de Transporte
<b>Trabajador</b>	Especialista de Cajero Central
<b>Resumen</b>	El caso de uso comienza cuando el Departamento de Transporte solicita la entrega de las tarjetas de combustible que le fueron asignadas para su consumo mensual al Departamento de Cajero Central, él almacena los datos de los equipos, rutas o extras a las que fueron asignadas y la cantidad que estas disponían, mediante el Modelo de Entrega de Tarjetas. El caso de uso finaliza cuando el Departamento de Transporte recibe sus tarjetas.
<b>Precondiciones</b>	Para ser asignada las tarjetas de combustible primero debe ser asignado a la universidad mediante el proceso que se realiza en el Departamento de Energía y ser aprobada la solicitud de combustible elaborada por Transporte.
<b>Flujo Normal de Eventos</b>	
<b>Sección 1</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
1. El Departamento de Transporte solicita la entrega de las tarjetas de combustible.	2. El especialista de Cajero Central elabora el modelo de entrega de tarjetas de combustible con los datos de los equipos (pesados o ligeros), rutas o extras, a las que fueron asignadas.
4. Recibe las tarjetas de combustible.	3. El Especialista de Cajero Central entrega las tarjetas de combustible a Transporte.
<b>Postcondiciones</b>	

Tabla 9: CUN\_ Realizar Entrega de Tarjetas

<b>Caso de Uso del Negocio</b>
--------------------------------

<b>Caso de Uso</b>	Realizar Historial de Grupos Electrógenos
<b>Actor</b>	Organización Básica Eléctrica (OBE)
<b>Trabajador</b>	Director de Mantenimiento de Grupos Electrógenos
<b>Resumen</b>	El caso de uso comienza cuando la OBE solicita el historial de los grupos electrógenos para conocer el estado de cada uno de ellos dentro de la universidad, al Director de grupos electrógenos el cual confecciona dicho modelo.
<b>Precondiciones</b>	
<b>Flujo Normal de Eventos</b>	
<b>Sección 1</b>	
<b>Acción del Actor</b>	<b>Respuesta del Negocio</b>
1. La Organización Básica Eléctrica solicita los Historiales de Grupos Electrógenos	2. El Director de Mantenimiento de grupos electrógenos elabora los historiales de grupos electrógenos, describiendo en ellos su trayectoria diaria. 3. El director envía los historiales de grupos electrógenos a la OBE.
4. Recibe los historiales de grupos electrógenos.	
<b>Postcondiciones</b>	

Tabla 10: Caso de Uso Realizar Historial de Grupo Electrónico

Los diagramas de actividades correspondientes a las descripciones de los casos de uso del negocio se encuentran en el Anexo V.

### 2.1.7 Modelo de Objetos del Negocio

El modelo de objetos del negocio es un modelo interno del negocio. Describe cómo cada caso de uso de negocio es llevado a cabo por parte de un conjunto de trabajadores que utilizan un conjunto de entidades y unidades de trabajo. (22)

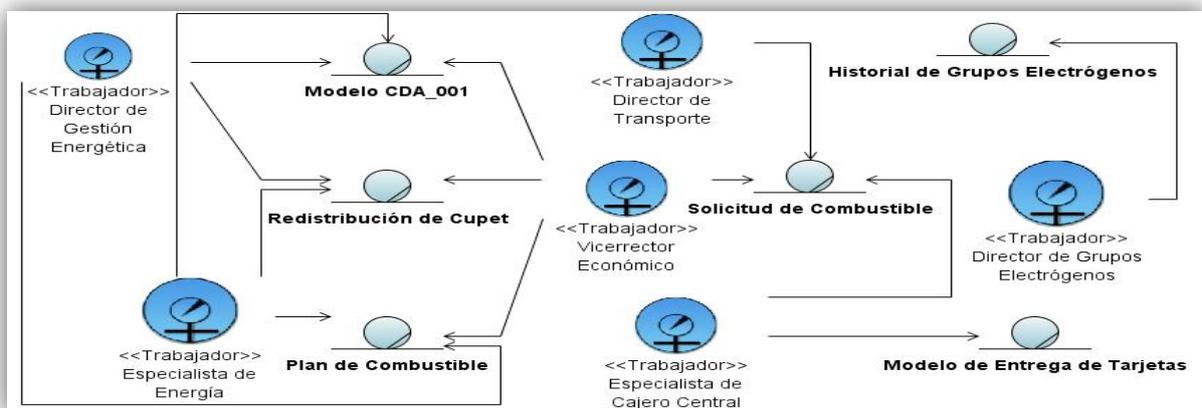


Figura 2: Diagrama de Objetos del Negocio

### 2.1.8 Reglas del Negocio

Las reglas del negocio describen políticas que deben cumplirse o condiciones que deben satisfacerse, por lo que regulan algún aspecto del negocio. Para la creación del sistema se tendrá presente:

- La información debe ser accedida sólo por personal autorizado.
- No pueden existir dos códigos con el mismo valor en el sistema.
- El precio del combustible sólo debe ser dado en MN y CUC.
- No pueden existir dos áreas con el mismo nombre en el sistema.
- No pueden existir dos actividades con el mismo nombre en un mismo plan de combustible o CDA\*\_001.
- No pueden existir dos directivos con el mismo CI (número de carné de identidad) en el mismo sistema.
- No pueden existir dos empresas con el mismo nombre en el sistema.
- No pueden existir dos planes de combustible con el mismo nombre, ni con la misma fecha en el sistema.
- No pueden existir dos tarjetas con el mismo número o pin en el sistema.

### 2.2 Requisitos de Software

Los requisitos de software definen las condiciones o capacidades que necesita el usuario para resolver un problema o lograr un objetivo. Existen dos tipos de requisitos de software, los requerimientos funcionales, que son las capacidades o condiciones que el sistema debe cumplir, y los requerimientos no funcionales, que son las propiedades o cualidades que el producto de software debe tener.

#### 2.2.1 Requerimientos Funcionales

Los requerimientos funcionales del sistema están presentes en el Anexo VI. Destacándose en el documento los requerimientos significativos con los cuales se trabajará a lo largo de todo el documento.

##### Módulo de Acceso

##### **RF\_3: Autenticar Usuario**

3.1: Verificar Privilegios

##### Módulo de Transporte

##### **RF\_12: Gestionar Solicitud de Combustible**

12.1: Mostrar un listado con todas las solicitudes de combustible.

12.2: Adicionar solicitudes de combustible.

12.3: Modificar solicitudes de combustible.

12.4: Eliminar solicitudes de combustible.

### **Módulo de Energía**

#### **RF\_23: Gestionar Plan de Combustible**

23.1: Mostrar un listado con todos los planes de combustible.

23.2: Adicionar plan de combustible.

23.3: Modificar plan de combustible.

23.4: Eliminar plan de combustible.

#### **RF\_38: Asignar Actividad a Plan**

38.1: Buscar plan de combustible.

38.2: Asignar actividad a plan.

38.3: Modificar asignación de actividad a plan.

38.4: Eliminar asignación de actividad a plan.

#### **RF\_43: Generar Reportes**

43.1: Reporte de Consumo por Años.

43.2: Reporte de Altos Consumidores.

43.3: Reporte de Consumo de Actividad\_ Mes.

43.4: Reporte de Consumo de Combustible.

### **Módulo de Cajero Central**

#### **RF\_46: Gestionar Liquidación**

46.1: Mostrar un listado con todas las liquidaciones.

46.2: Adicionar liquidación.

46.3: Modificar liquidación.

46.4: Eliminar liquidación.

### **Módulo de Mantenimiento**

#### **RF\_57: Gestionar Historial de Grupo Electrónico**

57.1: Mostrar un listado con todos los historiales de grupos electrónicos.

57.2: Adicionar historial de grupo electrónico.

57.3: Modificar historial de grupo electrónico.

57.4: Eliminar historial de grupo electrónico.

### 2.2.2 Requerimientos no Funcionales

#### RNF\_1: Usabilidad

Comprende todo lo relacionado con las facilidades de uso que debe brindar el sistema a los usuarios, las cuales incluyen: una interfaz amigable, una vía rápida y efectiva para realizar búsquedas de información, una rápida adaptación a usuarios con poca experiencia. Incluye el empleo de los perfiles de usuarios, para la diferenciación de opciones e interfaces de acceso al sistema mediante diferentes roles.

#### RNF\_2: Seguridad

- **Confiabilidad:** La información manejada por el sistema debe estar protegida de acceso no autorizado y divulgación.
  - Hacer uso del protocolo seguro https\*
  - Codificar las contraseñas mediante el método de encriptación MD5\*
  - Utilizar de los filtros de seguridad que implementa el *framework*\* Symfony.
- **Integridad:** La información manejada por el sistema debe ser objeto de una cuidadosa protección contra la corrupción y estados de inconsistencia.
  - Realizar salvallas periódicas de las base de datos\*.
- **Disponibilidad:** La aplicación deberá estar disponible en todo momento para aquellas personas que tienen acceso a su información. Los mecanismos utilizados para lograr la seguridad no deben ser un obstáculo para los usuarios a la hora de obtener los datos deseados en un momento dado.

#### RNF\_3: Portabilidad

El sistema debe permitir ser ejecutado en diferentes ambientes (ser multiplataforma).

#### RNF\_5: Apariencia o interfaz externa

La interfaz externa deberá ser sencilla con colores suaves a la vista y sin cúmulo de imágenes u objetos que distraigan al cliente del objetivo de su empleo.

#### RNF\_6: Interfaz Interna

La interfaz interna estará determinada por los desarrolladores, construyendo así una vista escalable de las clases o agrupaciones de clases que permitirán un mejor encapsulamiento de las funcionalidades y una mayor abstracción modular del sistema.

#### RNF\_6: Soporte

El sistema debe permitir posteriores modificaciones y actualizaciones a fin de alcanzar un mayor número de funcionalidades, en caso de cambios en el negocio.

### **RNF\_7: Diseño e implementación**

Para desarrollar el análisis y diseño del sistema se utilizará: como metodología RUP, como lenguaje de modelado UML y como herramienta Visual Paradigm. En la implementación del sistema se utilizará como lenguaje de programación PHP, como herramienta NetBeans como *Framework\** Symfony y la librería EXT.

### **RNF\_8: Software**

- Para el cliente: sistema operativo con interfaz gráfica y conexión a red. Navegador web: (Opera, Chrome, Safari y Mozilla FireFox Recomendado).
- Para el servidor: sistema operativo Linux o Windows.
- Servidor web: Apache 2.2.6.
- Gestor de Base de Datos: PostgreSQL 8.4.
- Lenguaje de programación: PHP 5.2.5.
- El *framework\** base de desarrollo que se utilizará es: *Framework\** Symfony 1.2.8.
- Para la implementación: NetBeans 6.5.
- Como SGDB: PostgreSQL 8.4.
- Herramienta para el diseño de la base de datos\*: Visual Paradigm 6.4.
- Herramienta de modelado UML: Visual Paradigm 6.4.

### **RNF\_9: Hardware**

#### **Para el cliente**

Las condiciones mínimas de hardware son: que las computadoras tengan procesador Intel Celeron a 2.80 GHz, superior o equivalente, disco duro de 40 GB y 256 MB de RAM.

Se requiere que las computadoras estén conectadas a la red y de una impresora estándar para la impresión de reportes.

#### **Para el servidor:**

Las condiciones mínimas de hardware son: que las computadoras tengan procesador Intel a 3.2 GHz, superior o equivalente, disco duro de 120 GB y 1 GB de RAM.

### **2.3 Descripción del Sistema Propuesto**

Con el propósito de automatizar los procesos de planificación y control del combustible en la UCI, se propone la realización del análisis, diseño e implementación de un sistema informático que agilice y eleve la calidad del mismo. Dicho software presentará entre sus principales funcionalidades permitir: adicionar, modificar, eliminar y mostrar listado (gestionar), realizar

asignaciones, facilitar la búsqueda de modelos, exportar documentos a formato Excel o pdf, imprimir documentos, generar reportes, ordenar alfabética o numéricamente documentos, etc.

### 2.3.1 Actores del sistema

Cada trabajador del negocio que tiene actividades a automatizar es considerado un candidato a actor del sistema. Si algún actor del negocio va a interactuar con el sistema, entonces también será un actor del sistema.

<b>Actor del Sistema</b>	<b>Descripción</b>
Vicerrector de Economía	Es el encargado de revisar finalmente los documentos desarrollados tanto en el Departamento de Transporte, como en Energía para la solicitud y distribución del combustible en la universidad.
Director de Transporte	Es el encargado de revisar los documentos desarrollados en el Departamento de Transporte, también puede adicionar, modificar, asignar y eliminar cualquier dato o modelo dentro de este departamento.
Director de Gestión Energética	Es el encargado de revisar los documentos desarrollados en el Departamento de Energía; también puede adicionar, modificar, asignar y eliminar cualquier dato o modelo dentro de este departamento.
Especialista de Combustible	Es el encargado de adicionar, modificar, asignar y eliminar los procesos desarrollados en el Departamento de Transporte entre los que se destacan la gestión de: solicitud de combustible, así como la gestión de combustible a equipo ligero, pesado o por ruta.
Especialista de Energía	Es el encargado de adicionar, modificar, asignar y eliminar todos los procesos desarrollados en el Departamento de Energía entre los que se destacan, la gestión de: planes de combustible, modelos CDA*_002 y CDA*_001, asignación de combustible y redistribución de Cupet; así como las asignaciones de: actividad a plan y actividad a modelo CDA*_002 y CDA*_001.
Especialista de Cajero Central	Es el encargado de adicionar, modificar, asignar y eliminar los diferentes procesos que se desarrollan en Cajero Central, entre los que se destacan: la gestión de

	tarjetas, liquidaciones y entregas de combustible, así como las asignaciones de tarjetas a rutas y tarjetas a equipos.
Director de Mantenimiento de Grupos Electrógenos	Es en encargado de adicionar, modificar, asignar y eliminar los diferentes procesos desarrollados en el Departamento de Mantenimiento, entre los que se destacan: la gestión de ubicación, grupos electrógenos e historial de grupos electrógenos.
Administrador	Es la persona que tiene todos los privilegios sobre la aplicación, realizando la gestión de usuario y roles para el acceso a la misma.

Tabla 11: Actores del sistema

### 2.3.2 Diagrama de Casos de Uso del Sistema

#### Módulo de Acceso:

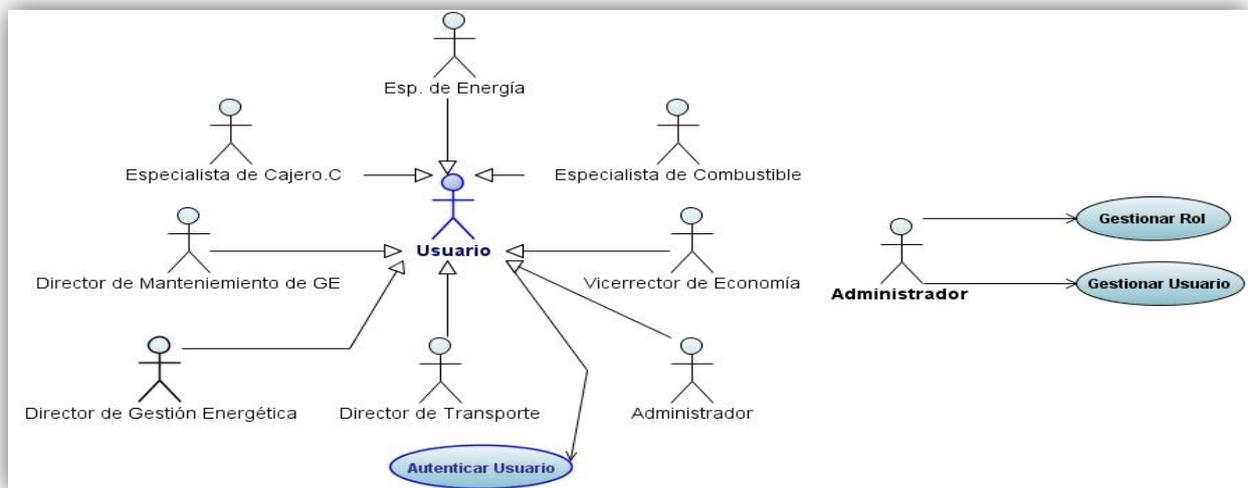


Figura 3: DCUS\_ Módulo de Acceso

## Módulo de Transporte:

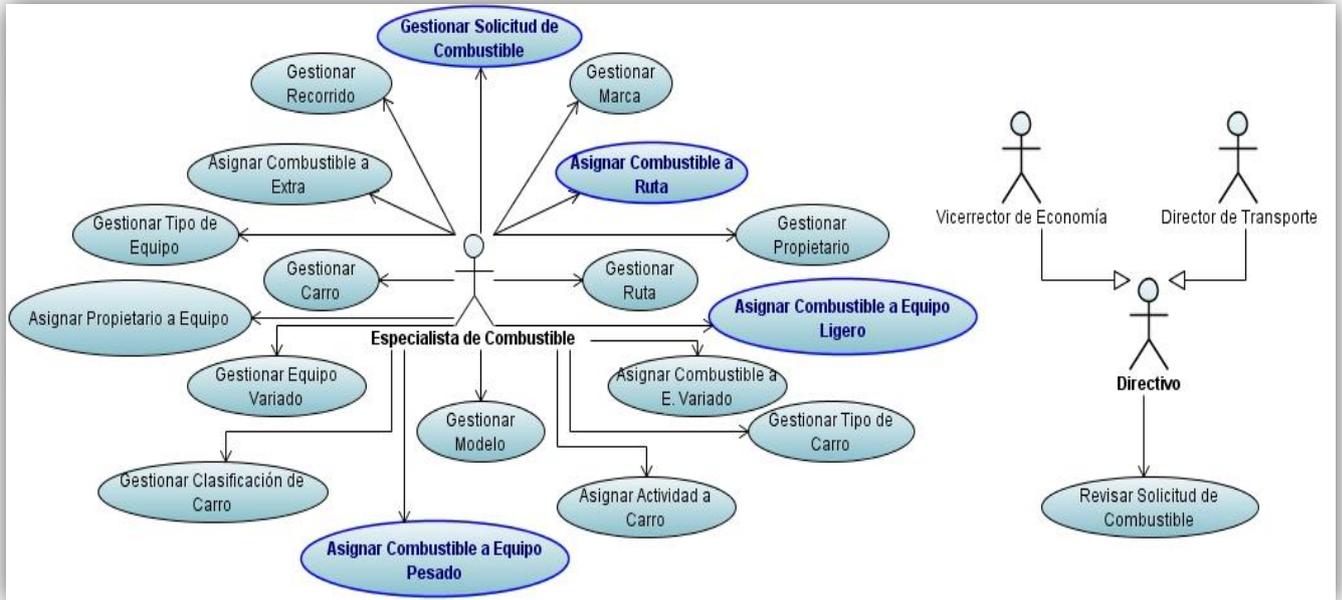


Figura 4: DCUS\_ Módulo de Transporte

## Módulo de Energía:

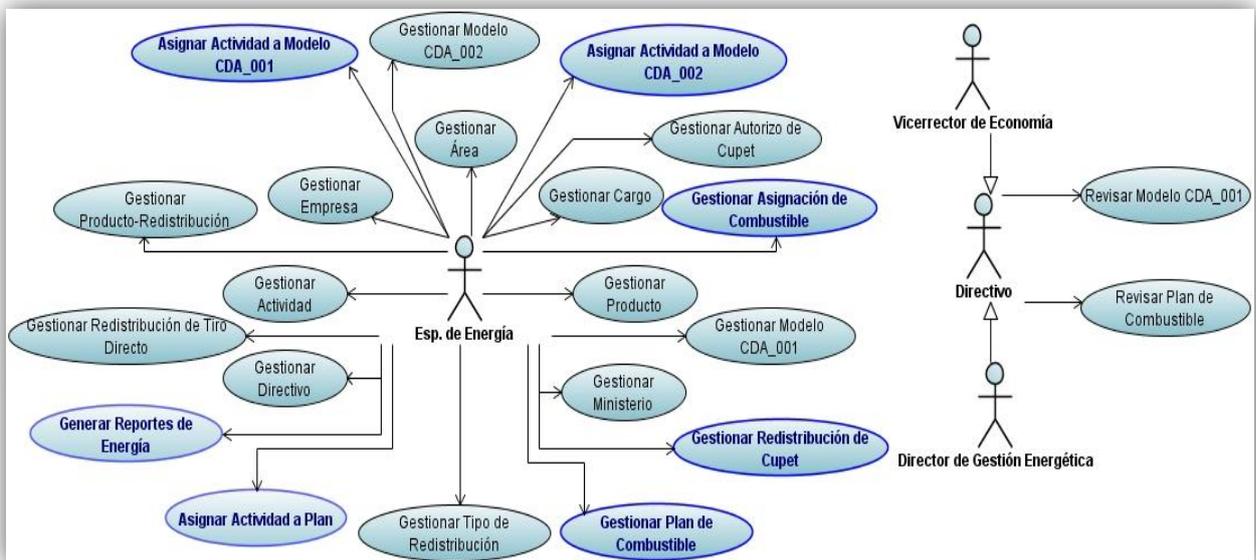


Figura 5: DCUS\_ Módulo de Energía

### Módulo de Cajero Central

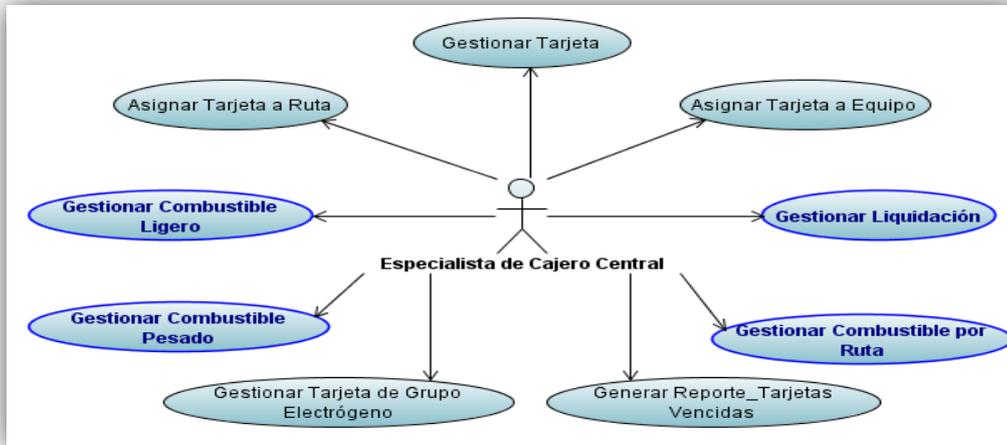


Figura 6: DCUS\_ Módulo de Cajero Central

### Módulo de Mantenimiento

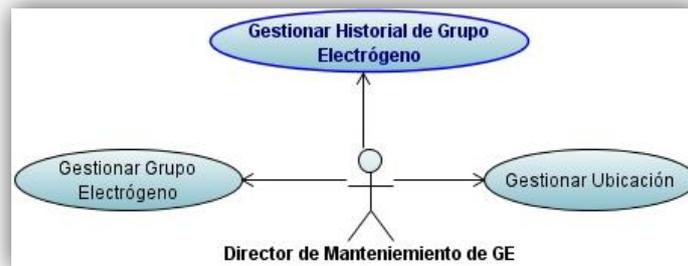


Figura 7: DCUS\_ Módulo de Mantenimiento

### 2.3.3 Descripciones abreviadas de los casos de uso del sistema

Las descripciones detalladas de los casos de uso se encuentran en el Anexo VII.

#### Módulo de Acceso:

<b>CUS_ : Autenticar Usuario</b>	
<b>Actor</b>	Administrador, Especialista de Energía, Especialista de Cajero Central, Especialista de Transporte, Director de Gestión Energética, Director de Transporte, Director de Mantenimiento, Vicerrector de Economía.
<b>Descripción</b>	El caso de uso se inicia cuando el actor ejecuta la aplicación y ésta inmediatamente le pide que se autentique introduciendo su usuario y contraseña, al introducir sus datos correctamente el sistema verifica sus privilegios y le da los permisos de acceso según el rol que este ocupe finalizando así el caso de uso.
<b>Referencia</b>	RF_3, 3.1

Tabla 12: CUS\_ Autenticar Usuario

<b>CUS_12: Gestionar Solicitud de Combustible</b>	
<b>Actor</b>	Especialista de Combustible
<b>Descripción</b>	El caso de uso se inicia cuando el actor ejecuta la aplicación y selecciona gestionar solicitud de combustible e inmediatamente se le muestra un listado con todas las solicitudes de combustible; brindando las opciones de adicionar, modificar y eliminar. El caso de uso termina cuando el sistema realiza la operación elegida, y el actor designado selecciona otra opción o sale de la aplicación.
<b>Referencia</b>	RF_12, 12.1, 12.2, 12.3, 12.4

Tabla 13: CUS\_Gestionar Solicitud de Combustible

### Módulo de Energía:

<b>CUS_23: Gestionar Plan de Combustible</b>	
<b>Actor</b>	Especialista de Energía
<b>Descripción</b>	El caso de uso se inicia cuando el actor ejecuta la aplicación y selecciona gestionar plan de combustible e inmediatamente se le muestra un listado con todos los planes de combustible; brindando las opciones de adicionar, modificar y eliminar. El caso de uso termina cuando el sistema realiza la operación elegida, y el actor designado selecciona otra opción o sale de la aplicación.
<b>Referencia</b>	RF_23, 23.1, 23.2, 23.3, 23.4

Tabla 14: CUS\_Gestionar Plan de Combustible

<b>CUS_38: Asignar Actividad a Plan</b>	
<b>Actor</b>	Especialista de Energía
<b>Descripción</b>	El caso de uso se inicia cuando actor ejecuta la aplicación y selecciona Asignar Actividad a Plan e inmediatamente se le muestra un formulario que le permite seleccionar un plan (sobre el cual se modificar o eliminar una actividad seleccionada); o realizar una nueva asignación. El caso de uso termina cuando el sistema realiza la operación elegida, y el actor designado selecciona otra opción o sale de la aplicación.
<b>Referencia</b>	RF_38, 38.1, 38.2, 38.3, 38.4

Tabla 15: CUS\_Asignar Actividad a Plan

<b>CUS_45: Generar Reportes de Energía</b>	
<b>Actor</b>	Especialista de Energía
<b>Descripción</b>	El caso de uso se inicia cuando el actor ejecuta la aplicación y selecciona realizar reporte por: consumo de combustible, consumo por año, altos consumidores o consumo por actividad_mes, se le muestra un listado con sus datos, permitiendo imprimir estos en formato Excel o pdf. El caso de uso termina cuando el sistema realiza la operación elegida, y el actor designado selecciona otra opción o sale de la aplicación.
<b>Referencia</b>	RF_45, 45.1, 45.2, 45.3, 45.4

Tabla 16: CUS\_Reporte de Consumo de Combustible

### Módulo de Cajero Central

<b>CUS_48: Gestionar Liquidación</b>	
<b>Actor</b>	Especialista de Cajero Central
<b>Descripción</b>	El caso de uso se inicia cuando el actor ejecuta la aplicación y selecciona gestionar liquidación e inmediatamente se le muestra un listado con todas las liquidaciones; brindando las opciones de adicionar, modificar y eliminar. El caso de uso termina cuando el sistema realiza la operación elegida, y el actor designado selecciona otra opción o sale de la aplicación.
<b>Referencia</b>	RF_48, 48.1, 48.2, 48.3, 48.4

Tabla 17: CUS\_ Gestionar Liquidación

### Mantenimiento

<b>CUS_57: Gestionar Historial de Grupo Electrónico</b>	
<b>Actor</b>	Director de Mantenimiento de Grupos Electrónicos
<b>Descripción</b>	El caso de uso se inicia cuando el actor ejecuta la aplicación y selecciona gestionar historial de grupo electrónico e inmediatamente se le muestra un listado con todos los historiales de grupos electrónicos; brindando las opciones de adicionar, modificar y eliminar. El caso de uso termina cuando el sistema realiza la operación elegida, y el actor designado selecciona otra opción o sale de la aplicación.
<b>Referencia</b>	RF_57, 57.1, 57.2, 57.3, 57.4

Tabla 18: CUS\_ Gestionar Historial de Grupo Electrónico

### **Conclusiones Parciales**

- Se obtiene mediante el desarrollo del modelo del negocio una panorámica de las funcionalidades que la aplicación debe cumplir.
- Se define las funcionalidades de la aplicación mediante el levantamiento de requisitos funcionales y no funcionales, logrando una visión concreta de los usuarios y las actividades que realizan.
- Se adquiere una visión más acertada de los procesos para la realización del análisis y diseño.

## CAPÍTULO III: ANÁLISIS Y DISEÑO

### Introducción

En este capítulo se desarrollará el Análisis y Diseño de las principales funcionalidades a implementar en el desarrollo del Sistema de Gestión del Combustible en la UCI. Serán modelados los casos de uso significativos del sistema, en cuanto al análisis con: los diagramas de clases del análisis y el diseño: los diagramas de clases del diseño y los diagramas de interacción (secuencia), concluyendo en el modelo de diseño de la base de datos\* que sienta las bases para la implementación.

### 3.1 Análisis

En el análisis se utiliza una combinación de textos y diagramas para describir los requisitos funcionales del software, estructurándolos en función de hacer más fácil su comprensión, modificación y en general su mantenimiento. A través del mismo se presenta una vista interna del sistema refinando los requisitos y agrupándolos en base a clases y paquetes.

#### 3.1.1 Modelo de Análisis

El modelo de análisis debe lograr tres objetivos primarios: describir lo que requiere el cliente, establecer una base para la creación del diseño del software, y definir un conjunto de requisitos que se puedan validar una vez que se construya el software. (20)

Clases del Análisis		
Nombre	Características	Representación
<b>Interfaz</b>	Modelan la interacción entre el sistema y los actores.	 Clase Interfaz
<b>Controladora</b>	Representan coordinación, secuencia, y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto.	 Clase Control
<b>Entidad</b>	Modelan la información que posee una vida larga y que a menudo es persistente.	 Clase Entidad

Tabla 19: Clases del Análisis

3.1.2 Diagramas de Clases del Análisis

**Módulo de Acceso:**

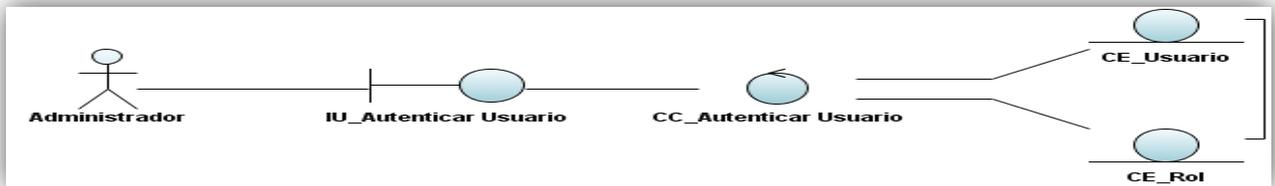


Figura 8: DCA Autenticar Usuario

**Módulo de Transporte:**

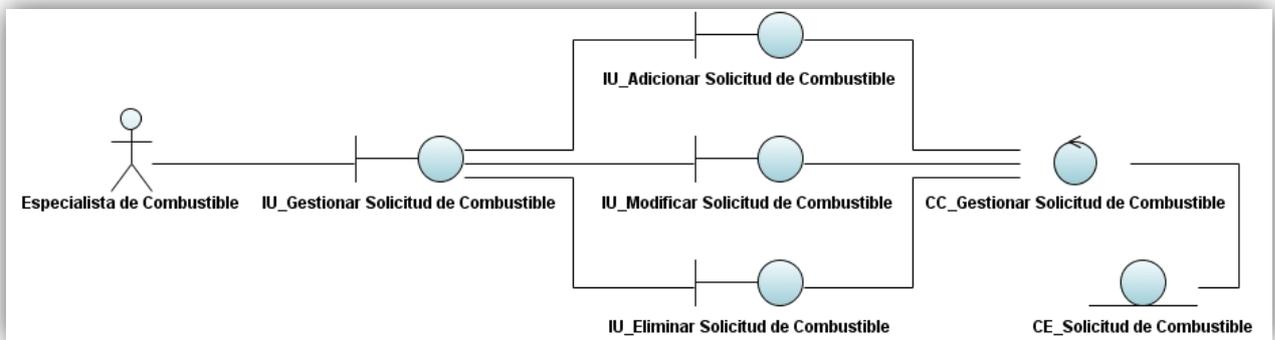


Figura 9: Diagrama de clases de Análisis CUS: Gestionar Solicitud de Combustible

**Módulo de Energía:**

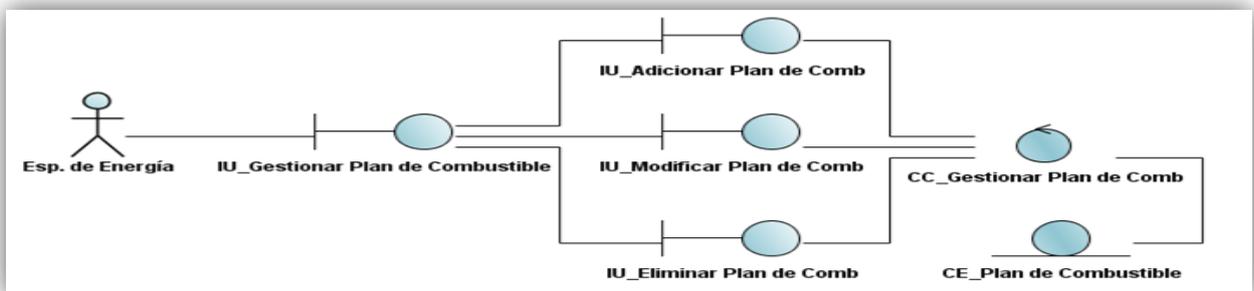


Figura 10: Diagrama de clases de Análisis CUS: Gestionar Plan de Combustible

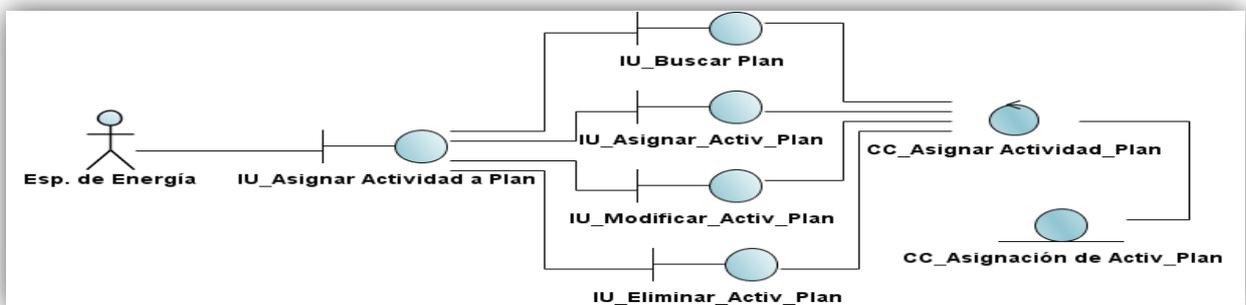


Figura 11: Diagrama de clases de Análisis CUS: Asignar Actividad a Plan

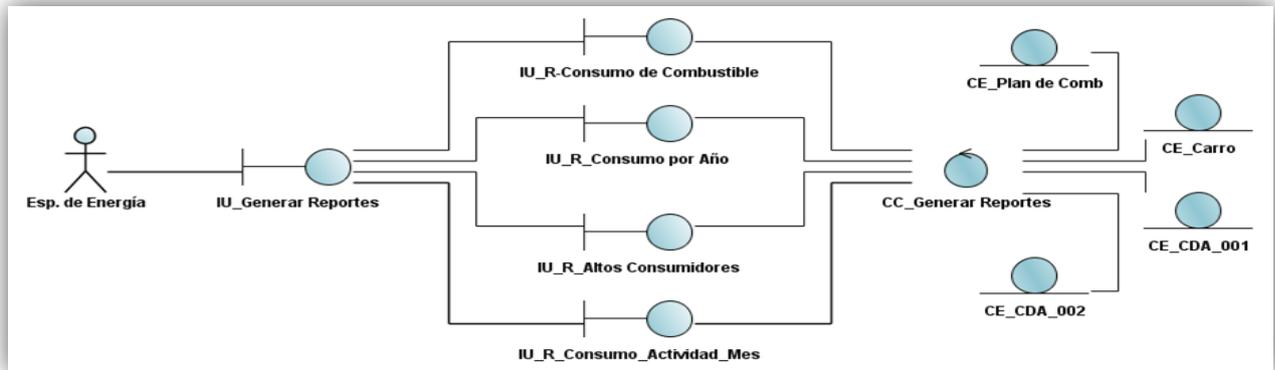


Figura 12: Diagrama de clases de Análisis CUS: Generar Reportes

**Módulo de Cajero Central:**

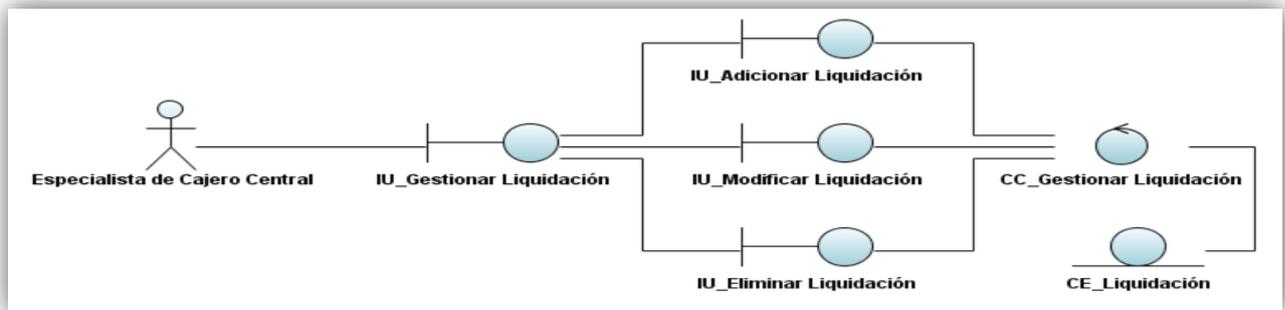


Figura 13: Diagrama de clases de Análisis CUS: Gestionar Liquidación

**Módulo de Mantenimiento:**

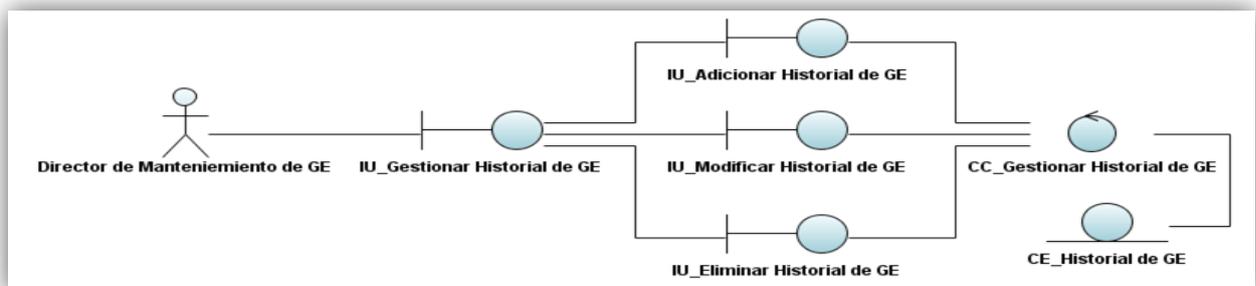


Figura 14: Diagrama de clases de Análisis CUS: Gestionar Historial de Grupo Electrónico

**3.2 Diseño**

El diseño es un proceso iterativo mediante el cual los requisitos se traducen en un plano para construir el software, el cual se encuentra en el núcleo técnico de la ingeniería de software. La importancia del diseño se puede traducir en una sola palabra, calidad, ya que sin este se corre el riesgo de construir un sistema inestable, que fallará cuando se lleven a cabo cambios; que

pueden resultar difíciles de comprobar, y cuya calidad no puede evaluarse hasta muy avanzado el proceso. (20)

### 3.2.1 Modelo del diseño

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

#### Clases del diseño

Una clase de diseño es una construcción similar a una clase en la implementación del sistema.

Estereotipos web		
Nombre	Descripción	Representación
<b>Página Cliente</b>	Es una instancia de Página Cliente, es una página web, con formato HTML*.	 ««Client page»»
<b>Formulario</b>	Colección de elementos de entrada que son parte de una página cliente.	 ««Form»»
<b>Página Servidora</b>	Representa la página web que tiene código, que se ejecuta en el servidor.	 ««Server page»»

Tabla 20: Estereotipos web

### 3.2.2 Diagramas de Clases del Diseño Módulo de Acceso

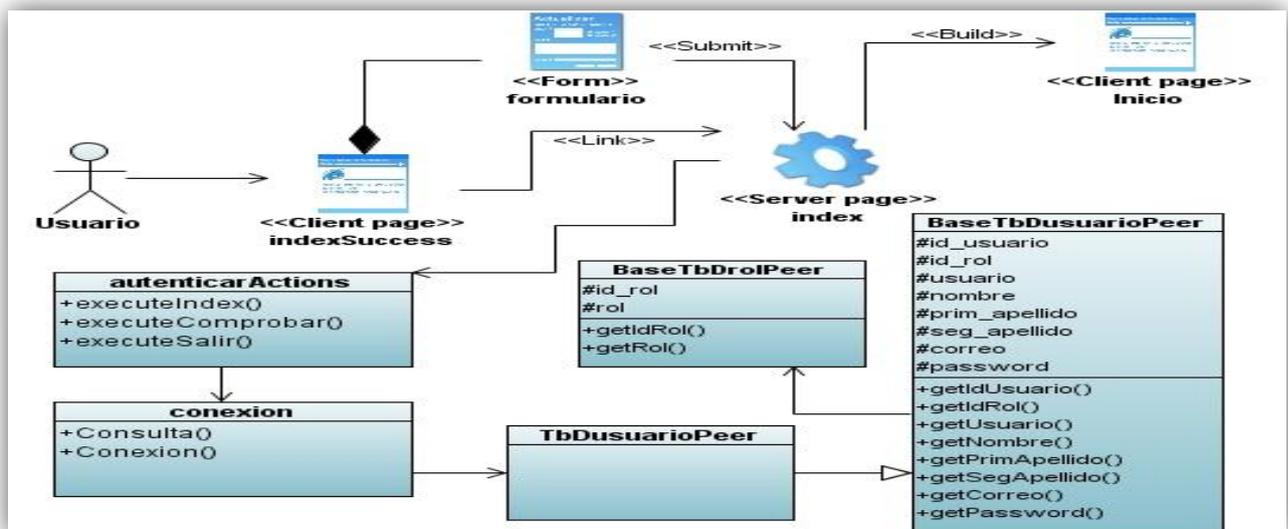


Figura 15: Diagrama de Clases del Diseño CUS: Autenticar Usuario

**Módulo de Transporte:**

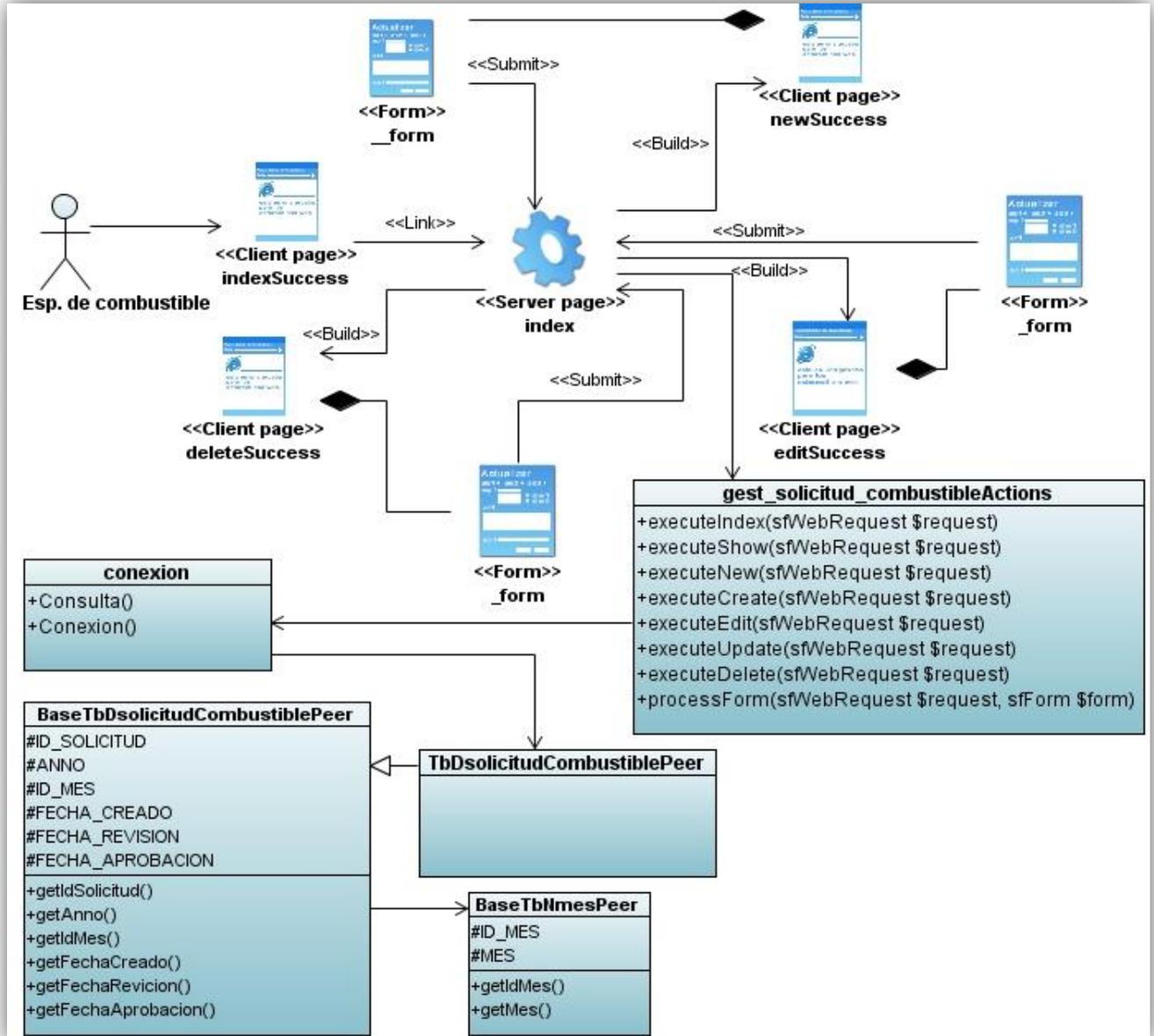


Figura 16: Diagrama de Clases del Diseño CUS: Gestionar Solicitud de Combustible







**Módulo de Cajero Central**

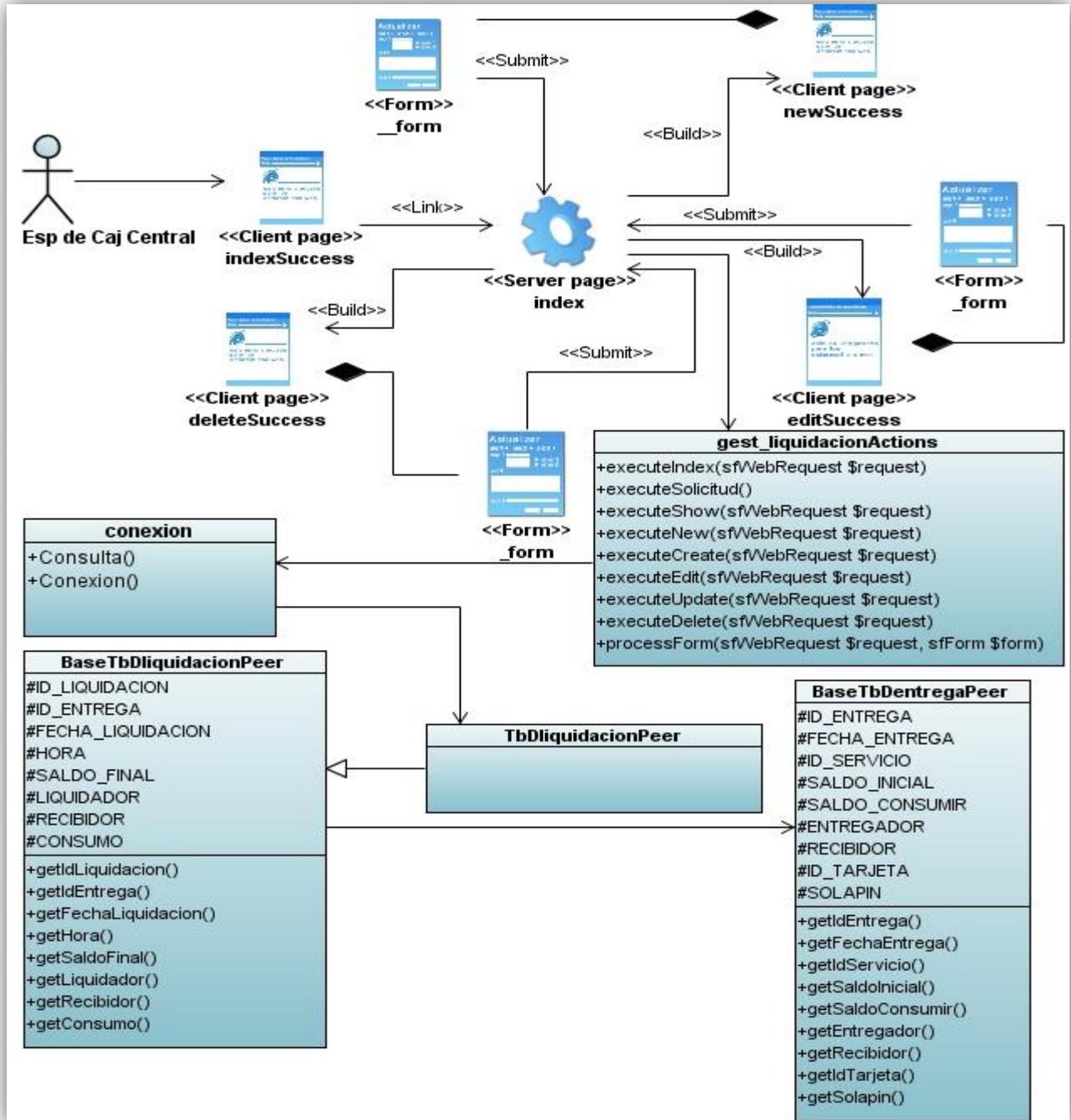


Figura 20: Diagrama de Clases del Diseño CUS: Gestionar Liquidación

**Módulo de Mantenimiento:**

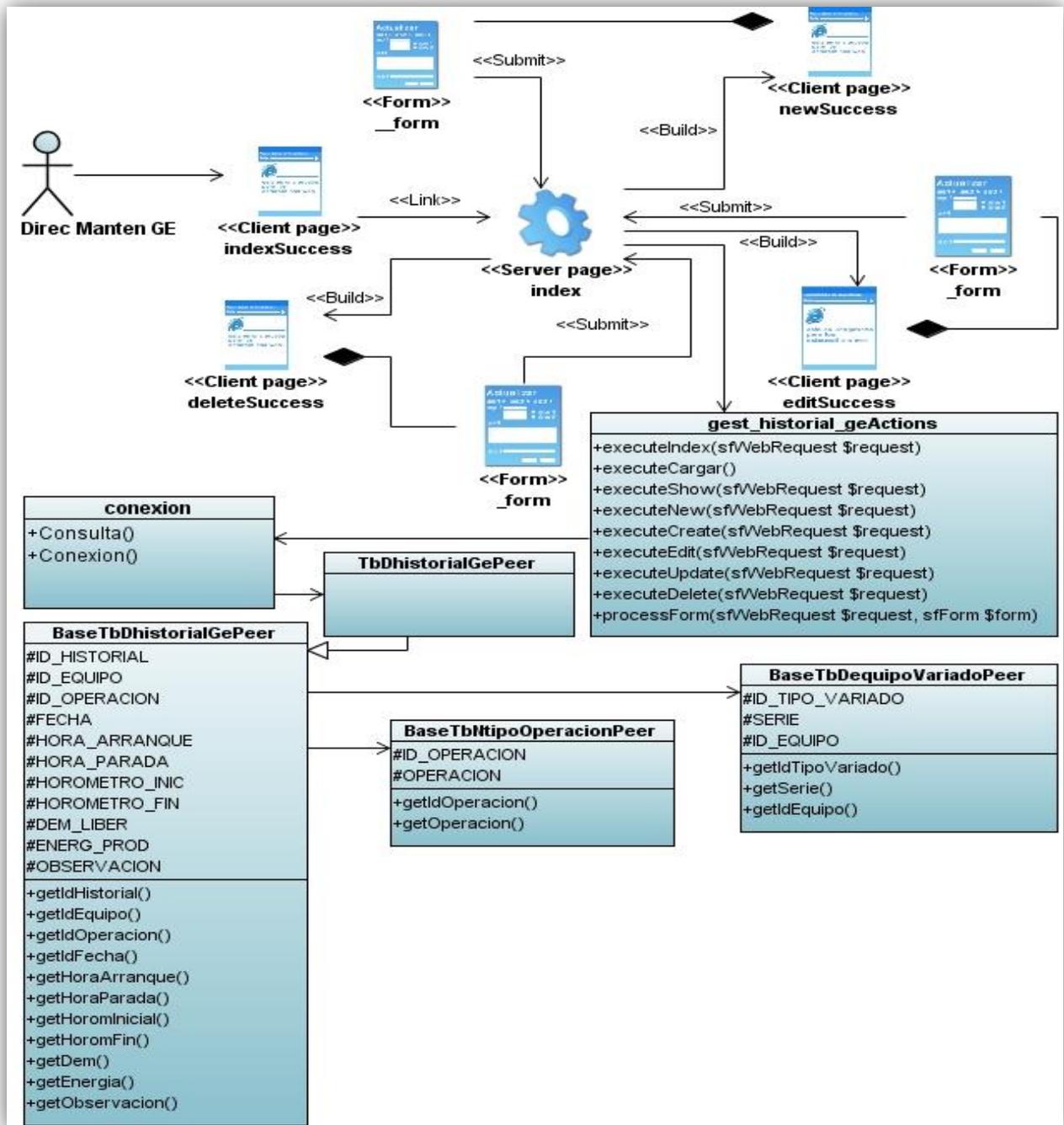


Figura 21: Diagrama de Clases del Diseño CUS: Gestionar Historial de Grupo Electrógeno

### 3.2.3 Diagramas de Interacción

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema. Un diagrama de interacción consiste en un conjunto de objetos y sus relaciones, incluyendo los mensajes que se pueden enviar entre ellos.

**Diagrama de Secuencia:** Un diagrama de secuencia es un diagrama de interacción que destaca la ordenación temporal de los mensajes (mensaje simple, síncrono, asíncrono, y/o de retorno). Este tipo de diagrama se destaca por tener la línea de vida y el foco de control que representa el período de tiempo durante el cual un objeto ejecuta una acción. Ver Anexo VIII.

### 3.2.4 Descripción de clases del diseño

#### Clases Controladoras

Nombre: autenticarActions	
<b>Tipo de clase:</b> controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	executeIndex() executeSalir() executeComprobar()
<b>Descripción:</b>	Esta clase hereda de la clase sfActions haciendo uso de los métodos definidos en la misma.

Tabla 21: autenticarActions

Nombre: gest_solicitud_combustibleActions	
<b>Tipo de clase:</b> controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	executeIndex(sfWebRequest \$request) executeShow(sfWebRequest \$request) executeNew(sfWebRequest \$request) executeCreate(sfWebRequest \$request) executeEdit(sfWebRequest \$request) executeUpdate(sfWebRequest \$request) executeDelete(sfWebRequest \$request) processForm(sfWebRequest \$request, sfForm \$form)
<b>Descripción:</b>	Esta clase hereda de la clase sfActions haciendo uso de los métodos definidos en la misma.

Tabla 22: gest\_solicitud\_combustibleActions

Nombre: gest_planActions	
<b>Tipo de clase:</b> controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	executeIndex(sfWebRequest \$request) executeCargar() executeShow(sfWebRequest \$request) executeNew(sfWebRequest \$request) executeCreate(sfWebRequest \$request)

	executeEdit(sfWebRequest \$request) executeUpdate(sfWebRequest \$request) executeDelete(sfWebRequest \$request) processForm(sfWebRequest \$request, sfForm \$form)
<b>Descripción:</b>	Esta clase hereda de la clase sfActions haciendo uso de los métodos definidos en la misma.

Tabla 23: gest\_planActions

<b>Nombre: gest_actividad_planActions</b>	
<b>Tipo de clase:</b> controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	executeIndex(sfWebRequest \$request) executeActividad() executeDato() executeShow(sfWebRequest \$request) executeNew(sfWebRequest \$request) executeCreate(sfWebRequest \$request) executeEdit(sfWebRequest \$request) executeUpdate(sfWebRequest \$request) executeDelete(sfWebRequest \$request) processForm(sfWebRequest \$request, sfForm \$form)
<b>Descripción:</b>	Esta clase hereda de la clase sfActions haciendo uso de los métodos definidos en la misma.

Tabla 24: gest\_actividad\_planActions

<b>Nombre: reporteActions</b>	
<b>Tipo de clase:</b> controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	executeConsumo_combustible() executeConsumo_combustible_cargar() executeConsumo_combustible_imprimir() executeConsumo_anno() executeConsumo_anno_cargar() executeConsumo_anno_imprimir() executeAlto_consumidor() executeAlto_consumidor_cargar() executeAlto_consumidor_imprimir() executeConsumo_actividad_mes() executeConsumo_actividad_mes_cargar() executeConsumo_actividad_mes_imprimir() executeConsumo_historico_actividad() executeGet_anno() executeGet_producto() executeGet_mes()
<b>Descripción:</b>	Esta clase hereda de la clase sfActions haciendo uso de los métodos definidos en la misma.

Tabla 25: reporteActions

<b>Nombre: gest_liquidacionActions</b>	
<b>Tipo de clase:</b> controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	executeIndex(sfWebRequest \$request)

	executeSolicitud() executeShow(sfWebRequest \$request) executeNew(sfWebRequest \$request) executeCreate(sfWebRequest \$request) executeEdit(sfWebRequest \$request) executeUpdate(sfWebRequest \$request) executeDelete(sfWebRequest \$request) processForm(sfWebRequest \$request, sfForm \$form)
<b>Descripción:</b>	Esta clase hereda de la clase sfActions haciendo uso de los métodos definidos en la misma.

Tabla 26: gest\_liquidacionActions

<b>Nombre: gest_historial_geActions</b>	
<b>Tipo de clase:</b> controladora	
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	executeIndex(sfWebRequest \$request) executeCargar() executeShow(sfWebRequest \$request) executeNew(sfWebRequest \$request) executeCreate(sfWebRequest \$request) executeEdit(sfWebRequest \$request) executeUpdate(sfWebRequest \$request) executeDelete(sfWebRequest \$request) processForm(sfWebRequest \$request, sfForm \$form)
<b>Descripción:</b>	Esta clase hereda de la clase sfActions haciendo uso de los métodos definidos en la misma.

Tabla 27:gest\_historial\_geActions

### Clases entidades Padres

<b>Nombre: BaseTbDsolicitudCombustiblePeer</b>	
<b>Tipo de clase:</b> entidad	
<b>Atributo</b>	<b>Tipo</b>
#ID_SOLICITUD	Int
#ID_MES	Int
#ANNO	Int
#FECHA_CREADO	Date
#FECHA_REVISION	Date
#FECHA_APROBACION	Date
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	+getIdSolicitud() +getAnno() +getIdMes() +getFechaCreado() +getFechaRevicion() +getFechaAprobacion()
<b>Descripción:</b>	Esta es una clase entidad abstracta

Tabla 28: BaseTbDsolicitudCombustiblePeer

<b>Nombre: BaseTbRactividadPlanPeer</b>	
<b>Tipo de clase:</b> entidad	
<b>Atributo</b>	<b>Tipo</b>
#ID	Int

#ID_ACTIVIDAD	Int
#ID_PLAN	Int
#ID_UNIDAD_MEDIDA	Int
#NIVEL_REAL	Float
#NIVEL_ESTIMADO	Float
#CONSUMO_ESTIMADO	Float
#NIVEL_PROPOSTA	Float
#CONSUMO_PROPOSTA	Float
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	+getId() +getIdActividad() +getIdActividad() +getIdUM() +getNivelReal() +getConsumoReal() +getNivelEstimado() +getConsumoEstimado() +getNivelPropuesta() +getCosnumoPropuesta()
<b>Descripción:</b>	Esta es una clase entidad abstracta

Tabla 29: BaseTbRactividadPlanPeer

<b>Nombre: BaseTbDplanPeer</b>	
<b>Tipo de clase:</b> entidad	
<b>Atributo</b>	<b>Tipo</b>
#ID_PLAN	Int
#ID_EMPRESA	Int
#ID_PRODUCTO	Int
#ANNO	Int
#CODIGO	String
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	+getIdPlan() +getIdEmpresa() +getIdProducto() +getAnno() +getCodigo()
<b>Descripción:</b>	Esta es una clase entidad abstracta

Tabla 30: BaseTbDplanPeer

<b>Nombre: BaseTbDhistorialGePeer</b>	
<b>Tipo de clase:</b> entidad	
<b>Atributo</b>	<b>Tipo</b>
#ID_HISTORIAL	Int
#ID_EQUIPO	Int
#ID_OPERACION	Int
#FECHA	Date
#HORA_ARRANQUE	Time

#HORA_PARADA	Time
#HOROMETRO_INIC	Time
#HOROMETRO_FIN	Time
#DEM_LIBER	Float
#ENERG_PROD	Float
#OBSERVACION	String
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	+getIdHistorial() +getIdEquipo() +getIdOperacion() +getIdOperacion() +getHoraArranque() +getHoraParada() +getHoromInicial() +getHoromFin() +getDem() +getEnergia() +getObservacion()
<b>Descripción:</b>	Esta es una clase entidad abstracta

Tabla 31: BaseTbDhistorialGePeer

Las descripciones de las clases del diseño se encuentran en el Anexo IX.

### 3.3 Diseño de la Base de Datos

Se muestra en el Anexo X.

### Conclusiones Parciales

- Se elaboraron los artefactos correspondientes al análisis y diseño de la aplicación con el fin de detallar sus funcionalidades para dar inicio a la fase de implementación. Destacándose en:

**El análisis:** Los diagramas de clases del análisis.

**El diseño:** Los diagramas de clases del diseño, los diagramas de secuencia, la descripción de clases del diseño y el modelo de base de datos\*.

# CAPÍTULO IV: IMPLEMENTACIÓN Y PRUEBA

## Introducción

En el presente capítulo se abordarán los flujos de trabajo correspondientes a implementación y prueba. Dentro de la implementación se confeccionarán los diagramas de despliegue y componentes; en el proceso de pruebas se utilizará el método de caja negra con el objetivo de comprobar el funcionamiento del sistema.

### 4.1 Modelo de Implementación

El modelo de implementación describe cómo los elementos del diseño se implementan en términos de componentes, ficheros de código fuente, ejecutables etc. Para llegar a abarcar la organización y composición del sistema mediante los diagramas de despliegue y componentes.

#### 4.1.1 Diagrama de Despliegue

El diagrama de despliegue permite conocer la ubicación física de los nodos, así como los protocolos de conexión entre ellos y la biblioteca utilizada en la comunicación del servidor de aplicación con la base de datos\*.

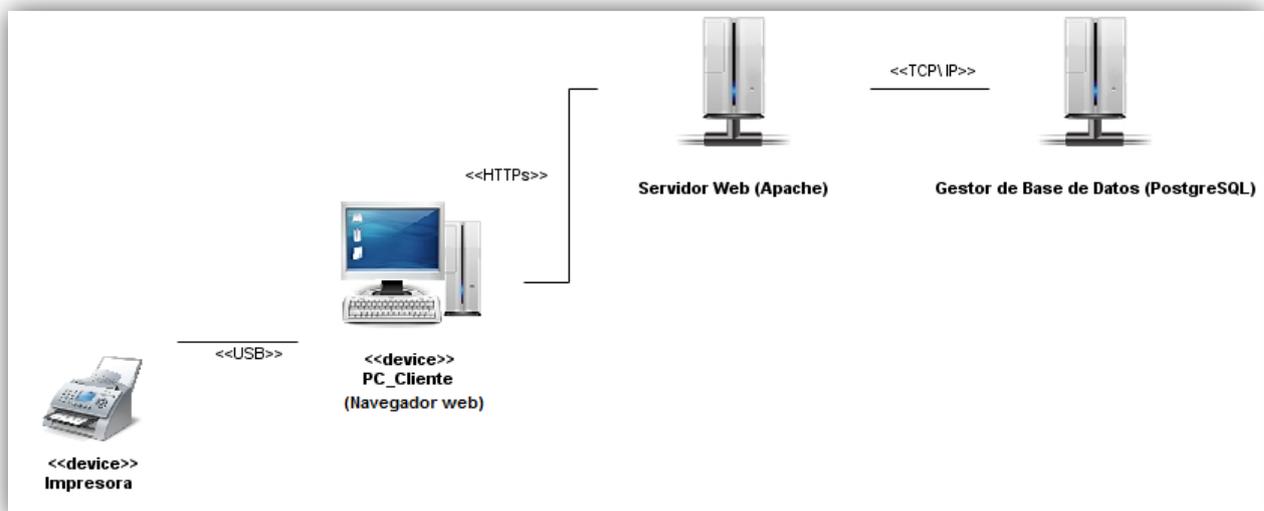


Figura 22: Diagrama de Despliegue

#### 4.1.2 Diagrama de Componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, sean estos componentes de código fuente, binarios o ejecutables. Se

tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo.

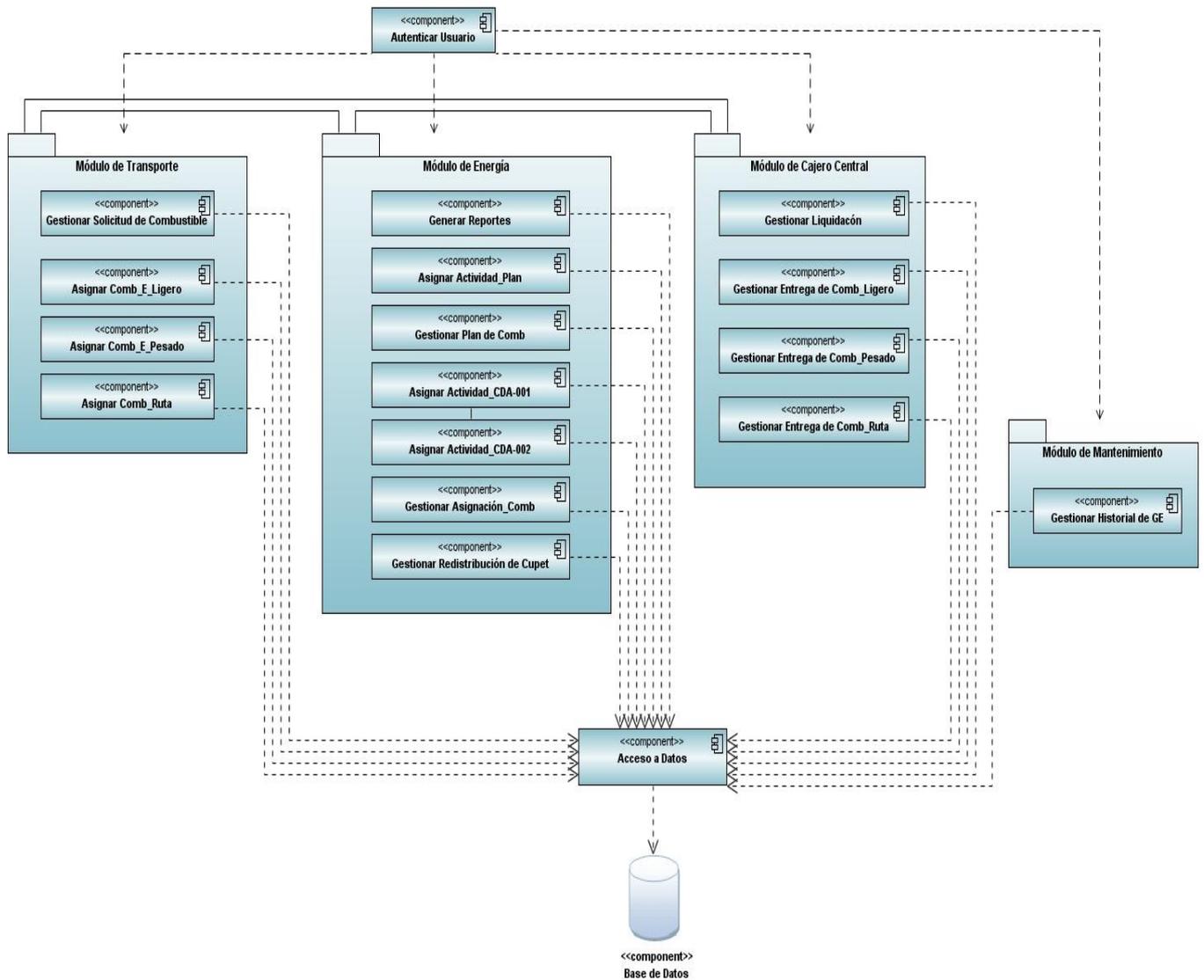


Figura 23: Diagrama de Componentes

Diagramas de Componentes por Casos de Uso

Módulo de Acceso

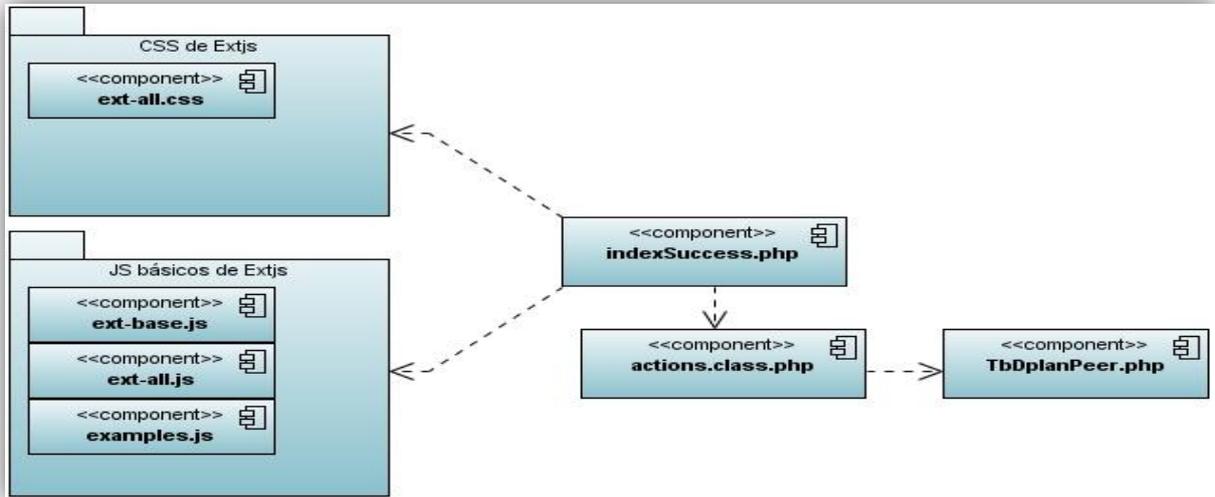


Figura 24: CUS\_ Autenticar Usuario

Módulo de Transporte

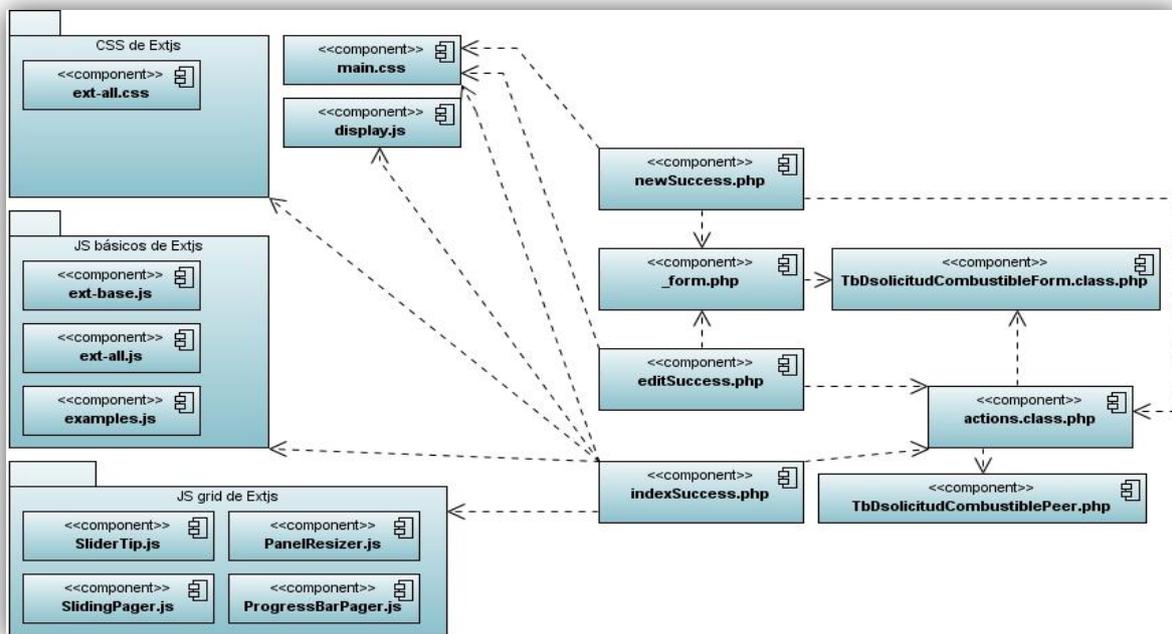


Figura 25: CUS\_ Gestionar Solicitud de Combustible

**Módulo de Energía**

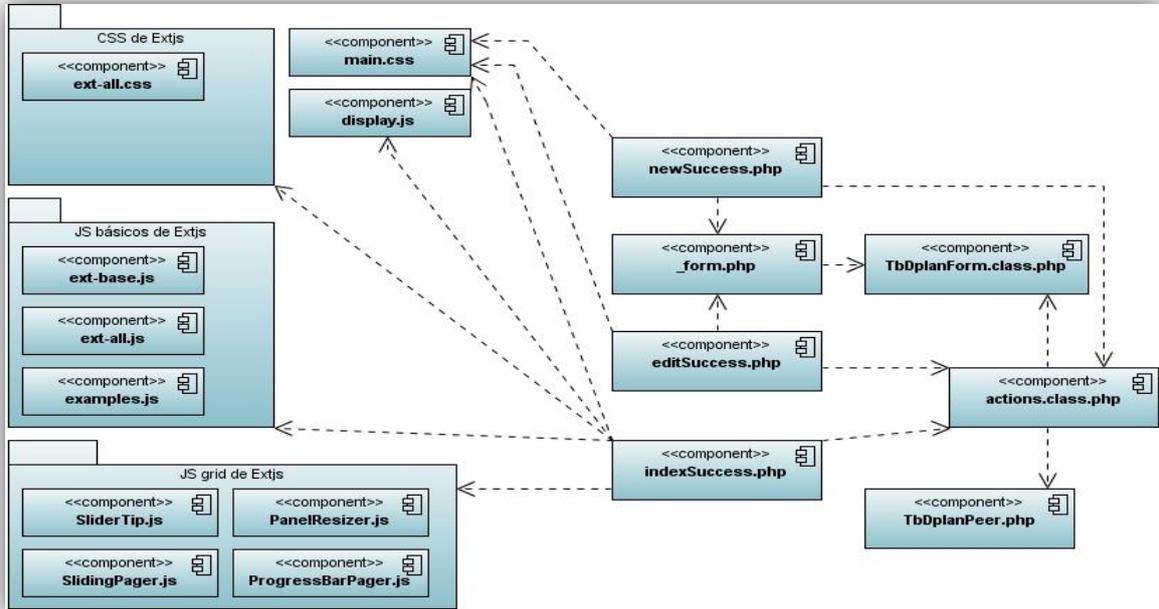


Figura 26: CUS\_ Gestionar Plan de Combustible

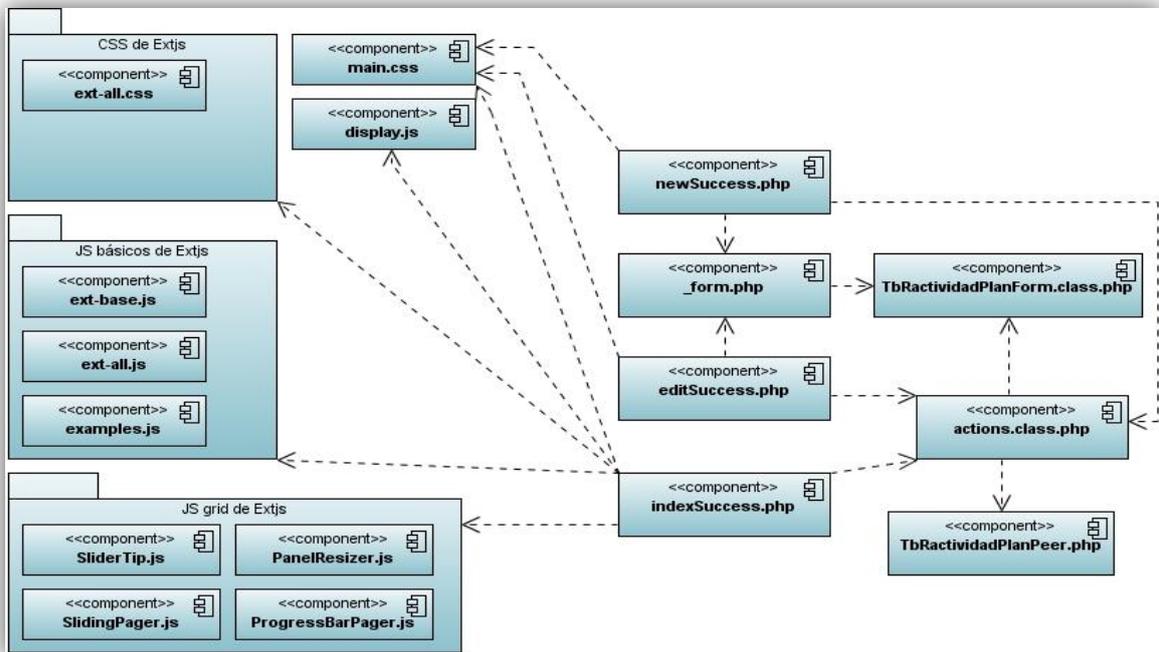


Figura 27: CUS\_ Asignar Actividad a Plan

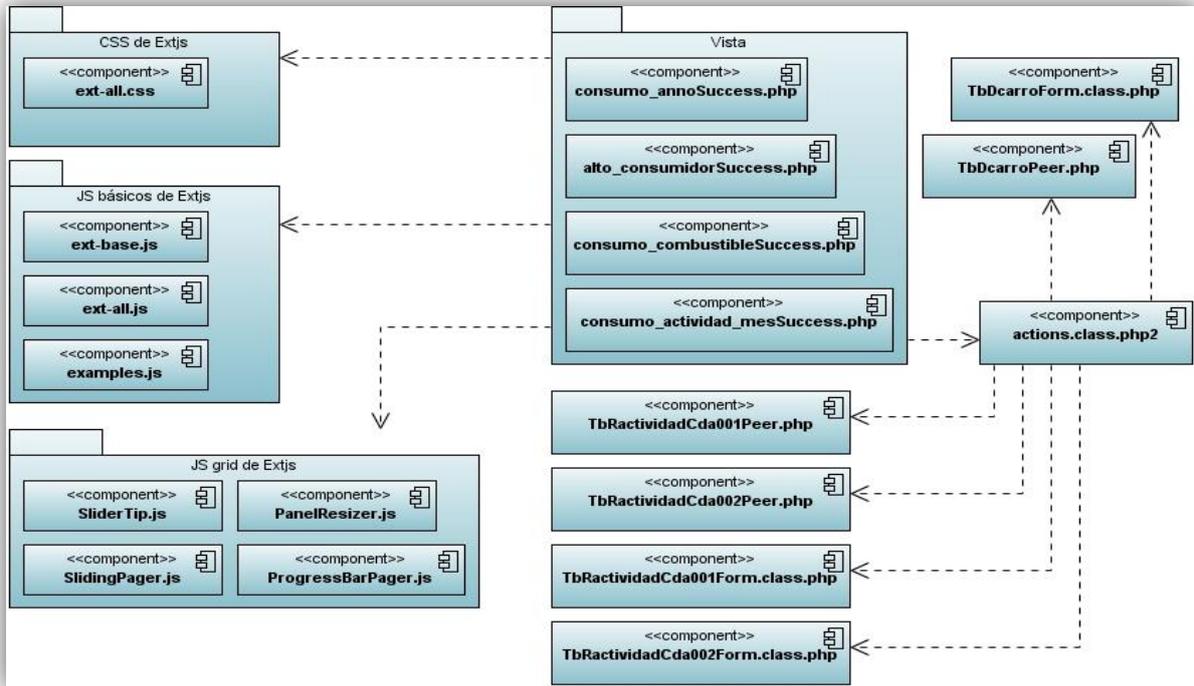


Figura 28: CUS\_Generar Reportes

**Módulo de Cajero Central**

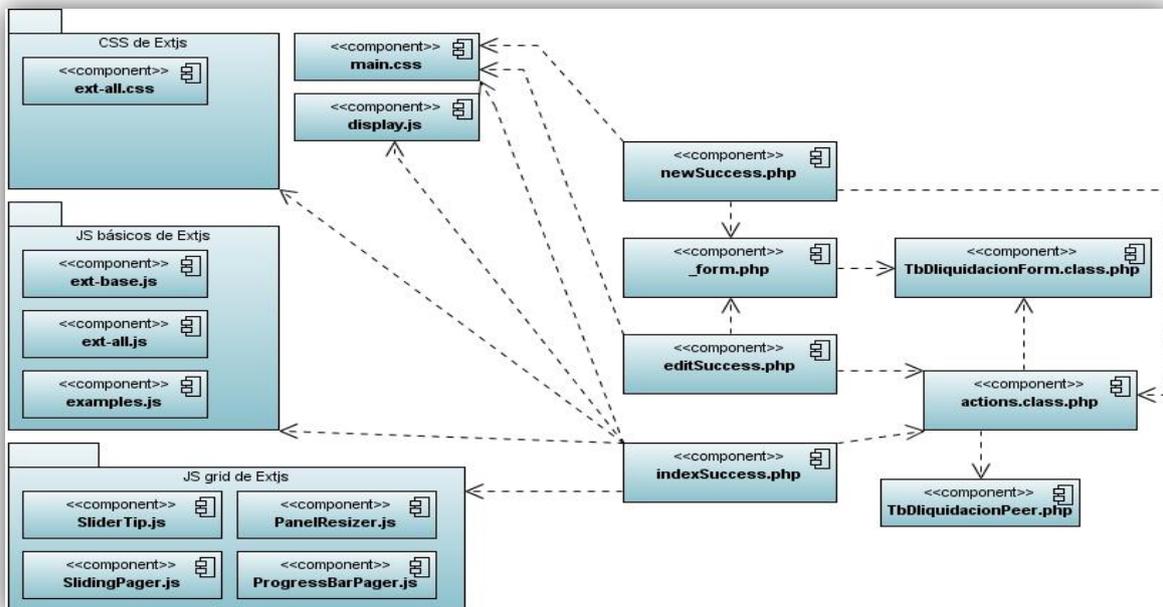


Figura 29: CUS\_Gestionar Liquidación

**Módulo de Mantenimiento**

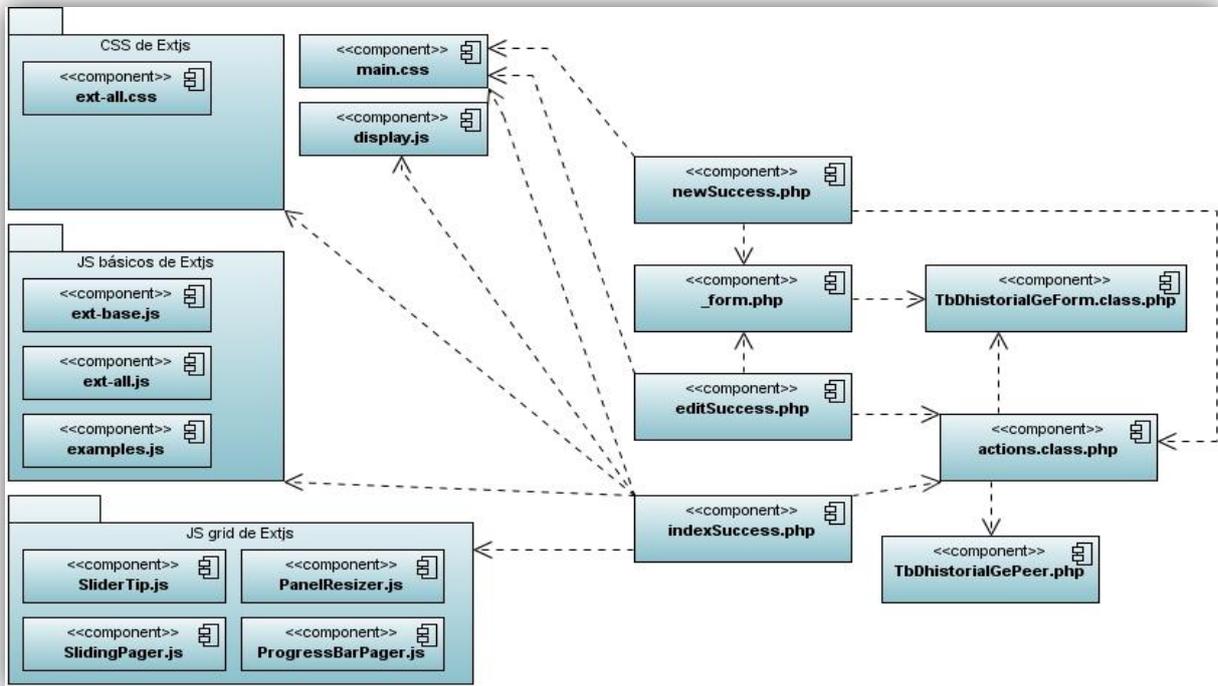


Figura 30: Gestionar Historial de Grupo Electrónico

Los diagramas de componentes por casos de uso se encuentran en el Anexo XI.

**4.2 Modelo de Prueba**

Las pruebas de software son los procesos de ejercitar un programa con la intención específica de encontrar errores previos a la entrega del sistema al usuario final. Están encaminadas a garantizar la calidad del software en todo momento del desarrollo. Durante este flujo de trabajo se comprueba el resultado de la implementación, planificando, diseñando e implementando los casos de prueba.

**4.2.1 Pruebas de Caja Negra**

Las Pruebas de Caja Negra son llevadas a cabo sobre la interfaz del software, pretendiendo demostrar que sus funcionalidades son operativas, que la entrada se acepta de forma apropiada, que se produce una salida de forma eficiente y que la integridad de la información externa se mantiene. Se derivan conjuntos de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales del programa. Anexo XII.

### Casos de Prueba (CP)

Un caso de prueba es un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados para un objetivo particular. Los casos de pruebas se pueden derivar de los casos de usos del sistema o de la realización de estos en el modelo de diseño, permitiendo así validar los requerimientos funcionales del sistema.

Para realizar las pruebas del sistema se realizaron pruebas de cajas negras basado en un método de tres pasos obteniendo un conjunto de casos de prueba del sistema a partir de los CU; los pasos son los siguientes:

Paso	Descripción	Resultado
1	Generar escenarios de uso	Obtener los posibles caminos de ejecución de cada caso de uso. Cada camino es un escenario de uso.
2	Identificar casos de prueba	Conjunto de casos de prueba a partir de los escenarios anteriores.
3	Identificar los valores a probar	Valores de prueba asociados a cada caso de prueba anterior.

Tabla 32: Pasos para realizar los casos de prueba

### Para generar los casos de prueba se seguirán los siguientes pasos:

- Para cada CU, generar un sistema completo de los escenarios.
- Para cada escenario, identificar los casos de pruebas y las condiciones que hagan que se ejecute.
- Para cada caso de prueba, identificar los valores de los datos con los cuales se harán las pruebas.
- Las celdas de la tabla contienen V o I, siendo válido e inválido correspondientemente.

### Módulo Transporte

#### Casos de prueba con valores

#### CUS: Gestionar Solicitud de Combustible

Escenario	Nombre	Flujo de comienzo	Alternativo
1	Crear solicitud	Flujo Normal	Flujo alternativo 1
2	Modificar solicitud	Flujo Normal	Flujo alternativo 2
3	Eliminar solicitud	Flujo Normal	

Tabla 33: Escenarios de Prueba

Escenario del CUS_ Gestionar Solicitud de Combustible			
ID	Escenario	Datos	Resultado esperado
CP1	1	V	Se adiciona la solicitud en la base de datos* del sistema. Se notifica al usuario que fueron adicionados correctamente

CP2	1	I	Se visualizan en el formulario los errores correspondientes, dándole la oportunidad al usuario de corregir los mismos
CP3	2	V	Se modifica la solicitud en la base de datos* del sistema. Se notifica al usuario que los cambios fueron realizados correctamente.
CP4	2	I	Se visualizan en el formulario los errores correspondientes, dándole la oportunidad al usuario de corregir los mismos
CP5	3	V	Se elimina la solicitud de la base de datos* del sistema. Se le notifica al usuario "Eliminado correctamente".

Tabla 34: Escenario del CUS\_ Gestionar Solicitud de Combustible

Casos de Prueba			
ID	Escenario	Datos	Resultado esperado
CP1	1	Se registran los datos correctamente.	Siendo los datos correctos se almacena en la base de datos* la solicitud. Además, se le notifica al usuario con un mensaje de éxito (Adicionado correctamente).
CP2	1	Se registran datos incorrectos o vacíos.	Es notificado al usuario los errores cometidos en el formulario de inserción, dando la posibilidad de corregirlos y proceder a la inserción
CP3	2	Se registran los datos correctamente.	Siendo los nuevos datos correctos se procede a modificar la solicitud en la base de datos* con los mismos, notificando posteriormente al usuario con un mensaje de éxito(Adicionado correctamente)
CP4	2	Se registran datos incorrectos o vacíos.	Es notificado al usuario los errores cometidos en el formulario de modificación, dando la posibilidad de corregirlos y proceder a la actualización.
CP5	3	El usuario selecciona la solicitud a eliminar, seleccionado posteriormente el botón de eliminar	La solicitud es eliminada de la base de datos*, notificando al usuario con un mensaje de éxito (Eliminado correctamente).

Tabla 35: Casos de Prueba

### Módulo Energía

#### Casos de prueba con valores

##### CUS: Gestionar Plan de Combustible

Escenario	Nombre	Flujo de comienzo	Alternativo
1	Crear Plan	Flujo Normal	Flujo alternativo 1
2	Modificar Plan	Flujo Normal	Flujo alternativo 2
3	Eliminar Plan	Flujo Normal	

Tabla 36: Escenarios de Prueba

Escenario del CUS_ Gestionar Plan de Combustible			
ID	Escenario	Datos	Resultado esperado
CP1	1	V	Se adiciona el plan en la base de datos* del sistema. Se notifica al usuario que fueron adicionados correctamente

CP2	1	I	Se visualizan en el formulario los errores correspondientes, dándole la oportunidad al usuario de corregir los mismos
CP3	2	V	Se modifica el plan en la base de datos* del sistema. Se notifica al usuario que los cambios fueron realizados correctamente.
CP4	2	I	Se visualizan en el formulario los errores correspondientes, dándole la oportunidad al usuario de corregir los mismos
CP5	3	V	Se elimina el plan de la base de datos* del sistema. Se le notifica al usuario "Eliminado correctamente".

Tabla 37: Escenario del CUS\_ Gestionar Plan de Combustible

Casos de Prueba			
ID	Escenario	Asignación	Resultado esperado
CP1	1	Se registran los datos correctamente.	Siendo los datos correctos se almacena en la base de datos* el plan. Además, se le notifica al usuario con un mensaje de éxito (Adicionado correctamente).
CP2	1	Se registran datos incorrectos o vacíos.	Es notificado al usuario los errores cometidos en el formulario de inserción, dando la posibilidad de corregirlos y proceder a la inserción
CP3	2	Se registran los datos correctamente.	Siendo los nuevos datos correctos se procede a modificar el plan en la base de datos* con los mismos, notificando posteriormente al usuario con un mensaje de éxito(Adicionado correctamente)
CP4	2	Se registran datos incorrectos o vacíos.	Es notificado al usuario los errores cometidos en el formulario de modificación, dando la posibilidad de corregirlos y proceder a la actualización.
CP5	3	El usuario selecciona el plan, luego pulsa sobre el botón Eliminar	El plan es eliminado de la base de datos*, notificando al usuario con un mensaje de éxito (Eliminado correctamente).

Tabla 38: Casos de Prueba

### Casos de prueba con valores

#### CUS: Asignar Actividad a Plan

Escenario	Nombre	Flujo de comienzo	Alternativo
1	Crear Asignación	Flujo Normal	Flujo alternativo 1
2	Modificar Asignación	Flujo Normal	Flujo alternativo 2
3	Eliminar Asignación	Flujo Normal	

Tabla 39: Escenarios de Prueba

Escenario del CUS_ Asignar Actividad a Plan			
ID	Escenario	Datos	Resultado esperado
CP1	1	V	Se adiciona la asignación en la base de datos* del sistema. Se notifica al usuario que fueron adicionados correctamente
CP2	1	I	Se visualizan en el formulario los errores correspondientes, dándole la oportunidad al usuario de corregir los mismos

CP3	2	V	Se modifica la asignación en la base de datos* del sistema. Se notifica al usuario que los cambios fueron realizados correctamente.
CP4	2	I	Se visualizan en el formulario los errores correspondientes, dándole la oportunidad al usuario de corregir los mismos
CP5	3	V	Se elimina la asignación de la base de datos* del sistema. Se le notifica al usuario "Eliminado correctamente".

Tabla 40: Escenario del CUS\_ Asignar Actividad a Plan

Casos de Prueba			
ID	Escenario	Datos	Resultado esperado
CP1	1	Se registran los datos correctamente.	Siendo los datos correctos se almacena en la base de datos* la asignación. Además, se le notifica al usuario con un mensaje de éxito (Adicionado correctamente).
CP2	1	Se registran datos incorrectos o vacíos.	Es notificado al usuario los errores cometidos en el formulario de inserción, dando la posibilidad de corregirlos y proceder a la inserción
CP3	2	Se registran los datos correctamente.	Siendo los nuevos datos correctos se procede a modificar la asignación en la base de datos* con los mismos, notificando posteriormente al usuario con un mensaje de éxito(Adicionado correctamente)
CP4	2	Se registran datos incorrectos o vacíos.	Es notificado al usuario los errores cometidos en el formulario de modificación, dando la posibilidad de corregirlos y proceder a la actualización.
CP5	3	El usuario selecciona la asignación, luego pulsa sobre el botón Eliminar	La solicitud es eliminada de la base de datos*, notificando al usuario con un mensaje de éxito (Eliminado correctamente).

Tabla 41: Casos de Prueba

### Módulo de Cajero Central

#### Casos de prueba con valores

#### CUS: Gestionar Liquidación

Escenario	Nombre	Flujo de comienzo	Alternativo
1	Crear liquidación	Flujo Normal	Flujo alternativo 1
2	Modificar liquidación	Flujo Normal	Flujo alternativo 2
3	Eliminar liquidación	Flujo Normal	

Tabla 42: Escenarios de Prueba

Escenario del CUS_ Gestionar Liquidación			
ID	Escenario	Datos	Resultado esperado
CP1	1	V	Se adiciona la liquidación en la base de datos* del sistema. Se notifica al usuario que fueron adicionados correctamente
CP2	1	I	Se visualizan en el formulario los errores correspondientes, dándole la oportunidad al usuario de corregir los mismos
CP3	2	V	Se modifica la liquidación en la base de datos* del sistema.

			Se notifica al usuario que los cambios fueron realizados correctamente.
CP4	2	I	Se visualizan en el formulario los errores correspondientes, dándole la oportunidad al usuario de corregir los mismos
CP5	3	V	Se elimina la liquidación de la base de datos* del sistema. Se le notifica al usuario "Eliminado correctamente".

Tabla 43: Escenario del CUS\_ Gestionar Liquidación

<b>Casos de Prueba</b>			
ID	Escenario	Datos	Resultado esperado
CP1	1	Se registran los datos correctamente.	Siendo los datos correctos se almacena en la base de datos* la liquidación. Además, se le notifica al usuario con un mensaje de éxito (Adicionado correctamente).
CP2	1	Se registran datos incorrectos o vacíos.	Es notificado al usuario los errores cometidos en el formulario de inserción, dando la posibilidad de corregirlos y proceder a la inserción
CP3	2	Se registran los datos correctamente.	Siendo los nuevos datos correctos se procede a modificar la liquidación en la base de datos* con los mismos, notificando posteriormente al usuario con un mensaje de éxito(Adicionado correctamente)
CP4	2	Se registran datos incorrectos o vacíos.	Es notificado al usuario los errores cometidos en el formulario de modificación, dando la posibilidad de corregirlos y proceder a la actualización.
CP5	3	El usuario selecciona la liquidación, luego pulsa sobre el botón Eliminar	La liquidación es eliminada de la base de datos*, notificando al usuario con un mensaje de éxito (Eliminado correctamente).

Tabla 44: Casos de Prueba

### Módulo de Mantenimiento

#### Casos de prueba con valores

#### CUS: Gestionar Historial de Grupo Electrónico

Escenario	Nombre	Flujo de comienzo	Alternativo
1	Crear Historial	Flujo Normal	Flujo alternativo 1
2	Modificar Historial	Flujo Normal	Flujo alternativo 2
3	Eliminar Historial	Flujo Normal	

Tabla 45: Escenarios de Prueba

<b>Escenario del CUS_ Gestionar Historial de Grupo Electrónicos</b>			
ID	Escenario	Datos	Resultado esperado
CP1	1	V	Se adiciona el historial en la base de datos* del sistema. Se notifica al usuario que fueron adicionados correctamente
CP2	1	I	Se visualizan en el formulario los errores correspondientes, dándole la oportunidad al usuario de corregir los mismos
CP3	2	V	Se modifica el historial en la base de datos* del sistema. Se notifica al usuario que los cambios fueron realizados

			correctamente.
CP4	2	I	Se visualizan en el formulario los errores correspondientes, dándole la oportunidad al usuario de corregir los mismos
CP5	3	V	Se elimina el historial de la base de datos* del sistema. Se le notifica al usuario "Eliminado correctamente".

Tabla 46: Escenario del CUS\_ Gestionar Historial de Grupo Electrónos

<b>Casos de Prueba</b>			
ID	Escenario	Datos	Resultado esperado
CP1	1	Se registran los datos correctamente.	Siendo los datos correctos se almacena en la base de datos* el historial. Además, se le notifica al usuario con un mensaje de éxito (Adicionado correctamente).
CP2	1	Se registran datos incorrectos o vacíos.	Es notificado al usuario los errores cometidos en el formulario de inserción, dando la posibilidad de corregirlos y proceder a la inserción
CP3	2	Se registran los datos correctamente.	Siendo los nuevos datos correctos se procede a modificar el historial en la base de datos* con los mismos, notificando posteriormente al usuario con un mensaje de éxito(Adicionado correctamente)
CP4	2	Se registran datos incorrectos o vacíos.	Es notificado al usuario los errores cometidos en el formulario de modificación, dando la posibilidad de corregirlos y proceder a la actualización.
CP5	3	El usuario selecciona el historial, luego pulsa sobre el botón Eliminar	El historial es eliminado de la base de datos*, notificando al usuario con un mensaje de éxito (Eliminado correctamente).

Tabla 47: Casos de Prueba

### Conclusiones Parciales

- Se obtuvo una aplicación web para la gestión de la información del combustible en la UCI, que elimina el trabajo manual y la comunicación mediante el correo electrónico.
- Culminó la implementación de la aplicación cumpliendo con las funcionalidades propuestas.
- Se desarrollaron pruebas a la aplicación para comprobar su correcto funcionamiento y fácil manejo por parte de los usuarios finales, para lograr así una mayor eficiencia y manejo de la información del combustible en la Universidad.

### CONCLUSIONES GENERALES

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

- Una Aplicación Web para la Gestión de Información del Combustible en la Universidad de las Ciencias Informáticas (AGIC-UCI), que integra los procesos desarrollados en los Departamentos de Transporte, Energía, Mantenimiento y Cajero Central, logrando una reducción del tiempo empleado para el control y asignación de combustible, así como eficiencia en la búsqueda de información manejada por el usuario, para la creación de informes y reportes.
- Una documentación sólida de los métodos y modelos utilizados para la realización de la aplicación.
- El cumplimiento del objetivo elaborado en el diseño de la investigación, comprobando la hipótesis como respuesta del problema científico que tributa directamente en la gestión de la información del combustible en la Universidad de las Ciencias Informáticas.
- La validación teórica y funcional de la aplicación.

### **RECOMENDACIONES**

- Hacer uso de los servicios web que brinda la universidad.
- Cambiar la interfaz gráfica acorde a las pautas de diseño del Centro de Informatización Universitaria.
- Hacer extensivo el sistema a otras entidades.

## BIBLIOGRAFÍA REFERENCIADA

1. Capote Belina “La gestión de información como herramienta fundamental en el desarrollo de los centros toxicológicos”, Ciudad de La Habana, Cuba, (2003), [http://bvs.sld.cu/revistas/aci/vol11\\_2\\_03/aci030203.htm](http://bvs.sld.cu/revistas/aci/vol11_2_03/aci030203.htm)
2. Internet History 10 de febrero de 2009 . [En línea] [http://www.computerhistory.org/internet\\_history](http://www.computerhistory.org/internet_history) .
3. Culebro Juarez M, Gómez Herrera W, “*Software libre vs software propietario*”. 2006.
4. H. Fayol, Taylor, F.W (1916), “Administración Industrial y General Principio de la Administración Científica”, [En línea] <http://dspace.ucbscz.edu.bo/dspace/bitstream/123456789/2723/1/1705.pdf>.
5. “Popempresas”, [En línea] <http://queretaro.popempresas.com/post/32tf/software-para-administrar-flotillas-sofflot-ver-20> ..
6. “AndSoft”, *Software de Gestión de Transporte y Logística*. [En línea] [www.andorrasoft.com/INC/ESP/products.asp](http://www.andorrasoft.com/INC/ESP/products.asp)
7. “Simplexity”. [En línea] <http://www.simplexity.com.co/>
8. “Portalprogramas”. [En línea] <http://gratis.portalprogramas.com/>
9. Canós José H, Patricio Letelier ,Mª Carmen Penadés, “*Metodologías Ágiles en el Desarrollo de Software*” , DSIC-Universidad Politécnica de Valencia [En Línea] <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.4553&rep=rep1&type=pdf#page=9>.
10. González, Pilar, “Estudio de la aplicación de metodologías ágiles para la evolución de productos de Software”. s.l. : Universidad Politécnica de Madrid, 2008.
11. Jacobson, I. 1998. "Applying UML in the Unified Process" Presentación. Rational Software. Presentación disponible en <http://www.rational.com/uml> como UMLconf.zip
12. Díaz María, María Gabriela , Angélica Pérez. “Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistemática”. [En línea] <http://www.academia-interactiva.com/ise.fdf/>
13. Sánchez, María A [En línea] <http://www.willidev.net/InsiteCreation/v1.0/descargas/cualmetodología.pdf/>, junio de 2004.

14. Revista Máster Magazine. "Revista digital líder de la informática". [En línea]  
<http://www.mastermagazine.info/termino/7006.php> .
15. Rumbaugh, J. y Jacobson, I. y Booch, G. "El Lenguaje Unificado de Modelado", 2000.
16. Potencier, Fabien. "Symfony la guía definitiva", [En línea]  
<http://quijote.biblio.iteso.mx/catia/LibrosElectronicos/cat.aspx?cmn=view&id=289135>, 2008 .
17. Documentación Ext JS JavaScript Library . <http://extjs.com>. [En línea] 2008.  
<http://extjs.com/deploy/dev/docs>.
18. Solarte, Diego Samir. "Bases de datos con software libre Ingeniero de Sistemas". Facultad de Ingeniería. Universidad de Manizales : s.n.
19. B, Keber Flores, "*Implementación*", "Sistema de Reconocimiento de Patrones de actividad Neural Srpan". s.l. : Universidad de Santiago de Chile, 2008.
20. PostgreSQL. [En línea] <http://archives.postgresql.org/pgsql-es-fomento/2009-07/msg00000.php> .
21. Visual Paradigm. . [En línea] 2008. . <http://www.visual-paradigm.com>..
22. Rumbaugh, Jacobson, Booch, 2000

## BIBLIOGRAFÍA CONSULTADA

- Apache: <http://httpd.apache.org/>
- Bertino José F, "Perspectivas del comercio mundial". Equipo Federal del Trabajo, Año II, Revista nº 21, (2007), págs. 3-21, [http://www.eft.org.ar/pdf/eft2007\\_21pp3-27.pdf](http://www.eft.org.ar/pdf/eft2007_21pp3-27.pdf)
- Brito Kareny Acuña. "Selección de Metodologías de Desarrollo para Aplicaciones Web en la Facultad de Informática de la Universidad de Cienfuegos". Universidad de Cienfuegos. (2009). Pág. (38).
- Castro Fidel Ruz. Reflexión:"La Factura Petrolera y el Desarrollo". 19 de noviembre del 2007.
- Castro Fidel Diaz-Balart. "Ciencia, Tecnología y Sociedad hacia un desarrollo sostenible en la era de la Globalización". Pág. 283 (2004).
- Canós José H, Patricio Letelier, M<sup>a</sup> Carmen Penadés,"Metodologías Ágiles en el Desarrollo de Software", DSIC -Universidad Politécnica de Valencia.
- "Disciplina para la Administración de Proyectos MSF";  
<http://www.planetacursos.com/curso/MjkSMA/>
- La Web de Definiciones Legales <http://www.definicionlegal.com/> .
- Martínez Osvaldo, Ramón Pichs. "Temas de Economía Mundial". La Habana, Cuba (2006). <http://www.redem.buap.mx/ciem/temas11.pdf>, Pág (148).
- Melo Diego Samir. "Bases de datos con software libre", Facultad de Ingeniería, Universidad de Manizales. [mdiego@athenea.umanizales.edu.co](mailto:mdiego@athenea.umanizales.edu.co) .
- Palacios. Juan," Adaptando los Procesos de la Empresa, Flexibilidad con Scrum, Principios de Diseño y implantación de campos de Scrum" (2008).
- PetrolCap: <http://gratis.portalprogramas.com/PetrolCap.html>
- Potencier Fabien, Symphony la guía definitiva, (2008),  
<http://quijote.biblio.iteso.mx/catia/LibrosElectronicos/cat.aspx?cmn=view&id=289135>,
- Rodríguez Pilar González."Estudio de la aplicación de metodologías ágiles para la evolución de productos de Software", Facultad de Informática, Universidad Politécnica de Madrid (2008).
- Rumbaugh, J. Jacobson, I. y Booch, G."El Proceso Unificado de Desarrollo de Software". (2000). Pág (106- 110).
- Visual Paradigm: <http://www.visual-paradigm.com/> (2008).

### GLOSARIO DE TÉRMINOS

**API:** Interfaz de programación de aplicaciones (Applications Programming Interface): una serie de funciones que están disponibles para realizar programas para un cierto entorno.

**Base de Datos:** Colección de datos almacenados en un dispositivo informático no volátil, los datos deben estar disponibles permitiendo acciones como: adicionar, eliminar, actualizar y consultar; manteniendo su integridad y coherencia.

**BSD:** Licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). La licencia BSD al contrario que GPL\* permite el uso del código fuente en software no libre.

**CDA:** Modelos desarrollados en el Departamento de Energía donde se refleja el consumo de combustible por las actividades realizadas mensualmente (CDA\_001, CDA\_002).

**CSRF (*Cross-site request forgery*):** es un tipo de *exploit* (explotar) malicioso de un sitio web en el que comandos no autorizados son transmitidos por un usuario en el cual el sitio web confía. Esta vulnerabilidad es conocida también por otros nombres como XSRF, enlace hostil, ataque de un click, cabalgamiento de sesión, y ataque automático.

**CSS (Hojas de Estilo en Cascada):** son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML\* (y por extensión en XHTML). La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

**DOM (Modelo de Objetos de Documento):** Es una plataforma que proporciona un conjunto estándar de objetos a través de la cual se pueden crear documentos HTML y XML\*, navegar por su estructura y, modificar, añadir y borrar tanto elementos como contenidos. Al no apoyarse en un lenguaje de programación en particular, DOM facilita el diseño de páginas web activas, proporcionando una interfaz estándar para que otro *software* manipule los documentos.

**Framework:** Se caracteriza por simplificar el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes; proporciona también una estructura al código fuente y facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

**GNU:** Nombre otorgado por la *Free Software Fundación* a todo *software* que se distribuye con su código fuente y que el usuario puede adecuarlo de acuerdo con sus necesidades.

**GPL (Licencia Pública Genera):** Usar la GPL de GNU exige que todas las versiones mejoradas que se publiquen sean software libre.

**HTTP (Protocolo de Transferencia de Hipertexto):** Es el método más común de intercambio de información en la World Wide Web, es el método mediante el cual se transfieren las páginas web a un ordenador.

**HTTPs (Protocolo Seguro de Transferencia de Hipertexto):** URL\* creada por Netscape Communications Corporation para designar documentos que llegan desde un servidor WWW seguro. Esta seguridad es dada por el protocolo SSL\* (Secure Sockets Layer) basado en la tecnología de encryptación y autenticación desarrollada por la RSA Data Security Inc.

**HTML (Lenguaje de Marca de Hipertextos):** Es un lenguaje de programación muy sencillo que se utiliza para crear los textos y las páginas web, es justamente un lenguaje que se basa en las marcas para crear los hipertextos.

**Inyección SQL:** Es una vulnerabilidad informática en el nivel de la validación de las entradas a la base de datos\* de una aplicación. Su origen es el filtrado incorrecto de las variables utilizadas en las partes del programa con código SQL. Es, de hecho, un error de una clase más general de vulnerabilidades que pueden ocurrir en cualquier lenguaje de programación o de script que este incrustado dentro de otro.

**JavaScript:** Es un lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C.

**LGPL (Licencia Pública General Reducida de GNU):** LGPL es una licencia que es prácticamente igual a la GPL, pero permite que programas con esta licencia estén integrados en programas privativos, está adaptada para poder crear también aplicaciones privativas.

**MD5 (Algoritmo de Resumen del Mensaje 5):** es un algoritmo de reducción criptográfico de 128 bits ampliamente usado, diseñado por el profesor Ronald Rivest del MIT (Instituto Tecnológico de Massachusetts), en 1991. En PHP se utiliza la función MD5 (*Message Digest 5*), que es una función hash irreversible (de un sólo sentido), es decir, encripta el password tecleado por el usuario y es imposible que partiendo desde la cadena encriptada se vuelva a la contraseña origen. Por esto mismo no hay problema de que alguien pueda acceder al campo encriptado de la base de datos\*.

**SQL (Structured Query Language):** Lenguaje de Consulta Estructurado es un lenguaje declarativo de acceso a bases de datos relacionales

**SSL (Secure Socket Layer):** Es un protocolo desarrollado por Netscape Communications Corporation para dar seguridad a la transmisión de datos en transacciones comerciales en

Internet. Utilizando la criptografía de llave pública, SSL provee autenticación del servidor, encriptar de datos, e integridad de los datos en las comunicaciones cliente/servidor.

**TICs (Tecnologías de la Informática y las Comunicaciones):** Conjunto de servicios y tecnologías integradas a un sistema con el objetivo de mejorar los procesos y la calidad de vida.

**URL\* (Localizador de Recurso Uniforme):** La dirección global de documentos y otros recursos en la World Wide Web.

**WWW (*World Wide Web*):** Llevado al español su traducción significa Red Global Mundial. Es un sistema de documentos de hipertexto o hipermedios enlazados y accesibles a través de Internet.

**XML (Lenguaje de Marcas Extensible):** XML no es un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Es un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium (W3C)*.

**XSS (*Cross-site scripting*):** es un tipo de agujero de seguridad basado en la explotación de vulnerabilidades del sistema de validación de HTML incrustado. Las vulnerabilidades de XSS originalmente abarcaban cualquier ataque que permitiera ejecutar código de "*scripting*", como VBScript o JavaScript, en el contexto de otro sitio web.