

Universidad de las Ciencias Informáticas

Facultad 1



**Análisis y diseño del Sistema de Gestión de Almacenes en la Central
Termoeléctrica Renté**

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

Autor:

Yeriuska Esther Díaz Oliva

Tutores:

Ing. Alexey Peña Paz

Ing. Ilmaris Delia López Arias

Ciudad de La Habana

Junio 2010

Año 52 de la Revolución

*Cualquier tecnología suficientemente avanzada es
indistinguible de la magia.*

Arthur Clarke

A mi madre por el sacrificio, el amor, y el apoyo que me ha dado.

A mis hermanos, por ser ambos tan especiales.

A mi tía Gladita y Yane por todo su cariño.

A mis vecinos, familia y amigos, por su infinito apoyo.

A mi Madre:

Por ser la persona que más quiero, por ser mi ejemplo a seguir. A ella que supo ser Padre y Madre, y aunque hubo momentos difíciles, siempre estuvo ahí para mí en todo momento. A ella, por ser siempre ella: mi Madre. Gracias a ti, hoy veo realizado lo que hace poco era solo una ilusión.

Te quiero mamá.

A mis hermanitos por ser las personas que más quiero en la vida, por ser mis mejores mis amigos.

A mi tía Gladita, mi segunda mamá y mi prima Yane por su cariño tan especial y su confianza de siempre.

A Ernesto por su apoyo incondicional y por toda la ayuda que me ha brindado para salir adelante.

A mi familia por estar pendiente de mí cada día y por darme su apoyo incondicional.

A mis grandes amigas de todos los tiempos Yanni, Diliany, Ariamana, Leyany, Alba, Lianet, gracias por vivir pendientes de mí.

A todos los amigos que me ayudaron, me motivaron y que estuvieron presentes en cada momento, especialmente Yaité, Yanelys, Dayanna, Chanel, Yadira, Adianez, Yunaldís, Geidys, Papitín, Alex, Alain, Julito, muchas gracias por su linda amistad.

A mis tutores Alexey y Ilmaris por su paciencia y dedicación.

A Israel pues de no ser por tí, mi sueño nunca se hubiese hecho realidad.

Muchas gracias.

A Yuniel, Geídys, Mavi y Julito gracias por prestarme algo de su poco tiempo.

Muchas gracias

A nuestro Comandante en Jefe y a la Revolución Cubana por habernos dado la oportunidad de estudiar en la Universidad de las Ciencias Informáticas donde hicimos realidad nuestro gran sueño.

A todos los que de una forma u otra nos ayudaron para que el presente trabajo saliera adelante y me pudiera graduar, a todos muchas gracias.

Declaro ser la única autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año 2010.

Firma del Autor
Yeriuska Esther Díaz Oliva

Firma del Tutor
Alexey Peña Paz

Firma del Tutor
Ilmaris Delia López Arias

En la Central Termoeléctrica Renté (CTR) se realizan procesos asociados a la gestión de almacenes. Debido al creciente y constante volumen de información que se gestiona en esta área y dado que la mayoría se realizan usualmente de forma manual, se hace necesario automatizar gran parte de los procesos allí realizados.

En la CTR se encuentra actualmente un sistema informático que no satisface todas sus necesidades, pues ha quedado prácticamente obsoleto debido a variaciones en los procesos del negocio y a la creciente interoperabilidad requerida por más de un usuario. Teniendo en cuenta las dificultades descritas, la presente investigación persigue como objetivo el diseño de un nuevo sistema capaz de cumplir con los requerimientos funcionales deseados. Para realizar el diseño de la aplicación propuesta, se utilizó como metodología, el Proceso Unificado de Desarrollo (RUP, por sus siglas en inglés), como herramienta CASE: Visual Paradigm, como lenguaje de programación PHP, como framework el Symfony, la librería ExtJS, y como herramientas: Visual Paradigm, PostgreSQL, NetBeans y el servidor web Apache. El diseño del sistema propuesto permitirá un mejor control de todos materiales relacionado con el almacén, debido a la centralización de toda la información requerida, así como el fácil y rápido acceso a la misma.

Palabras Claves: Gestión de almacenes, información.

Tabla de contenido

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 INTRODUCCIÓN.....	6
1.2 CONCEPTOS BÁSICOS RELACIONADOS CON EL DOMINIO DEL PROBLEMA	6
1.3 SISTEMAS INFORMÁTICOS RELACIONADOS CON LA GESTIÓN DE ALMACENES	7
1.3.1 LOS SISTEMAS DE GESTIÓN DE ALMACENES EN EL MUNDO	7
1.3.2 LOS SISTEMAS DE GESTIÓN DE ALMACENES EN CUBA.....	8
1.4 TENDENCIAS ACTUALES DEL SOFTWARE LIBRE	9
1.5 HERRAMIENTAS, LENGUAJES Y METODOLOGÍAS A UTILIZAR.....	9
1.5.1 METODOLOGÍAS DE DESARROLLO DEL SOFTWARE	10
1.5.2 LENGUAJE UNIFICADO DE MODELADO (UML)	12
1.5.3 HERRAMIENTA CASE PARA EL MODELADO	12
1.5.4 FUNDAMENTACIÓN DE LA METODOLOGÍA Y HERRAMIENTA DE MODELADO A UTILIZAR	13
1.6 HERRAMIENTAS, TECNOLOGÍAS Y LENGUAJES PARA EL DESARROLLO.....	14
1.6.1 FRAMEWORK.....	14
1.6.2 OBJETO RELACIONAL DE MAPEO	16
1.6.3 SERVIDOR WEB APACHE.....	17
1.6.4 LENGUAJES PARA EL DESARROLLO WEB.....	17
1.6.5 SISTEMAS GESTORES DE BASE DE DATOS (SGBD)	21
1.6.6 ENTORNOS INTEGRADOS DE DESARROLLO.....	22
1.6.7 FUNDAMENTACIÓN DE LAS TECNOLOGÍAS, LENGUAJES Y HERRAMIENTAS A UTILIZAR.....	23
1.7 CONCLUSIONES.....	24
CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA	25
2.1 INTRODUCCIÓN.....	25
2.2 MODELO DEL NEGOCIO	25
2.2.1 ANÁLISIS DE LOS PROCESOS DEL NEGOCIO.....	25
2.2.2 REGLAS DEL NEGOCIO A CONSIDERAR.....	26
2.2.3 ACTORES DEL NEGOCIO	28
2.2.4 TRABAJADORES DEL NEGOCIO	28
2.2.5 DIAGRAMA DE CASOS DE USO DEL NEGOCIO	29
2.2.6 REALIZACIÓN DE CASOS DE USO DEL NEGOCIO	29
2.2.7 DIAGRAMA DE ACTIVIDADES	37
2.2.8 MODELO DE OBJETO	38
2.3 ESPECIFICACIÓN DE LOS REQUISITOS EL SISTEMA.	39
2.3.1 REQUISITOS FUNCIONALES.....	39
2.3.2 REQUISITOS NO FUNCIONALES	40
2.4 MODELO DEL SISTEMA	43

2.4.1	ACTORES DEL SISTEMA	43
2.4.2	DIAGRAMA DE CASOS DE USO DEL SISTEMA	44
2.4.3	EXPANSIÓN DE LOS CASOS DE USO DEL SISTEMA	44
2.5	CONCLUSIONES.....	48
CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA.		49
2.1	INTRODUCCIÓN.....	49
2.2	ESTIMACIÓN DEL ESFUERZO.....	49
2.3	ANÁLISIS.....	56
2.3.1	DIAGRAMA DE CLASES DEL ANÁLISIS.....	56
2.3.2	DIAGRAMA DE INTERACCIÓN	57
2.4	DISEÑO	58
2.4.1	PATRONES DEL DISEÑO.....	58
2.4.2	DIAGRAMAS DE CLASES DEL DISEÑO	61
2.5	DISEÑO DE LA BASE DE DATOS.....	62
2.6	DIAGRAMA DE DESPLIEGUE.....	63
2.7	CONCEPCIÓN GENERAL DE AYUDA.....	64
2.8	TRATAMIENTO DE ERRORES	64
2.9	CONCLUSIONES.....	65
CONCLUSIONES.....		66
RECOMENDACIONES		67
BIBLIOGRAFÍA		68
TRABAJOS CITADOS		71
GLOSARIO DE TÉRMINOS		74

Tabla 1: Actores del negocio	28
Tabla 2: Trabajadores del negocio	28
Tabla 3: CUN Registrar entrada de materiales.....	30
Tabla 4: CUN Realizar reclamación al proveedor.....	32
Tabla 5: CUN Realizar solicitud de materiales.	33
Tabla 6: CUN Realizar devolución de materiales.	34
Tabla 7: CUN Realizar movimiento de materiales.	35
Tabla 8: CUN Realizar Solicitud de Compra de Materiales.	36
Tabla 9: CUN Realizar Solicitud de Información de Materiales.....	37
Tabla 10: Actores del sistema.....	43
Tabla 11: CUS Registrar entrada de materiales.	45
Tabla 12: CUS Rebajar materiales.	45
Tabla 13: CUS Buscar materiales.	45
Tabla 14: CUS Gestionar Tarjeta de Estiba.	45
Tabla 15: CUS Gestionar solicitud de materiales.	46
Tabla 16: CUS Gestionar Vale de Entrega de materiales.	46
Tabla 17: CUS Realizar devolución de materiales.....	46
Tabla 18: CUS Realizar reclamación al proveedor.	46
Tabla 19: CUS Gestionar Informe de Faltante de Materiales.	47
Tabla 20: CUS Gestionar Solicitud de Compra de Materiales.	47
Tabla 21: CUS Generar reportes.	47
Tabla 22: CUS Autenticar usuario.....	48
Tabla 23: CUS Gestionar usuario.	48
Tabla 24: CUS Gestionar rol.....	48
Tabla 25: Factor de Peso de los Casos de Uso sin ajustar	50
Tabla 26: Complejidad de los CU según sus Transacciones.....	51
Tabla 27: Factor de Complejidad Técnica.	52
Tabla 28: Factor de Ambiente	53
Tabla 29: Esfuerzo por Actividades.....	55
Tabla 31: Clases del análisis.....	56

Figura 1: Características metodologías ágiles y tradicionales.	10
Figura 2: Diagrama de Casos de Uso del Negocio.	29
Figura 3: Modelo de objeto del negocio.	38
Figura 4: Diagrama de CUS del Sistema	44
Figura 5: Diagrama de clases del análisis del CUS Registrar entrada de materiales	57
Figura 6: Diagrama de colaboración del CUS Registrar entrada de materiales.	58
Figura 7: Diagrama de clases del diseño del CUS Registrar entrada de materiales.	62
Figura 8: Modelo de la base de datos	63
Figura 9: Diagrama de despliegue	64

Introducción

En la actualidad, debido a la complejidad de las actividades y procesos que tienen lugar en una organización, se requiere del uso de herramientas que faciliten la gestión y el control de las operaciones y transacciones entre las distintas áreas. Por tal motivo, existen en el mercado multitud de aplicaciones que se encargan de dirigir y controlar tareas. Dentro de una empresa los sistemas que se encargan de llevar el control de todos sus recursos, desempeñan un papel muy importante, pues facilitan el trabajo del personal encargado de administrar dichos recursos y agilizan todos los procesos que se realizan dentro de la misma. (1)

Desde el triunfo de la Revolución uno de los principales objetivos del estado socialista, ha sido la superación cultural del pueblo, para ello se le ha concedido prioridad máxima a la socialización de las Tecnologías de la Informática y las Comunicaciones (TIC). Cuando se hace referencia a la informatización de la sociedad cubana, se refiere a la aplicación ordenada y masiva de las tecnologías de la informática, teniendo un uso racional y adecuado para lograr una mayor generación de riquezas.

En este proceso complejo de cambio estructural de las sociedades modernas se puede destacar, al menos, una dimensión: la generación de nuevos avances científicos y, especialmente, la difusión de nuevas tecnologías, singularmente, las TIC; esto hace que el proceso de informatización de las empresas esté convirtiéndose en un hecho cada vez más necesario. Actualmente se está iniciando un proceso encaminado a la fusión e integración a través de redes de empresas, que utilizan las TIC con el objetivo de generar sistemas empresariales, altamente productivos y flexibles, con el propósito de mejorar su eficacia y desarrollar sus actividades comerciales. (2)

Es en este punto donde los Sistemas de Gestión del Almacén (SGA) adquieren una especial relevancia. Este método permite grandes beneficios como son: seguridad, centralización, organización de la información y maximización de la productividad de la empresa. También ayudará a la empresa a tener más tiempo para las principales funciones de gestión comercial que necesita el negocio, lográndose minimizar los errores que ocurren cuando se trabaja de forma manual.

En la Central Termoeléctrica Renté se han utilizado varios sistemas para la gestión de los recursos en los

almacenes, pero estos no han sido de gran utilidad porque no dan solución a todos los procesos que se manejan en la administración de dichos recursos, por tal motivo en muchas ocasiones el trabajo se realiza de forma manual, archivando toda la información generada en formato duro, con el objetivo de poseer un registro histórico de los procesos no informatizados. Esta medida puede traer las siguientes consecuencias: inconsistencia de la información debido a que la misma no se encuentra centralizada, dificultad a la hora de generar reportes estadísticos fiables, deterioro de la documentación por factores como la humedad, gasto excesivo de recursos, repercutiendo todo esto desfavorablemente en el tiempo que se necesita para realizar la tramitación. Además, Renté utiliza actualmente un sistema propietario que fue comprado en otro país y por el que tiene que pagar licencia de uso.

Dada la situación expuesta anteriormente se plantea como **Problema a resolver:**

¿Cómo realizar el análisis y diseño de un sistema para la gestión de los almacenes de la Central Termoeléctrica Renté?

Se define como **objeto de estudio:** los procesos de gestión de almacenes, donde el **campo de acción** estaría enmarcado en los procesos de gestión de los almacenes en la Central Termoeléctrica Renté.

Para resolver el problema que se refleja en la situación anterior se plantea como **objetivo general:** elaborar el análisis y diseño de un sistema para la gestión de almacenes en la Central Termoeléctrica Renté. De ahí se derivan los siguientes **objetivos específicos:**

- ✓ Determinar aspectos teóricos-conceptuales sobre sistemas de gestión relacionados con los almacenes.
- ✓ Delimitar las necesidades del cliente para el análisis y diseño del SGA en la Central Termoeléctrica Renté.
- ✓ Analizar y diseñar el sistema de almacenes en la Central Termoeléctrica Renté.

Para guiar la investigación se traza la siguiente **idea a defender:** la realización del análisis y diseño del sistema de gestión de almacenes en la Central Termoeléctrica Renté facilitará y contribuirá a la posterior implementación del mismo.

Para dar cumplimiento al objetivo planteado se han propuesto las siguientes **tareas para la investigación**:

- ✓ Caracterización de algunos de los sistemas existentes en Cuba y el Mundo para la gestión de los almacenes, haciendo énfasis en los procesos de gestión que se analizan en los mismos.
- ✓ Realización de un estudio y selección de metodologías, tecnologías y herramientas para el desarrollo del trabajo.
- ✓ Realización de un estudio y descripción de los procesos de gestión de los almacenes en la Central Termoeléctrica Renté.
- ✓ Determinación de las funcionalidades y restricciones que el sistema debe de cumplir.
- ✓ Realización del análisis y diseño del sistema para la gestión de los almacenes en la Central Termoeléctrica Renté.

Para dar cumplimiento a las tareas y objetivos trazados se utilizaron **métodos científicos** empleados en el transcurso de la investigación, que permitieron observar el problema cuidadosamente y con objetividad.

Métodos teóricos

Método histórico-lógico: en esta investigación dicho método se usa con el objetivo de estudiar todo lo referente a los procesos de gestión de almacenes en el ámbito nacional e internacional, para de esta forma conocer más del tema, sus características e importancia. Este método ayuda a dar cumplimiento al estudio del estado del arte de los procesos de gestión de almacenes.

Método de la modelación: partiendo de todo lo investigado y aprendido se realizan los modelos correspondientes al ciclo de vida del software, que ayudan a dar cumplimiento a las tareas de diseño de los procesos involucrados en la solución.

Análisis-sintético: este método se usa para organizar y sintetizar toda la información obtenida del estudio del estado del arte, las tecnologías, metodologías, lenguajes y herramientas propuestas para la solución e identificar los principales problemas del sistema actual. La utilización de este método permite comprender mejor toda la información anterior, la que será de gran utilidad para lograr un adecuado análisis y diseño del sistema.

Métodos empíricos

Entrevista: este método permite obtener información a partir de experiencias, puntos de enfoque y posibles ideas de personas con experiencia en el objeto de investigación, ayuda a definir mejor los objetivos y el desarrollo del trabajo.

Método de la observación: permite investigar los procesos externamente sin tener que llegar a la esencia de los mismos, lo que sirve de ayuda para realizar un seguimiento del comportamiento, sirviendo como guía para conocer si se retrocede o se avanza hacia el objetivo final.

Con esta investigación se prevé obtener una propuesta de diseño para un sistema informático capaz de gestionar e integrar la información que se genera en la Termoeléctrica Renté de manera rápida y organizada, elevándose así la eficiencia y rapidez en los procesos que en esta se llevan a cabo.

Para una mejor comprensión del presente documento, la estructura del contenido queda conformada de la siguiente manera:

Capítulo 1: “Fundamentación teórica”: este capítulo contiene la fundamentación teórica del tema. Se realiza un estudio del estado del arte en el ámbito internacional y nacional. Se muestran los principales conceptos manipulados en el transcurso de la investigación, se hace referencia a las metodologías de desarrollo, los lenguajes de modelado y herramientas, seleccionando las que serán utilizadas en el desarrollo del presente trabajo.

Capítulo 2: “Características del sistema”: este capítulo cuenta con una amplia descripción del negocio, presentando los diagramas y casos de uso correspondientes a dicho modelado de negocio para de esta manera identificar las mejoras potenciales en la organización, y comprender el contexto donde será implantado el sistema. Luego se capturan los requisitos funcionales y no funcionales para la realización del modelo de casos de uso del sistema.

Capítulo 3: “Análisis y diseño del sistema”: este capítulo se modelan los diagramas de clases del análisis y del diseño, así como los diagramas de interacción correspondientes. Se hace referencia a la arquitectura y patrones utilizados para la construcción del diseño propuesto. Se muestra el modelo de la

base de datos, y el diagrama de despliegue. Además se realiza la estimación del esfuerzo por el método de análisis de Puntos de Casos de Uso.

Capítulo 1: Fundamentación teórica

1.1 Introducción

Con el desarrollo de este capítulo se realizará un estudio del estado del arte a nivel mundial y nacional. Además, se analizarán los principales conceptos relacionados con el campo de acción y los posibles lenguajes de programación, herramientas y metodologías de desarrollo de software a utilizar en la construcción del sistema, especificándose sus principales características y las ventajas de su utilización.

1.2 Conceptos básicos relacionados con el dominio del problema

Gestión: Se refiere a la acción de administrar o gestionar un negocio. A través de una gestión se llevarán a cabo diversas diligencias, trámites, las cuales conducirán al logro de un objetivo determinado. Por lo tanto, la misma estará orientada a resolver un problema específico, a concretar un proyecto, un deseo. (3)

Sistema de gestión: Un sistema de gestión es una estructura probada para la gestión y mejora continua de las políticas, los procedimientos y procesos de la organización, que logra los objetivos predeterminados de una institución, mediante una serie de estrategias, que incluyen la optimización de procesos, el enfoque centrado en la gestión y el pensamiento disciplinado.

La implementación de un sistema de gestión eficaz puede ayudar a:

- ✓ Mejorar la efectividad operativa.
- ✓ Reducir costos.
- ✓ Aumentar la satisfacción de clientes y partes interesadas.
- ✓ Aumentar la confiabilidad de la empresa.
- ✓ Lograr mejoras continuas.
- ✓ Potenciar la innovación.
- ✓ Eliminar las barreras al comercio.
- ✓ Aportar claridad al mercado.

Por lo tanto se puede decir que el uso de un sistema de gestión probado permitirá renovar constantemente los objetivos, estrategias, operaciones y niveles de servicio en una organización. (4)

1.3 Sistemas informáticos relacionados con la gestión de almacenes

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 de almacenes, en el ámbito internacional y nacional, es un factor clave para la realización de un nuevo sistema.

1.3.1 Los sistemas de gestión de almacenes en el mundo

Sistema de gestión de almacenaje por radio frecuencia (ADAIA)

ADAIA es un sistema de gestión para centros de distribución (C.D.) y almacenaje basado en la aplicación de las tecnologías de códigos de barras y terminales de radio frecuencia. Este sistema registra en tiempo real todos los movimientos dentro del almacén, recoge los datos de sus operarios y la mercadería, procesa dicha información y la utiliza para realizar una gestión óptima del C.D., obteniendo el máximo rendimiento de los recursos.

Beneficios:

- ✓ Dirige y gobierna los movimientos de los recursos.
- ✓ Maximiza el espacio disponible en el C.D.
- ✓ Aprovecha el tiempo al máximo.
- ✓ Agiliza la gestión.
- ✓ Proporciona información exacta y control en tiempo real.
- ✓ Mejora la calidad de servicio. (5)

CASA & IDEAS CHILE

Es un sistema de gestión de almacenes que permite optimizar las actividades de almacén, incluido el procesamiento de entrada y de salida, el inventario físico, la gestión de instalaciones y el almacenamiento.

También permite optimizar los traslados internos y el almacenamiento de mercancías, así como planificar y ejecutar el inventario físico. Para ayudar a maximizar el uso y la productividad del personal de almacén, este proporciona la funcionalidad de gestión de la mano de obra, con el objetivo de analizar el rendimiento de los empleados y compararlo con estándares de mano de obra y otras métricas de rendimiento. (6)

Los sistemas antes mencionados brindan un correcto manejo y control de los recursos de los almacenes pero para su uso deben ser adquiridos mediante la compra del producto, en la que se obtiene la licencia y los permisos del propietario. Esto trae como consecuencia un gasto adicional que tendría que hacer el país para obtener estos beneficios, con el fin de lograr un mejor control de los medios dedicados que se encuentran en el almacén.

1.3.2 Los sistemas de gestión de almacenes en Cuba

Talmacen: es el sistema de gestión de almacén que se encuentra utilizando actualmente la Central Termoeléctrica pero este no satisface todas las necesidades establecidas en Renté debido a que los requerimientos identificados en sus inicios no se corresponden con los procesos de negocio que se desarrollan en estos momentos, ya que este software fue comprado a empresas mexicanas hace varios años.

CementoSig: es un sistema de gestión de almacenes que está siendo utilizado actualmente en la Fábrica de Cemento de Siguaney, que se encuentra ubicada en la provincia de Sancti Spíritus. Este software dentro de sus principales funciones tiene la de automatizar las entradas del almacén, presentar informes de recepción, conexiones automáticas al departamento de contabilidad y automatizar el despacho del almacén.

Después del análisis de las propuestas anteriores se decide que no es conveniente adoptar ninguno de estos sistemas de gestión ya que son ineficientes dadas las necesidades actuales de la Central Termoeléctrica Renté, pues contienen términos, clasificaciones y automatiza procesos, que no son los mismos que allí se desarrollan.

Por tal motivo, el objetivo del presente trabajo está encaminado a obtener una propuesta de solución que permita mejorar aquellas características que dificultan el uso de los ya existentes, primeramente debe estar desarrollado sobre tecnologías web que garanticen su versatilidad en la gestión de la información

independientemente del sistema operativo usado y como requisito indispensable debe automatizar la gestión de los procesos o eventos que se desarrollan en el almacén, en la forma en que se realizan actualmente.

1.4 Tendencias actuales del software libre

Software libre se refiere a la libertad de los usuarios para ejecutar, copiar, distribuir, estudiar, cambiar y mejorar el software. De modo más preciso, se refiere a cuatro libertades de los usuarios del software:

- ✓ La libertad de usar el programa, con cualquier propósito (libertad 0).
- ✓ La libertad de estudiar el funcionamiento del programa, y adaptarlo a las necesidades (libertad 1).
- ✓ El acceso al código fuente es una condición previa para esto. La libertad de distribuir copias, con lo que puede ayudar a otros (libertad 2).
- ✓ La libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la comunidad se beneficie (libertad 3).

Resumiendo, la libertad para utilizar un programa significa que cualquier individuo u organización podrán ejecutarlo desde cualquier sistema informático, con cualquier fin y sin la obligación de comunicárselo subsiguientemente ni al desarrollador ni a ninguna entidad en concreto. De modo que puedes pagar o no por obtener copias de software libre, pero independientemente de la manera en que las obtengas, siempre tendrás libertad para copiar, modificar e incluso vender estas copias. (7)

1.5 Herramientas, lenguajes y metodologías a utilizar

Las especializaciones del software están impulsando cada vez más el desarrollo de las tecnologías y metodologías que se utilizan en este propósito, a nivel mundial existen herramientas que permiten la realización de grandes sistemas en tiempos breves y con gran calidad debido a las facilidades que implementan.

Como consecuencia de lo antes mencionado, a continuación se presenta un resumen realizado durante la investigación donde se analizan metodologías de desarrollo, lenguajes de programación, y gestores de bases de datos que serán de gran importancia para el desarrollo del trabajo de diploma.

1.5.1 Metodologías de desarrollo del software

Desarrollar un buen software depende de un sinnúmero de actividades y etapas, donde el impacto de elegir la mejor metodología para un equipo en un determinado proyecto, es trascendental para el éxito del producto. El papel preponderante de las metodologías es sin duda esencial en un proyecto y el paso inicial, que debe encajar en el equipo, es guiar y organizar actividades que conlleven a las metas trazadas en el grupo.

En el presente trabajo se detallan dos grandes enfoques, metodologías tradicionales y metodologías ágiles, las primeras recalcan el uso exhaustivo de documentación durante todo el ciclo del proyecto, mientras que las segundas ponen vital importancia en la capacidad de respuesta a los cambios y al mantener una buena relación con el cliente para llevar al éxito el proyecto. (8)

A continuación se muestran las principales características de ambas metodologías:

Metodologías Ágiles	Metodologías Tradicionales
Basadas en heurísticas provenientes de prácticas de producción de código	Basadas en normas provenientes de estándares seguidos por el entorno de desarrollo
Especialmente preparados para cambios durante el proyecto	Cierta resistencia a los cambios
Impuestas internamente (por el equipo)	Impuestas externamente
Proceso menos controlado, con pocos principios	Proceso mucho más controlado, con numerosas políticas/normas
No existe contrato tradicional o al menos es bastante flexible	Existe un contrato prefijado
El cliente es parte del equipo de desarrollo	El cliente interactúa con el equipo de desarrollo mediante reuniones
Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio	Grupos grandes y posiblemente distribuidos
Pocos artefactos	Más artefactos
Pocos roles	Más roles
Menos énfasis en la arquitectura del software	La arquitectura del software es esencial y se expresa mediante modelos

Figura 1: Características metodologías ágiles y tradicionales.

Metodologías tradicionales

Proceso Unificado de Desarrollo (RUP): El Proceso Unificado no es simplemente un proceso, sino un marco de trabajo extensible que puede ser adaptado a organizaciones o proyectos específicos. De la misma forma, RUP también es un marco de trabajo extensible, por lo que muchas veces resulta imposible decir si un refinamiento particular del proceso ha sido derivado del Proceso Unificado o del RUP. Por dicho motivo, los dos nombres suelen utilizarse para referirse a un mismo concepto.

Provee un enfoque disciplinado en la asignación de tareas y responsabilidades dentro de una organización de desarrollo. Su meta es asegurar la producción de software de muy alta calidad que satisfaga las necesidades de los usuarios finales, dentro de un calendario y presupuesto predecible. (9)

El proceso de software propuesto por RUP presenta tres características esenciales: **Anexo I**

Metodologías ágiles

Extreme Programming (XP)

La Programación Extrema es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Se basa en: realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Además, utiliza la técnica historia del usuario para especificar los requisitos del software. (10)

Scrum

Scrum es un proceso ágil y liviano que sirve para administrar y controlar el desarrollo de software. El desarrollo se realiza en forma iterativa e incremental (una iteración es un ciclo corto de construcción repetitiva). Scrum se focaliza en priorizar el trabajo en función del valor que tenga para el negocio, maximizando la utilidad de lo que se construye y el retorno de inversión. Está diseñado especialmente para adaptarse a los cambios en los requerimientos, por ejemplo en un mercado de alta competitividad. En Scrum el equipo se focaliza en una única cosa: construir software de calidad. Por el otro lado, la

gestión de un proyecto Scrum se focaliza en definir las características que debe tener el producto a construir (qué construir, qué no y en qué orden) y en remover cualquier obstáculo que pudiera entorpecer la tarea del equipo de desarrollo. Se busca que los equipos sean lo más efectivos y productivos posible. (8)

1.5.2 Lenguaje Unificado de Modelado (UML)

UML es ante todo un lenguaje que proporciona un vocabulario y reglas, para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema. Este nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- ✓ Visualizar: UML permite expresar de una forma gráfica un sistema de forma que otro lo puede entender.
- ✓ Especificar: UML permite especificar las características de un sistema antes de su construcción.
- ✓ Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.
- ✓ Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión. (11)

1.5.3 Herramienta CASE para el modelado

En la actualidad existen infinitudes de herramientas de modelado, entre las que se encuentran: *Umbrello*, *Visual Paradigm* y *Rational Rose Enterprise Edition*.

Rational Rose

Rational Rose es la herramienta CASE desarrollada por los creadores de UML (Booch, Rumbaugh y Jacobson). Cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables. Permite completar diferentes disciplinas (flujos fundamentales) de RUP; incluye, a su vez,

herramientas de ingeniería inversa y generación de código que facilitan el tránsito hasta el producto final. Rational Rose es considerado una de las mejores herramientas para traducir requisitos de alto nivel a una arquitectura basada en componentes. Esta herramienta está avanzada en cuanto al desarrollo de UML por lo que se ha convertido en una de las mejores opciones por la notación estándar que se provee para especificar, visualizar, construir productos de software y sistemas. (12)

Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Algunas de las características más atractivas de Visual Paradigm es que permite generar a partir del modelado código PHP, realiza ingeniería tanto directa como inversa y genera la documentación del proyecto automáticamente en varios formatos como Web o PDF, y además facilita a través de la herramienta *Team*(equipo) Framework el trabajo colaborativo. Es una herramienta multiplataforma que se puede utilizar tanto en Linux como en Windows, por lo que cumple con las políticas de migración a software libre. Tiene una interfaz muy intuitiva y es de fácil aprendizaje para los desarrolladores. Permite la generación automática de diagramas a partir de las descripciones de los casos de uso, por ejemplo diagramas de secuencia, permitiendo la agilidad en el trabajo del analista. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. (13)

1.5.4 Fundamentación de la metodología y herramienta de modelado a utilizar

Se considera a RUP dentro de las metodologías estudiadas como la más óptima para modelar el sistema debido a las grandes potencialidades que presenta para el análisis, implementación y documentación de sistemas orientados a objetos. Está diseñado para adaptarse a las necesidades del proyecto o institución que lo esté utilizando. Otra de las características favorables que ofrece RUP es que permite mantener desde sus propios inicios una documentación profunda y detallada de todo el proceso en sentido general, además, planifica todas las etapas del desarrollo, lo que es de gran importancia para el trabajo con equipos grandes y para la realización de proyectos complejos. Su característica de ser iterativo e incremental permitirá ir perfeccionando el software en cualquier momento del desarrollo teniendo en cuenta la variabilidad de las necesidades del cliente.

Se decidió utilizar como herramienta de modelado para dicho sistema, el Visual Paradigm, ya que a pesar de ser un software privativo, la UCI adquirió la licencia y lo tiene como política de herramienta CASE a utilizar. Por sus características, cubre todo el ciclo de vida de un proyecto, permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable.

1.6 Herramientas, tecnologías y lenguajes para el desarrollo

1.6.1 Framework

Un framework simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados para resolver las tareas comunes. Además, le proporciona estructura al código fuente, forzando al desarrollador a crear código más legible y más fácil de mantener. Por último, este facilita la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas. (14)

Symfony

Symfony es un completo framework diseñado para optimizar, gracias a sus características, el desarrollo de las aplicaciones web. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona varias herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación. Este framework está desarrollado completamente con PHP 5 y es compatible con la mayoría de gestores de bases de datos, como MySQL, PostgreSQL, Oracle y SQL Server de Microsoft. Se puede ejecutar tanto en plataformas Unix, Linux como en plataformas Windows. Está basado en el patrón MVC el que en Symfony probé 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(ayudantes)* 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. A continuación se hará referencia a las características más detalladas de este framework.

Symfony se diseñó para que se ajustara a los siguientes requisitos:

- ✓ Fácil de instalar y configurar en la mayoría de las plataformas.

- ✓ Independiente del sistema gestor de bases de datos.
- ✓ Sencillo de usar en la mayoría de casos, pero lo suficientemente flexible como para adaptarse a los casos más complejos.
- ✓ Sigue la mayoría de mejores prácticas y patrones de diseño para la web.
- ✓ Preparado para aplicaciones empresariales y adaptables a las políticas y arquitecturas propias de cada empresa, además de ser lo suficientemente estable como para desarrollar aplicaciones a largo plazo.
- ✓ Fácil de extender, lo que permite su integración con librerías desarrolladas por terceros. (14)

Zend framework

Es un framework de código abierto para PHP desarrollado por Zend, empresa encargada de la mayor parte de las mejoras hechas a PHP, por lo que se podría decir que es el framework “oficial”.

Es un framework para desarrollo de aplicaciones Web y servicios Web con PHP, brinda soluciones para construir sitios web modernos, robustos y seguros. Trabaja con PHP 5. Presenta una ventaja y es que es desarrollado por Zend, empresa que respalda comercialmente a PHP. El marco de Zend también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL, además completa documentación y *tests (pruebas)* de alta calidad y presenta robustas clases para autenticación y filtrado de entrada.

Zend Framework implementa el patrón MVC, es 100% orientado a objetos y sus componentes tienen un bajo acoplamiento por lo que los puedes usar en forma independiente. Un punto importante es que brinda un estándar de codificación que se debería seguir en los proyectos. (15)

ExtJS

Según la definición ExtJS es una librería JavaScript que permite construir aplicaciones complejas en Internet. Esta librería incluye:

- ✓ Componentes UI (Interfaz de Usuario) del alto performance y personalizables.

- ✓ Modelo de componentes extensibles.
- ✓ Licencias *Open Source* (código abierto) y comerciales.

ExtJS es un motor que permite crear aplicaciones mediante JavaScript. Una de las grandes ventajas de utilizar ExtJS es que nos permite crear aplicaciones complejas utilizando componentes predefinidos así como un manejador de layouts (capa) similar al que proporciona *Java Swing*, gracias a esto proporciona una experiencia consistente sobre cualquier navegador, evitando el tedioso problema de validar que el código escrito funcione bien en cada uno (Firefox, IE, Safari). Además la ventana flotante que facilita ExtJS es excelente por la forma en la que funciona. Al moverla o redimensionarla solo se dibujan los bordes haciendo que el movimiento sea fluido lo que le da una ventaja tremenda frente a otros.

Usar un motor de *render* como ExtJS nos permite tener además estos beneficios:

- ✓ Existe un balance entre Cliente – Servidor. La carga de procesamiento se distribuye, permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo.
- ✓ Comunicación asíncrona. En este tipo de aplicación el motor de *render* puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente se de cuenta.
- ✓ Eficiencia de la red. El tráfico de red puede disminuir al permitir que la aplicación elija que información desea transmitir al servidor y viceversa, sin embargo la aplicación que haga uso de la pre-carga de datos puede que revierta este beneficio por el incremento del tráfico. (16)

1.6.2 Objeto Relacional de Mapeo

Doctrine

Doctrine es un potente y completo sistema ORM (acrónimo en inglés de *Object Relational Mapper*, en español *Objeto Relacional de Mapeo*) para PHP 5.2. Entre muchas otras cosas brinda la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos.

Su principal ventaja radica en poder acceder a la base de datos utilizando la programación orientada a objetos (POO) debido a que doctrine utiliza el patrón *Active Record* (Récord Activo) para manejar la base de datos, tiene su propio lenguaje de consultas y trabaja de manera rápida y eficiente. Es fácilmente integrado a los principales framework de desarrollo utilizados actualmente, por lo que se propone su uso. (17)

1.6.3 Servidor web Apache

Apache es el servidor web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Apache es una muestra, al igual que el sistema operativo Linux (un Unix desarrollado inicialmente para PC), de que el trabajo voluntario y cooperativo dentro de Internet es capaz de producir aplicaciones de calidad profesional difícil de igualar.

Este software libre es grandemente reconocido en muchos ámbitos empresariales y tecnológicos, por algunas razones:

- ✓ Corre en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal.
- ✓ Es una tecnología gratuita de código fuente abierta.
- ✓ Es un servidor altamente configurable de diseño modular.
- ✓ Trabaja con PHP y otros lenguajes de script.
- ✓ Permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Es posible configurar Apache para que ejecute un determinado script cuando ocurra un error en concreto. (18)

1.6.4 Lenguajes para el desarrollo Web

El desarrollo de aplicaciones Web ha tenido un auge en todo el mundo gracias a las ventajas que las mismas le ofrecen a las empresas. En la actualidad los lenguajes de programación para la Web se clasifican en dos grupos: lenguajes del lado del cliente y lenguajes del lado del servidor. Las técnicas de desarrollo Web y lenguajes del lado del cliente más utilizados son HTML y JavaScript, mientras que los lenguajes de programación del lado del servidor más usados en software libre son PHP, JSP y ASP.

PHP

PHP usa una mezcla entre interpretación y compilación para intentar ofrecer a los programadores la mejor mezcla entre rendimiento y flexibilidad.

PHP compila para tu código una serie de instrucciones (llamadas *opcodes*) siempre sea accedida. Estas instrucciones son entonces ejecutadas una por una hasta que el script termina. Esto es diferente a la manera convencional de compilación de lenguajes como C++ donde el código es compilado a código ejecutable que es después ejecutado. PHP es recompilado cada vez que se solicita un script.

Proporciona una regeneración muy rápida durante el desarrollo. Si tienes errores en el código, PHP rechazará compilar la página hasta que arregles el problema, y tendrás que ejecutar tu código línea a línea hasta encontrar el problema.

Una ventaja importante de interpretar el código es que toda la memoria usada por tu código es manejada por PHP, y el lenguaje automáticamente vacía esta memoria cuando el *script (guión)* finaliza. Esto significa que no tienes que preocuparte de las conexiones a la base de datos, porque PHP lo hará por ti. Tampoco quiere decir que tu debes dejar hacer todo a PHP, los buenos programadores hacen esas funciones programándolas en código teniendo un mayor control del *script*. PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como Linux o Mac OS X) y Windows, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

Ventajas:

- ✓ Muy fácil de aprender.
- ✓ Se caracteriza por ser un lenguaje muy rápido.
- ✓ Soporta en cierta medida la orientación a objeto. Clases y herencia.
- ✓ Es un lenguaje multiplataforma: Linux, Windows, entre otros.
- ✓ Capacidad de conexión con la mayoría de los manejadores de base de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras.

- ✓ Posee documentación en su página oficial la que incluye descripción y ejemplos de cada una de sus funciones.
- ✓ Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- ✓ Incluye gran cantidad de funciones.
- ✓ No requiere definición de tipos de variables ni manejo detallado del bajo nivel. (19)

ASP

El objetivo estándar de ASP es permitir el uso de casi cualquier lenguaje de programación. Sin embargo, dado que se trata de una tecnología creada por Microsoft, el lenguaje que utiliza básicamente es el VBScript y, en segundo lugar, el *JScript*. El VBScript es una versión especial del Visual Basic, con el que comparte la mayoría de sus funciones y mandatos. El *JScript* es la versión de JavaScript realizada por Microsoft.

El código en ASP no necesita ser compilado, ya que es interpretado por el servidor, a diferencia de otros lenguajes de programación como, por ejemplo, el Java. La desventaja es que el código interpretado es más lento que el compilado. Sin embargo, consume menos tiempo para el programador, ya que no es necesario compilar el código cada vez que se quiere probar el programa. (20)

JSP

Es un lenguaje para la creación de sitios web dinámicos, acrónimo de *Java Server Pages*. Está orientado a desarrollar páginas web en Java. JSP es un lenguaje multiplataforma, creado para ejecutarse del lado del servidor. Fue desarrollado por *Sun Microsystems* y realizado para la creación de aplicaciones web potentes. Para su funcionamiento se necesita tener instalado un servidor *Tomcat*.

Características:

- ✓ Código separado de la lógica del programa.
- ✓ Las páginas son compiladas en la primera petición.
- ✓ Permite separar la parte dinámica de la estática en las páginas web.

- ✓ Los archivos se encuentran con la extensión (jsp).
- ✓ El código JSP puede ser incrustado en código HTML. (21)

AJAX

AJAX (Asynchronous JavaScript And XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma. AJAX es un patrón de diseño que propone un nuevo modelo de interacción Web combinando las tecnologías anteriores. Los navegadores que soportan las tecnologías mencionadas son las plataformas en las que se ejecutan las aplicaciones AJAX (Firefox, IExplorer, Opera, Konqueror y Safari). (22)

HTML

Desde el surgimiento de Internet se han publicado sitios web gracias al lenguaje HTML. Es un lenguaje estático para el desarrollo de sitios web (acrónimo en inglés *de HyperText Markup Language*, en español Lenguaje de Marcas Hipertextuales). Los archivos pueden tener las extensiones (htm, html).

Dentro de sus principales ventajas presenta:

- ✓ Texto presentado de forma estructurada y agradable.
- ✓ No necesita de grandes conocimientos cuando se cuenta con un editor de páginas web.
- ✓ Despliegue rápido.
- ✓ Lenguaje de fácil aprendizaje.
- ✓ Lo admiten todos los exploradores. (23)

JavaScript

Es un lenguaje interpretado, no requiere compilación. Fue creado por Brendan Eich en la empresa Netscape Communications. Utilizado principalmente en páginas web. Es similar a Java, aunque no es un

lenguaje orientado a objetos, el mismo no dispone de herencias. La mayoría de los navegadores en sus últimas versiones interpretan código JavaScript.

El código JavaScript puede ser integrado dentro de nuestras páginas web. Para evitar incompatibilidades el *World Wide Web Consortium (W3C)* diseñó un estándar denominado DOM (en inglés *Document Object Model*, en su traducción al español Modelo de Objetos del Documento). (23)

1.6.5 Sistemas Gestores de base de datos (SGBD)

Los Sistemas Gestores de Bases de Datos son un tipo de software muy específico, dedicado a servir de interfaz entre las bases de datos y las aplicaciones que la utilizan. Están diseñados para gestionar grandes volúmenes de información y tienen como objetivo simplificar y facilitar el acceso a los datos y hacer que los tiempos de respuesta a las solicitudes de los usuarios sean muy reducidos.

MySQL

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. Fue creado por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. (24)

PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostgreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos. Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos

gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. Incorpora una estructura de datos *array* (*arreglo*). Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc. (24)

1.6.6 Entornos integrados de desarrollo

Un entorno de desarrollo integrado (IDE de su significado en *inglés Integrated Development Environment*), es un conjunto de herramientas que ha sido creado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Un IDE provee un marco de trabajo amigable para la mayoría de los lenguajes de programación y en algunos casos puede funcionar como un sistema en tiempo de ejecución en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto. (25)

Eclipse

Eclipse es un entorno de desarrollo integrado que facilita enormemente las tareas de edición, compilación y ejecución de programas durante su fase de desarrollo. Aunque Eclipse pretende ser un entorno versátil soportando varios lenguajes de programación, es con el lenguaje Java con el que mejor se integra y con el que ha ganado su popularidad.

Es una aplicación gratuita y de código abierto, disponible en la red para su descarga e incluida ya en muchas distribuciones de Linux. Eclipse proporciona el entorno de desarrollo solamente, siendo necesario además, para el caso del lenguaje Java, disponer del *Java Development Kit* (JDK) para poder compilar y ejecutar las aplicaciones desarrolladas. (26)

NetBeans

El NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. *Sun Microsystems* fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

Es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java pero puede servir para cualquier otro lenguaje de programación. Este es un producto

libre y gratuito sin restricciones de uso, que además está escrito completamente en Java usando la plataforma NetBeans. Soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Dentro de sus principales características presenta mejoras en el editor de código, instalación y actualización simple, posee características visuales para el desarrollo web y tiene soporte para PHP. 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. (27)

1.6.7 Fundamentación de las tecnologías, lenguajes y herramientas a utilizar

El lenguaje a utilizar en el desarrollo de la propuesta de solución es PHP por sus sobradas ventajas sobre Perl o JSP en cuanto a facilidad a la hora de utilizarlo y por ser mucho más liviano. Es uno de los lenguajes para la Web más popular, tiene una comunidad muy grande de desarrolladores, es gratuito y de código abierto, por lo que si se desea agregar o modificar algo para obtener un funcionamiento de acuerdo a las necesidades de cualquier usuario puede hacerse con total libertad.

Como SGBD de software libre se decidió utilizar PostgreSQL, ya que es el más recomendado para sistemas que tengan alta concurrencia y que manejen gran cantidad de información como es el caso de la aplicación en cuestión. Otra razón por la que se seleccionó este SGBD es que facilita el trabajo con procedimientos almacenados y consultas, además que brinda una estabilidad muy alta y una gran seguridad de los datos.

Como servidor Web se usará Apache debido a que es el servidor más utilizado a nivel mundial y provee un alto grado de calidad y fortaleza para las implementaciones que utilizan el protocolo HTTP. Además, soporta el lenguaje PHP que es el que será usado en la implementación. Se usará NetBeans como IDE ya que es un producto de código abierto que ayuda a los equipos de desarrollo a utilizar las mejores prácticas y estándares de la industria del software para la productividad general del grupo.

Se trabajará con Symfony, ya que este framework 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).

Para programar la capa de presentación se usará el ExtJS pues permite crear interfaces de usuarios muy amigables y funcionales de una manera intuitiva.

Se utilizará como capa de abstracción de base de datos el marco de trabajo Doctrine pues presenta un gran rendimiento en ejecución y permite escribir de forma concisa consultas muy complejas.

1.7 Conclusiones

En este capítulo se analizaron los principales conceptos y partiendo de ahí se estudiaron algunos sistemas relacionados con el tema, para verificar si tenían características similares al que se propone o derivar nuevas ideas; llegando a la conclusión que ninguno presenta las características ideales para ser usado en la Central Termoeléctrica Renté. Se describieron los principales aspectos a tener en cuenta para lograr un mejor entendimiento de la metodología, tecnologías, lenguajes y herramientas a utilizar para el desarrollo de dicho sistema. 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 Paradigm 6.4 (modelado), PostgreSQL 8.4 (SGBD), NetBeans 6.5 y Apache 2.2.6.

Capítulo 2: Características del sistema

2.1 Introducción

En el presente capítulo se realiza la descripción de la propuesta de solución, para ello se describen los procesos involucrados en el negocio. Se detallan las reglas del negocio a tener en cuenta para el desarrollo del sistema, así como las características que debe tener el mismo, expresado en los requisitos funcionales y no funcionales, a partir de los cuales se identifican los casos de uso del sistema y sus descripciones.

2.2 Modelo del negocio

El primer paso realizado durante el modelado del negocio fue la captura de los procesos de negocio de la organización, lo que define los límites del proceso de modelado posterior. Los procesos de negocio identificados se caracterizan por una colección de datos que son producidos y manipulados mediante un conjunto de tareas, en las que ciertos agentes participan de acuerdo a un flujo de trabajo determinado. Además, estos procesos se encuentran sujetos a un conjunto de reglas de negocio, que determinan las políticas y la estructura de la información. Por tanto, la finalidad del modelado del negocio es describir cada proceso del negocio, especificando sus datos, actividades (o tareas), roles (o agentes) y reglas.

2.2.1 Análisis de los 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.”

Básicamente en el área de almacenaje y las secciones asociadas a esta, se identificaron los siguientes procesos:

Recepción de los materiales suministrados por algún proveedor: proceso que se realiza por parte del almacenero en el almacén central, que incluye las actividades de recepción y conteo de la mercancía y a partir de esto se realiza el Informe de Recepción de Materiales, documento que avala la operación.

Reclamación al proveedor: proceso que tiene lugar una vez que se haya hecho efectiva la recepción y se haya producido algún tipo de anomalía en la calidad o cantidad de la mercancía recepcionada, que incluye la confección de un documento de reclamación, por parte del personal de economía.

Solicitud de materiales en los almacenes: proceso que tiene lugar cuando se produce alguna solicitud de algún material por parte de un responsable de área dentro de la empresa, mediante un Documento de Solicitud de Materiales entregado al almacenero, y se procede a satisfacer el pedido, confeccionándose el Vale de Entrega.

Solicitud de compra de materiales: proceso que tiene lugar cuando el almacenero después de haber recibido una solicitud de materiales, por parte del Responsable de área procede a buscar el material en los diferentes almacenes para responder la solicitud y se percata que no existe físicamente dicho material. El almacenero elabora un documento de Solicitud de Compra de Materiales (SCM) donde registra la no existencia de los mismos y se le envía al personal de economía, el que analiza la solicitud y elabora una factura de compra, archivando los dos documentos para su posterior solución.

Movimiento de materiales: proceso tiene lugar una vez que se vaya a realizar la recepción de los materiales suministrados por algún proveedor y no exista capacidad en ninguna taquilla dentro de los diferentes almacenes que tiene la empresa, comenzando el análisis por el almacén central. Se procede a mover los materiales agrupándolos por categorías y así liberar espacio para la ubicación de los nuevos materiales suministrados.

Devolución de materiales: proceso que lleva a cabo el responsable de área cuando por alguna anomalía o dificultad en los materiales sustraídos de algún almacén, realiza la devolución del mismo confeccionando el Informe de Materiales Defectuosos, documento que avala dicha operación.

Solicitud de información de materiales: proceso que se realiza cuando algún directivo de la empresa solicita al almacenero información de los materiales contenidos en los diferentes almacenes. El almacenero procede a confeccionar un reporte con la información solicitada.

2.2.2 Reglas del negocio a considerar

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 solo por personal autorizado.
- ✓ De existir más de un almacén o área de responsabilidad de almacenamiento, resulta necesario que la enumeración consecutiva de los documentos sea individual por área, identificando con siglas o códigos que antecedan a la numeración, el área o almacén a que corresponden.
- ✓ Que los materiales solo sean despachados mediante los documentos de salida oficialmente establecidos por el Sistema Nacional de Contabilidad (SNC).
- ✓ No deben existir documentos originados y destinados al área de almacén, con errores, tachaduras y enmiendas.
- ✓ El almacén informa las existencias de cada material en todos los modelos de entradas y salidas, después de anotados estos movimientos.
- ✓ Realizar los despachos, teniendo en cuenta la documentación económica establecida al respecto, Vale de Entrega o Devolución.
- ✓ El IRM debe reflejar los materiales recepcionados.
- ✓ Se identifica cada material almacenado con una Tarjeta de Estiba.
- ✓ Solo podrán efectuarse reclamaciones de materiales que ya fueron entrados al almacén.
- ✓ Las anotaciones en la Tarjeta de Estiba deben realizarse inmediatamente después de haber efectuado el movimiento físico de los materiales.
- ✓ Revisar la factura en el área económica.
- ✓ Recibir los materiales en el área destinada en el almacén para esta labor, (Área de Recepción del Almacén Central).
- ✓ Confeccionar el Informe de Reclamación si no se recibe el materiales tal y como describe la factura.

2.2.3 Actores del Negocio

Un actor del negocio es cualquier individuo, grupo, entidad, organización, máquina o sistema de información externos; con los que el negocio interactúa. Lo que se modela como actor es el rol que se juega cuando se interactúa con el negocio para beneficiarse de sus resultados.

Actor	Descripción
Proveedor	Entidad externa que le provee al almacén central materiales previamente solicitados por el personal de economía.
Responsable de área	Persona facultada para solicitar materiales al almacén central.
Directivo	Persona facultada para solicitar información de los materiales existentes en los almacenes.

Tabla 1: Actores del negocio

2.2.4 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. [43]

Trabajador	Descripción
Almacenero	Responsable de darle entrada y salida a los materiales del almacén.
Personal de economía	Empleado del departamento de economía que atiende el almacén, llevando a cabo la contrapartida contable de cada una de las operaciones realizadas en el almacén central. Emite los comprobantes contables. Encargado de realizar las reclamaciones al proveedor.

Tabla 2: Trabajadores del negocio

2.2.5 Diagrama de casos de uso del negocio

Para comprender los procesos de negocio se construye el diagrama de casos de uso del negocio en el que aparece cada proceso del negocio relacionado con su actor.

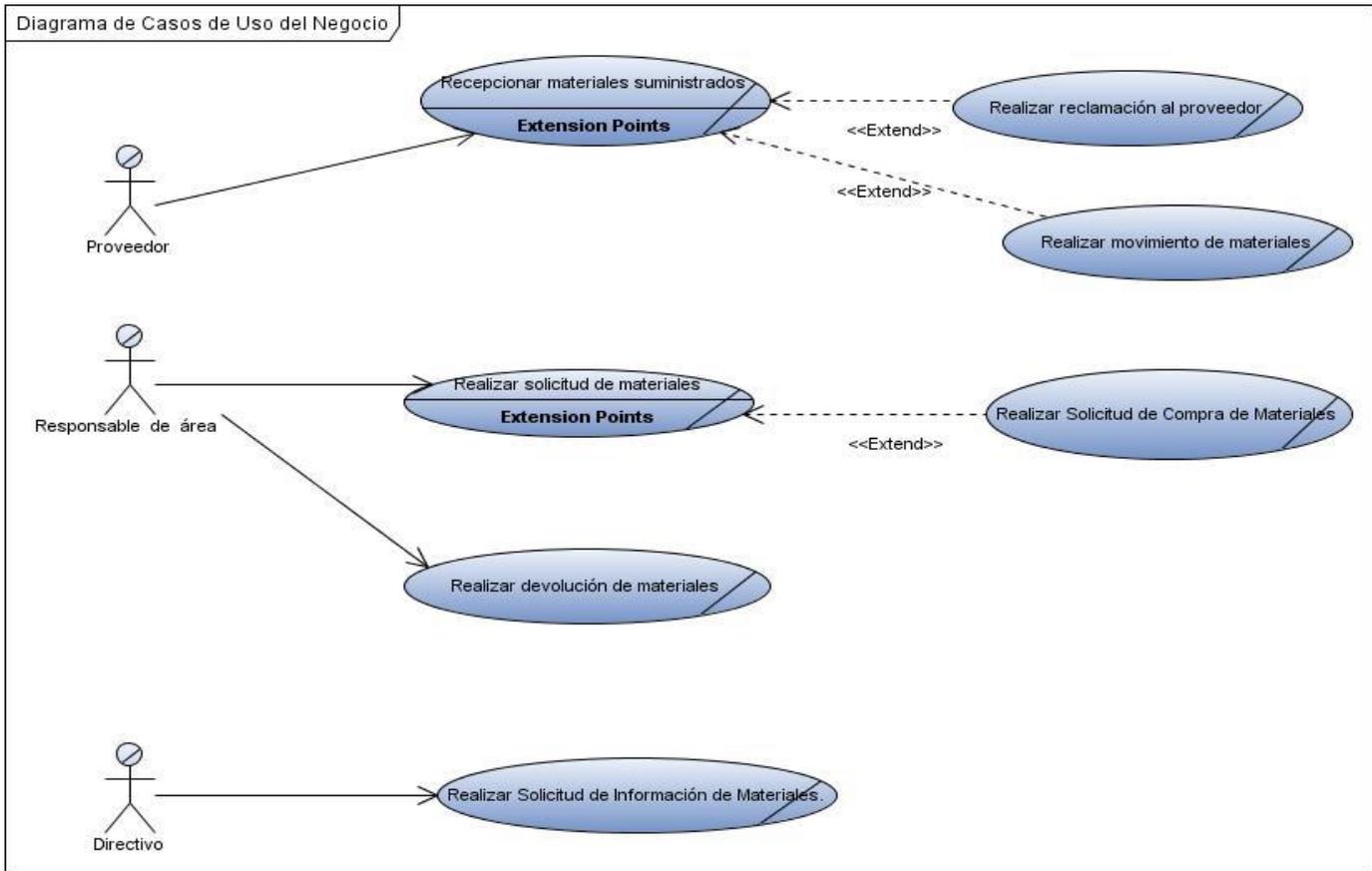


Figura 2: Diagrama de Casos de Uso del Negocio.

2.2.6 Realización de Casos de Uso del Negocio

Descripción textual del Caso de Uso: Recepcionar materiales suministrados.

Caso de Uso:	Recepcionar materiales suministrados.
Actores:	Proveedor (inicia).
Trabajadores:	Almacenero, personal de economía.

Resumen:	El caso de uso se inicia cuando el proveedor entrega los materiales correspondientes al almacén, el almacenero verifica la cantidad y que estén en buen estado, realiza un informe de recepción de materiales (IRM) y actualiza la tarjeta de estiba de los materiales en el almacén, el proveedor entrega la Factura de los materiales en el dpto. de economía, de existir algún problema se efectúa una reclamación.	
Precondiciones:	1. El proveedor debe entregar la Factura al departamento de economía.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. El proveedor entrega los materiales en el almacén central y la factura en el departamento de economía.	2. El almacenero recibe y cuenta los materiales y elabora el informe de recepción de materiales (IRM). 3. El almacenero envía el IRM al departamento de economía. 4. El personal de economía comprueba que la cantidad de materiales en el IRM coincide con la factura del proveedor. Si lo recibido no coincide con la Factura, se efectúa una reclamación (ver CU: Realizar reclamación al proveedor). 5. El almacenero procede a ubicar los materiales en el espacio físico disponible en algunos de los almacenes. 6. El almacenero actualiza la Tarjeta de Estiba.	
Poscondiciones	Se actualiza la cantidad de materiales.	

Tabla 3: CUN Registrar entrada de materiales.

Descripción textual del Caso de Uso: Realizar reclamación al proveedor.

Caso de Uso:	Realizar reclamación al proveedor.
---------------------	------------------------------------

Capítulo 2: Características del sistema

Actores:	Proveedor
Trabajadores:	Almacenero, personal de economía.
Resumen:	El caso de uso se inicia cuando los materiales suministrados por el proveedor no se corresponden con las cantidades en la Factura o su calidad no es la requerida.
Precondiciones:	<ol style="list-style-type: none"> 1. Detectar anomalías en los materiales suministrados por el proveedor. 2. Estar dentro del plazo establecido para efectuar la reclamación.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
	<ol style="list-style-type: none"> 1. El personal de economía compara la Factura (F) entregada por el proveedor con el IRM. <ol style="list-style-type: none"> a) Si no existe problemas con los documentos. Ver flujo alternativo 1. b) Si existe problemas con los materiales el personal de economía elabora un Documento de Reclamación (DR) y lo envía al proveedor.
2. El proveedor recibe el DR.	<ol style="list-style-type: none"> 3. El personal de economía anexa al IRM, la Factura y el DR, en espera de una respuesta del proveedor. 4. El personal de economía envía una copia del DR al almacenero. 5. El almacenero archiva la copia del DR.
Flujo Alternativo: 1	
	<ol style="list-style-type: none"> 1. El personal de economía archiva Factura y el IRM.

Poscondiciones	Se actualiza la Tarjeta de Estiba (TE) de los materiales involucrados en el proceso.
-----------------------	--------------------------------------------------------------------------------------

Tabla 4: CUN Realizar reclamación al proveedor

Descripción textual del Caso de Uso: Realizar solicitud de materiales.

Caso de Uso:	Realizar solicitud de materiales.
Actores:	Responsable de área
Trabajadores:	Almacenero, personal de economía.
Resumen:	El responsable del área envía al almacenero un Documento de Solicitud de Materiales (DSM), de los materiales que necesita; si no hay en el almacén o no hay suficiente para resolver la solicitud del pedido el almacenero da a conocer al personal de economía la insuficiencia de este material. Se realiza el despacho de los materiales.
Precondiciones:	1. Para atender el pedido deben existir materiales en el almacén.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. El responsable de área solicita materiales al almacén central mediante un DSM.	2. El almacenero recibe el DSM. 3. El almacenero verifica que puede satisfacer la solicitud. a) Si no puede satisfacer la solicitud Ver Flujo Alterno 1. 4. Localiza los materiales en el almacén. 5. Confecciona el Vale de Entrega de materiales en correspondencia con el DSM. 6. Despacha los materiales al responsable de área, entregándole una copia del VE. 7. El almacenero actualiza la TE. 8. EL almacenero archiva el DSM y VE. 9. El almacenero envía copia al personal de

Capítulo 2: Características del sistema

	<p>economía.</p> <p>10. El personal de economía recibe copias del VE.</p>
<p>11. El responsable de área recibe los materiales y el VE.</p> <p>a) Si el responsable de área detecta anomalía en los materiales. Ver Flujo Alterno 2.</p>	<p>12. El personal de economía emite un comprobante contable.</p>
Flujo Alterno: 1	
	<p>1. El almacenero elabora un documento donde especifica el faltante o no existencia de los materiales en los almacenes.</p> <p>2. El almacenero envía el documento al Personal de Economía. (Ver CU Realizar solicitud de compra de materiales.)</p>
Flujo Alterno: 2	
<p>1. Se informa al almacenero la relación de materiales defectuosos o el faltante.</p>	<p>2. El almacenero recibe el IMD donde se especifica la cantidad de materiales defectuosos en el almacén.</p>
Poscondiciones	Se actualiza la TE de los materiales involucrados.

Tabla 5: CUN Realizar solicitud de materiales.

Descripción textual del Caso de Uso: Realizar devolución de materiales.

Caso de Uso:	Realizar devolución de materiales.
Actores:	Responsable de área.
Trabajadores:	Almacenero, personal de economía

Capítulo 2: Características del sistema

Resumen:	El caso de uso se inicia cuando algún responsable de área desea devolver al almacén u otra área por algún motivo, materiales que previamente sustrajo. El almacenero o el responsable de área, recibe los materiales, confecciona los documentos que avalen la operación y actualiza la cantidad de los materiales.	
Precondiciones:	1. Los productos a devolver deben haber sido movidos previamente a dicha área.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. El responsable de área entrega los materiales a devolver mediante el Informe de Materiales Defectuosos (IMD).	2. El almacenero recibe los materiales y el IMD 3. El almacenero verifica que todo esté en orden. 4. El almacenero actualiza la TE de los materiales involucrados. 5. El almacenero envía copias de IMD al dpto. de economía. 6. El personal de economía archiva el IMD 7. El personal de economía emite un comprobante contable.	
Poscondiciones	Actualiza la TE de los materiales.	

Tabla 6: CUN Realizar devolución de materiales.

Descripción textual del Caso de Uso: Realizar movimiento de materiales.

Caso de Uso:	Realizar movimiento de materiales.
Actores:	Proveedor
Trabajadores:	Almacenero
Resumen:	El caso de uso se inicia cuando el proveedor hace entrega de los nuevos materiales al almacenero el que por falta de espacio en los almacenes, accede a mover los materiales ya sea dentro de un mismo almacén o entre los almacenes, tratando de agruparlo por categoría.

Precondiciones:	<ol style="list-style-type: none"> 1. El material debe existir en alguno de los almacenes. 2. Debe existir espacio para agrupar los materiales.
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. El proveedor hace entrega de los nuevos materiales al almacenero.	<ol style="list-style-type: none"> 2. El almacenero analiza la capacidad libre existente en los almacenes. 3. El almacenero realiza los movimientos correspondientes, agrupando los materiales por categoría. 4. El almacenero da entrada a los nuevos materiales. Ver CU Recepcionar materiales suministrados.
Flujo Alternativo 1	
	1. El almacenero envía una notificación al directivo de la empresa.
Poscondiciones	Se actualizan las existencias de los materiales transferidos en las secciones involucradas.

Tabla 7: CUN Realizar movimiento de materiales.

Descripción textual del Caso de Uso: Realizar Solicitud de Compra de Materiales.

Caso de Uso:	Realizar Solicitud de Compra de Materiales.
Actores:	Responsable de área
Trabajadores:	Almacenero, personal de economía.
Resumen:	El caso de uso se inicia cuando el responsable de área realiza una solicitud de materiales al almacenero. El mismo se percata de la falta de materiales para poder atender la solicitud y elabora un documento donde registra el faltante y lo envía al personal de economía.
Precondiciones:	<ol style="list-style-type: none"> 1. No se pueda atender la solicitud del Responsable de área.

Flujo Normal de Eventos	
Acción del Actor	Respuesta del Negocio
1. El responsable de área realiza una solicitud de materiales al almacenero.	2. El almacenero analiza la solicitud, a) Si existe parte o totalidad del material para responder la solicitud ver Flujo Alterno 1. 3. El almacenero elabora un documento donde registra la falta de materiales en los almacenes. 4. El almacenero envía el documento al personal economía. 5. El personal de economía analiza el documento. 6. El personal de economía elabora la Solicitud de Compra de Materiales (SCM) y es enviada al proveedor. 7. El almacenero envía una notificación al responsable de área donde especifica que no puede atender la solicitud.
8. El responsable de área recibe la notificación.	
Flujo Alterno 1	
	Ver CU Realizar Solicitud de materiales
Poscondiciones	El personal de economía debe recibir el documento donde se especifica el faltante de materiales en los almacenes.

Tabla 8: CUN Realizar Solicitud de Compra de Materiales .

Descripción textual del Caso de Uso: Realizar Solicitud de Información de Materiales.

Caso de Uso:	Realizar Solicitud de Información de Materiales.
Actores:	Directivo
Trabajadores:	Almacenero
Resumen:	El caso de uso se inicia cuando el directivo realiza la solicitud de información de

	los materiales existente en la empresa al almacenero. El almacenero mediante los reportes pertinentes satisface las necesidades del directivo.	
Precondiciones:	1. El directivo de la empresa solicite información a la empresa.	
Flujo Normal de Eventos		
Acción del Actor	Respuesta del Negocio	
1. El directivo de la empresa solicita información de los materiales al almacenero.	2. El almacenero analiza los criterios de información solicitados. 3. El almacenero a través de un reporte obtiene la información solicitada. 4. El almacenero le envía el reporte al directivo de la empresa.	
5. El directivo de la empresa analiza el reporte. a) Si el informe no cumple su solicitud ver Flujo Alterno 1. 6. El directivo envía una notificación al almacenero.	7. El almacenero recibe la notificación.	
Flujo Alterno 1		
1. El directivo de la empresa le envía una notificación al almacenero explicándole el desacuerdo en la información solicitada y realiza una nueva solicitud.	2. El almacenero recibe la notificación. 3. Continuar con la acción 2 del Flujo Normal de Eventos.	
Poscondiciones	Se generan los reportes correspondientes a la solicitud de los directivos.	

Tabla 9: CUN Realizar Solicitud de Información de Materiales.

2.2.7 Diagrama de actividades

El diagrama de actividad es un grafo que contiene los estados en que puede encontrarse la actividad a analizar. Cada estado de la actividad representa la ejecución de una sentencia de un procedimiento, o el funcionamiento de una actividad en un flujo de trabajo. En resumen describe un proceso que explora el orden de las actividades que logran los objetivos del negocio.

En el **Anexo II** se muestran los diagramas de actividades de los casos de uso del modelo del negocio.

2.2.8 Modelo de objeto

Un modelo de objetos del negocio es un modelo interno a un negocio. Describe como cada caso de uso del negocio es llevado a cabo por parte de un conjunto de trabajadores que utilizan un conjunto de entidades del negocio y unidades de trabajo.

Una entidad del negocio representa algo, que los trabajadores toman, examinan, manipulan, crean o utilizan en un caso de uso del negocio. El diagrama de clases del modelo de objeto, es un artefacto que se construye para describir el modelo de objetos del negocio. A continuación se muestra el modelo de objetos del negocio.

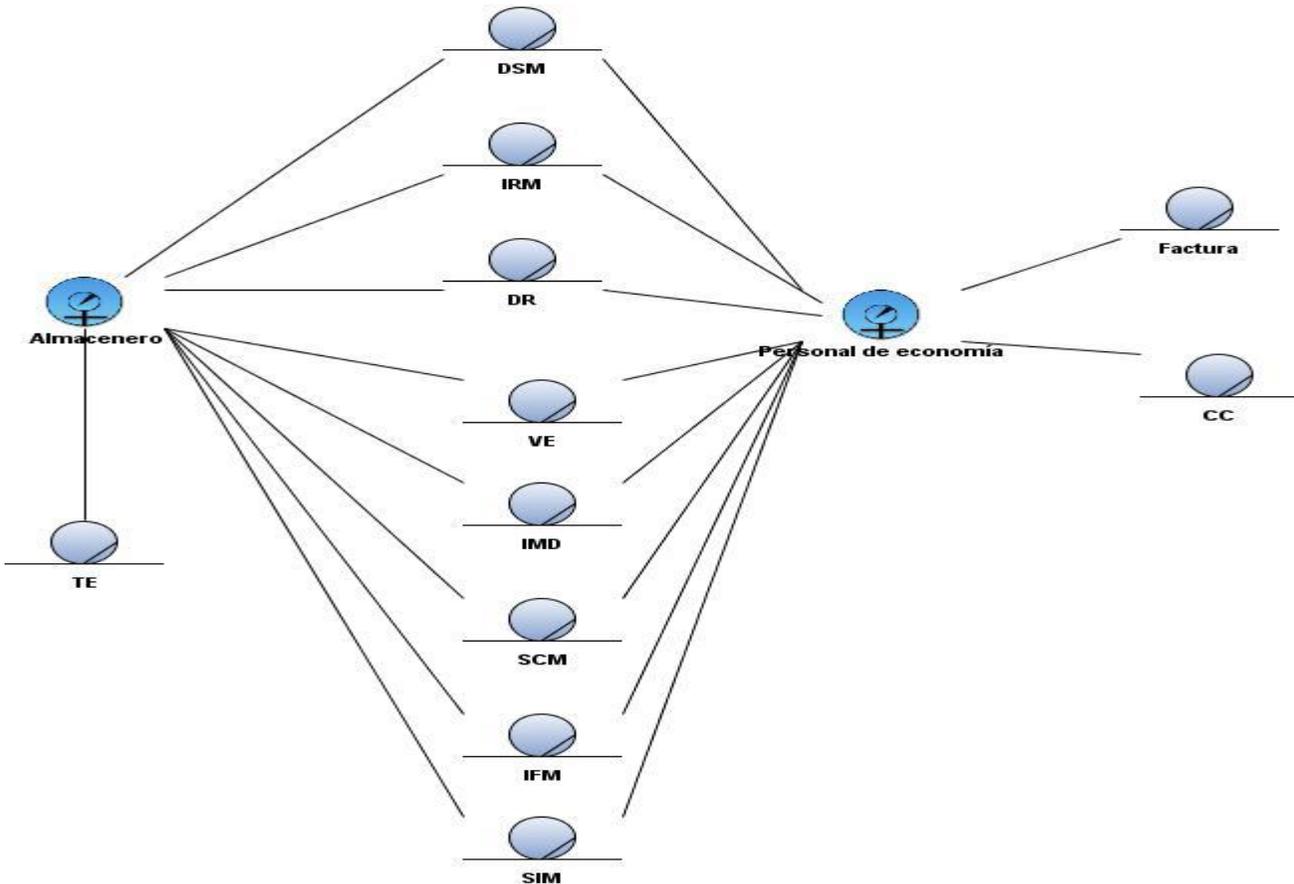


Figura 3: Modelo de objeto del negocio.

2.3 Especificación de los requisitos el sistema.

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.

2.3.1 Requisitos funcionales.

Los requerimientos funcionales permiten expresar una especificación más detallada de las responsabilidades del sistema que se propone. Ellos permiten determinar, de una manera clara, lo que debe hacer el mismo. Los requerimientos funcionales son los siguientes:

- RF1** Registrar entrada de materiales.
- RF2** Rebajar materiales.
- RF3** Buscar materiales.
- RF4** Crear tarjeta de estiba.
- RF5** Modificar tarjeta de estiba.
- RF6** Eliminar tarjeta de estiba.
- RF7** Adicionar solicitud de materiales.
- RF8** Modificar solicitud de materiales.
- RF9** Eliminar solicitud de materiales.
- RF10** Crear Vale de Entrega de Materiales.
- RF11** Modificar Vale de Entrega de Materiales
- RF12** Eliminar Vale de Entrega de Materiales
- RF13** Realizar devolución de materiales.

- RF14** Realizar reclamación al proveedor.
- RF15** Crear Informe de Faltante de materiales.
- RF16** Modificar Informe de Faltante de materiales.
- RF17** Eliminar Informe de Faltante de Materiales.
- RF18** Crear Solicitud de Compra de Materiales.
- RF19** Modificar Solicitud de Compra de Materiales.
- RF20** Eliminar Solicitud de Compra de Materiales.
- RF21** Generar reportes.
- RF22** Autenticar usuario.
- RF23** Adicionar usuario.
- RF24** Modificar usuario.
- RF25** Eliminar usuario.
- RF26** Adicionar rol de usuario.
- RF27** Modificar rol de usuario.
- RF28** Eliminar rol de usuario

2.3.2 Requisitos no funcionales.

Los requerimientos no funcionales especifican cualidades, propiedades del sistema; como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, etc.

Requerimientos de Apariencia o Interfaz Externa

Se podrán distinguir colores atractivos.

Debe poseer un ambiente amigable, intuitivo, sencillo y de fácil navegación, tratando así de impedir el rechazo por parte del usuario al tener que interactuar con un sistema no conocido.

Paginación de reportes de búsqueda y listados.

Diseño perfectamente encuadrado para resoluciones de 1024 x 768, pero preparado para verse en otras resoluciones.

Requerimientos de Usabilidad

La aplicación Web debe ser flexible y de fácil aprendizaje, pues se trata en todo lo posible de mantener un estándar de operabilidad que logre que las interacciones del usuario con el sistema sean predecibles y familiares.

El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora y de un ambiente Web en sentido general.

Requerimientos de Soporte

El sistema contará con una ayuda para facilitar al usuario el manejo de la aplicación.

Estará bien documentado para garantizar futuros mantenimientos.

Requerimientos de Portabilidad

El sistema podrá ejecutarse sobre plataforma Linux, Windows 98 o superior.

Requerimientos de Seguridad

Las funcionalidades deberán estar restringidas de acuerdo a los diferentes roles, con el objetivo de garantizar que el acceso a la información sea según el nivel de autorización.

El sistema deberá estar protegido ante el acceso no autorizado a la información.

Requerimientos de Software

Del lado del cliente:

El cliente podrá tener acceso al sistema a través de los navegadores Web Mozilla.

Sistema operativo Linux o Windows 98 ó Superior.

Del lado del servidor:

Sistema operativo Linux.

Servidor Web Apache 2.2.6 y PHP 5.

Servidor de Base de Datos PostgreSQL 8.4

Framework Symfony.

Requerimientos de Hardware

Del lado del cliente:

Procesador Pentium III o superior.

256 de memoria RAM o superior.

Monitor VGA o superior.

Tarjeta de red.

Del lado del servidor:

Procesador Pentium IV o superior.

2GB de memoria RAM o superior.

Disco Duro de 80 GB.

Monitor tipo VGA o superior.

Tarjeta de red.

Requerimientos de Diseño e Implementación

Se utilizará como metodología de desarrollo RUP, como lenguaje de modelado y como herramienta case el Visual Paradigm. Para la implementación como lenguaje de programación PHP y como IDE de desarrollo NetBeans.

2.4 Modelo del sistema

2.4.1 Actores del Sistema

Un actor no es más que un conjunto de roles que los usuarios de Casos de Uso desempeñan cuando interactúan con estos Casos de Uso. Los actores representan a terceros fuera del sistema que colaboran con el mismo. Una vez que se haya identificado los actores del sistema, está identificado el entorno externo del sistema. (28)

Actor	Descripción
Responsable de área	Usuario del sistema con permisos para realizar actividades como movimientos desde/hacia secciones, solicitar productos y las reversiones de movimientos.
Almacenero	Responsable del área central por donde entran los productos suministrados por el proveedor, registra este tipo de entrada.
Personal de economía	Usuario del sistema con privilegios para registrar las operaciones del sistema, y hacer las reclamaciones al proveedor.
Informático	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 10: Actores del sistema.

2.4.2 Diagrama de casos de uso del sistema

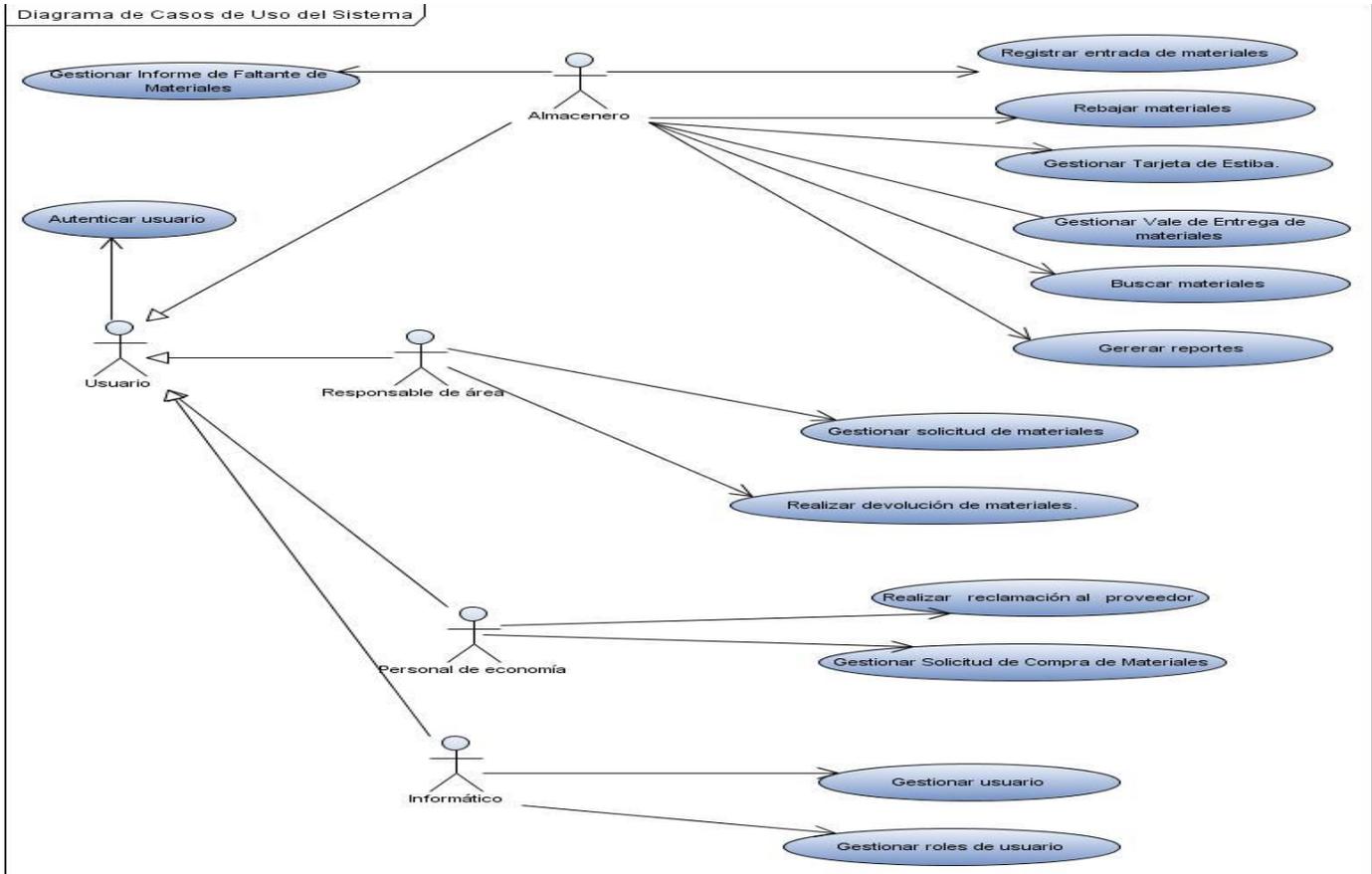


Figura 4: Diagrama de CUS del Sistema

2.4.3 Expansión de los Casos de Uso del Sistema

A continuación se muestran las descripciones abreviadas de cada caso de uso, las descripciones extendidas están disponibles **Anexo III**

Caso de Uso:	Registrar entrada de materiales.
Actores:	Almacenero
Resumen:	El caso de uso se inicia cuando el almacenero recibe los materiales y realiza la recepción

Capítulo 2: Características del sistema

	de los mismos a través del sistema. Además, este caso de uso permite darle entrada a los materiales que lleguen al almacén mediante la compra.
--	------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 11: CUS Registrar entrada de materiales.

Caso de Uso:	Rebajar materiales.
Actores:	Almacenero
Resumen:	El presente caso de uso tiene como objetivo posibilitar el flujo de materiales de una sección a otra. En el mismo se registran todos los datos necesarios para llevar a cabo la transferencia y el mismo comienza cuando el almacenero decide atender determinada solicitud de materiales hecha a su área de trabajo.

Tabla 12: CUS Rebajar materiales.

Caso de Uso:	Buscar materiales.
Actores:	Almacenero
Resumen:	El caso de uso inicia cuando el almacenero decide realizar una búsqueda de materiales.

Tabla 13: CUS Buscar materiales.

Caso de Uso:	Gestionar Tarjeta de Estiba.
Actores:	Almacenero
Resumen:	El caso de uso inicia cuando el almacenero realiza la recepción de materiales y decide atender una solicitud previamente realizada por el Responsable de área.

Tabla 14: CUS Gestionar Tarjeta de Estiba.

Caso de Uso:	Gestionar solicitud de materiales
Actores:	Responsable de área
Resumen:	Cuando un Responsable de área se da cuenta de que le falta un material simplemente realiza dicha solicitud a cualquiera de las secciones, solicitando la cantidad que necesita,

Capítulo 2: Características del sistema

	<p>si dicha sección no puede satisfacer la solicitud el sistema muestra una lista de la secciones que pueden satisfacer para que haga una nueva solicitud. El caso de uso termina cuando se guarda el documento Solicitud de Materiales. Este caso de uso posibilita al actor realizar solicitudes de materiales con el fin de abastecer su sección según los niveles de inventario de la misma, estas se registran hasta ser satisfechas o rechazadas por el Responsable de área la que se le solicita.</p>
--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Tabla 15: CUS Gestionar solicitud de materiales.

Caso de Uso:	Gestionar Vale de Entrega de materiales.
Actores:	Almacenero.
Resumen:	El caso de uso inicia cuando el almacenero decide atender una solicitud de materiales previamente hecha por el Responsable de área, cuando el mismo procede a despachar dicho materiales se emite el Vale Entrega de materiales, documento que avala dicha operación.

Tabla 16: CUS Gestionar Vale de Entrega de materiales.

Caso de Uso:	Realizar devolución de materiales.
Actores:	Responsable de área
Resumen:	El responsable de área selecciona el movimiento previamente registrado y realiza la reversión. El caso de uso permite que se revierta un movimiento de materiales en caso de algún error u otra situación.

Tabla 17: CUS Realizar devolución de materiales.

Caso de Uso:	Realizar reclamación al proveedor
Actores:	Personal de Economía
Resumen:	El caso uso inicia cuando el Personal de Economía se dispone a realizar la reclamación a el proveedor. Llena la reclamación y la registra en el sistema.

Tabla 18: CUS Realizar reclamación al proveedor.

Capítulo 2: Características del sistema

Caso de Uso:	Gestionar Informe de Faltante de Materiales.
Actores:	Almacenero
Resumen:	El caso de uso inicia cuando el Responsable de área le envía una solicitud de materiales al almacenero, y este después de realizar una búsqueda de materiales en los almacenes percibe que no existen estos materiales.

Tabla 19: CUS Gestionar Informe de Faltante de Materiales.

Caso de Uso:	Gestionar Solicitud de Compra de Materiales.
Actores:	Personal de economía.
Resumen:	El caso de uso inicia cuando el almacenero le envía al Personal de economía el IFM y a partir de este se hace una solicitud de compra de materiales, elaborándose la Solicitud de Compra de Materiales, documento que avala dicha operación.

Tabla 20: CUS Gestionar Solicitud de Compra de Materiales.

Caso de Uso:	Generar reportes.
Actores:	Almacenero
Resumen:	El caso de uso se inicia cuando el Almacenero ejecuta la aplicación y selecciona realizar generar reporte, de acuerdo a los datos previamente seleccionados por el mismo, 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 sale de la aplicación.

Tabla 21: CUS Generar reportes.

Caso de Uso:	Autenticar usuario
Actores:	Informático, Personal de economía, Almacenero, Responsable de área
Resumen:	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.
--	----------------------------------------------------

Tabla 22: CUS Autenticar usuario

Caso de Uso:	Gestionar usuario
Actores:	Informático
Resumen:	El caso de uso se inicia cuando el Informático ejecuta la aplicación y selecciona gestionar usuario e inmediatamente se le muestra un listado con todos los usuarios existentes, 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.

Tabla 23: CUS Gestionar usuario.

Caso de Uso:	Gestionar rol
Actores:	Informático
Resumen:	El caso de uso se inicia cuando el Informático ejecuta la aplicación y selecciona gestionar rol e inmediatamente se le muestra un listado con todos los roles existentes, brindando las opciones de adicionar, modificar y eliminar. El caso de uso termina cuando el sistema realiza la operación elegida, y el usuario designado selecciona otra opción o sale de la aplicación.

Tabla 24: CUS Gestionar rol.

2.5 Conclusiones

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 3: Análisis y diseño del sistema.

2.1 Introducción

En este capítulo se describe el desarrollo del Modelo de Análisis, que aporta una visión del nuevo sistema propuesto sobre los requisitos funcionales identificados. Para representar esto se usaron los diferentes artefactos del UML, diagramas de clases y modelados con Visual Paradigm aplicando RUP.

2.2 Estimación del esfuerzo

Una de las tareas de mayor importancia en la planificación de proyectos de software es la estimación, la que consiste en determinar con cierto grado de certeza, los recursos de hardware, software, costo, tiempo y esfuerzo necesarios para el desarrollo de los mismos. En el presente capítulo se realiza un estudio de factibilidad para la realización del sistema propuesto, utilizando para ello el cálculo de puntos de casos de uso.

Cálculo de puntos de casos de usos sin ajustar.

UUCP = UAW + UUCW donde:

UUCP: Puntos de Casos de Uso sin ajustar.

UAW: Factor de Peso de los Actores sin ajustar.

UUCW: Factor de Peso de los Casos de Uso sin ajustar.

Factor de Peso de los Actores sin ajustar (UAW)

El cálculo de este valor es realizado mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema.

Teniendo en cuenta que existen 4 actores definidos en el sistema, los cuales interactúan con el mismo mediante una interfaz gráfica, el tipo de actor y el factor de peso serían complejos e igual a 3 respectivamente.

UAW = 4×3

UAW = 12

Factor de Peso de los Casos de Uso sin ajustar (UUCW)

Capítulo 3: Análisis y diseño del sistema

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica.

Tipo de Caso de Uso	Descripción	Factor de Peso	Cantidad de Casos de Uso
Simple	El Caso de Uso contiene de 1 a 3 Transacciones.	5	
Medio	El Caso de Uso contiene de 4 a 7 Transacciones.	10	14
Complejo	El Caso de Uso contiene más de 8 Transacciones.	15	

Tabla 25: Factor de Peso de los Casos de Uso sin ajustar

Clasificación de la complejidad de los Casos de Uso de acuerdo al número de transacciones.

Caso de Uso	Transacciones	Factor de Peso
Registrar entrada de materiales.	3	5
Rebajar materiales.	3	5
Buscar materiales.	3	5
Gestionar tarjeta de estiba	3	5
Gestionar solicitud de materiales.	8	15
Gestionar Vale de Entrega de Materiales	8	15
Realizar devolución de materiales	8	15
Realizar reclamación al proveedor	3	5
Gestionar Informe de Faltante de Materiales	8	15

Gestionar Solicitud de Compra de Materiales	8	15
Generar reportes	3	5
Autenticar usuario.	3	5
Gestionar usuario	8	15
Gestionar rol	8	15

Tabla 26: Complejidad de los CU según sus Transacciones

Las tablas anteriores muestran que se tienen 14 Casos de Uso de ellos cuatro son simples, cuatro son casos de uso medio y dos son complejos.

UUCW = \sum cant. CU * Factor de Peso

UUCW = $(6*5) + (8*15)$

UUCW = 150

Una vez conocidos los valores de los factores de peso de los actores y de los casos de uso sin ajustar (UAW y UUCW) se puede calcular los Puntos de Casos de Uso sin ajustar (UUCP). Entonces finalmente tendríamos que:

UUCP = UAW + UUCW

UUCP = 12 + 150

UUCP = 162

Cálculo de Puntos de Casos de Uso ajustados.

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

UCP = UUCP x TCF x EF donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

TCF: Factor de complejidad técnica.

EF: Factor de ambiente.

Factor de Complejidad Técnica (TCF)

Capítulo 3: Análisis y diseño del sistema

El factor de complejidad técnica se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. El significado y el peso de cada uno de estos factores se muestran en la siguiente tabla:

Factor	Descripción	Peso	Valor	(Peso x Valor)
T1	Sistema distribuido	2	0	0
T2	Objetivos de performance (representación) o tiempo de respuesta	1	1	1
T3	Eficiencia del usuario final	1	3	3
T4	Procesamiento interno complejo	1	3	3
T5	El código debe ser reutilizable	1	4	4
T6	Facilidad de instalación	0.5	3	1.5
T7	Facilidad de uso	0.5	4	2
T8	Portabilidad	2	0	0
T9	Facilidad de cambio	1	4	4
T10	Concurrencia	1	0	0
T11	Incluye objetivos especiales de seguridad	1	5	5
T12	Provee acceso directo a terceras partes	1	5	5
T13	Se requieren facilidades especiales de entrenamiento a usuarios.	1	1	1
Total				29.5

Tabla 27: Factor de Complejidad Técnica.

Clasificación de Complejidad Técnica del Sistema

El factor de complejidad técnica se calcula mediante la ecuación siguiente:

$$TCF = 0.6 + (0.01 \times \Sigma (\text{Peso}_i \times \text{Valor asignado}_i))$$

$$TCF = 0.6 + (0.01 \times (0+1+3+3+4+1+2+0+4+0+5+5+1))$$

$$TCF = 0.6 + (0.01 \times 29.5)$$

$$TCF = 0.895$$

Cálculo del Factor de ambiente (EF)

Para el cálculo del factor de ambiente se tienen en cuenta las habilidades y el entrenamiento del grupo involucrado, los cuales son de gran impacto en las estimaciones de tiempo. Los factores se cuantifican con valores de 0 a 5, similar al cálculo del factor de complejidad técnica. En la siguiente tabla se muestra el significado y el peso de cada uno de los factores:

Factor	Descripción	Peso	Valor	Σ (Pesoi * Valori)
E1	Familiaridad con el modelo de proyecto utilizado	1.5	4	6
E2	Experiencia en la aplicación	0.5	4	2
E3	Experiencia en orientación a objetos	1	4	4
E4	Capacidad del analista líder	0.5	5	2.5
E5	Motivación	1	5	5
E6	Estabilidad de los requerimientos	2	2	4
E7	Personal part-time	-1	0	0
E8	Dificultad del lenguaje de programación	-1	2	-2
Total				21.5

Tabla 28: Factor de Ambiente

Clasificación del Factor de Ambiente del Sistema.

Para los factores E1 al E4, un valor asignado de 0 significa sin experiencia, 3 experiencia media y 5 amplia experiencia (experto).

Para el factor E5, 0 significa sin motivación para el proyecto, 3 motivación media y 5 alta motivación.

Para el factor E6, 0 significa requerimientos extremadamente inestables, 3 estabilidad media y 5 requerimientos estables sin posibilidad de cambios.

Para el factor E7, 0 significa que no hay personal part-time (es decir todos son full-time), 3 significa mitad y mitad, y 5 significa que todo el personal es part-time (nadie es full-time).

Para el factor E8, 0 significa que el lenguaje de programación es fácil de usar, 3 medio y 5 que el lenguaje es extremadamente difícil.

El Factor de ambiente se calcula mediante la siguiente ecuación:

$$EF = 1.4 - (0.03 \times \Sigma (\text{Pesoi} \times \text{Valor asignadoi}))$$

$$EF = 1.4 - (0.03 \times (6+2+4+2.5+5+4+0-2))$$

$$EF = 1.4 - (0.03 \times 21.5)$$

$$EF = 0.755$$

Una vez conocidos los valores (UUCP, TCF, EF), para calcular los Casos de Uso Ajustados (UCP), se pasa a resolver la siguiente ecuación:

$$UCP = UUCP \times TCF \times EF$$

$$UCP = 162 \times 0.895 \times 0.755$$

$$UCP = 109.46745$$

Cálculo de los Puntos de Casos de Uso a la estimación del esfuerzo (E).

Cuando se habla de esfuerzo en este contexto se refiere a la relación entre la cantidad de hombres y el tiempo. Para encontrar este valor depende del valor de los Puntos de Casos de Uso Ajustados y el Factor de Conversión, como lo muestra la siguiente ecuación:

$$E = UCP \times CF$$

Donde:

E: Esfuerzo estimado en horas-hombre.

UCP: Puntos de Casos de Uso ajustados.

CF: Factor de conversión.

Conversión de los Puntos de Casos de Uso Ajustados a Esfuerzo de Desarrollo.

Para la búsqueda del Factor de Conversión se siguen una serie de pasos:

Paso #1: Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6.

Para este estudio se tiene por debajo del valor medio (3) de los factores del E1 al E6, a 1 factor de los que afectan al Factor de ambiente (EF).

Paso #2: Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.

Para este estudio se tiene por encima del valor medio (3) de los factores E7 y E8, a ningún factor.

Paso # 3: Ahora se tiene que:

Si el total es 2 o menos, se utiliza el factor de conversión 20 horas/hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 20 horas/hombre.

Si el total es 3 o 4, se utiliza el factor de conversión 28 horas/hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 28 horas/hombre.

Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

Como se tiene que el total de los pasos 1 y 2 efectuados anteriormente es 1, entonces se el factor de conversión (CF) para el producto es de:

$$CF = 20 \text{ horas/hombre}$$

Entonces el esfuerzo estimado en horas/hombre sería:

$$E = UCP \times CF$$

$$E = 109.46745 \times 20$$

$$E = 2189.349 \sim 2189 \text{ Horas/Hombre}$$

El esfuerzo estimado de 2189 horas/hombre calculado anteriormente, representa un porcentaje del esfuerzo total del proyecto, entonces para una estimación más completa de la duración total del proyecto, hay que agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software. Para ello se puede tener en cuenta el siguiente criterio, que estadísticamente se considera aceptable. El criterio plantea la distribución del esfuerzo entre las diferentes actividades de un proyecto, según la siguiente aproximación:

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10%	218.9 horas-hombre
Diseño	20%	437.8 horas-hombre
Implementación	40%	875.6 horas-hombre
Prueba	15%	328.35 horas-hombre
Sobrecarga	15%	328.35 horas-hombre
Total	100%	horas-hombre

Tabla 29: Esfuerzo por Actividades

Esfuerzo Total (horas-hombres) 2517.35

Para la etapa de análisis y diseño se requiere de un esfuerzo de 656,7 horas-hombres, si se considera que trabajan 1 personas, 48 horas como promedio en la semana, esta etapa debe terminarse en aproximadamente 3 meses aproximadamente.

2.3 Análisis

El Análisis es el flujo de trabajo donde se refinan y estructuran los requisitos obtenidos con anterioridad, con el objetivo de facilitar la comprensión, preparación y modificación de los mismos. Un modelo de análisis es aquel que estructura los requisitos de un modo que facilita su comprensión y puede considerarse además como una primera aproximación al Modelo del Diseño.

2.3.1 Diagrama de clases del análisis

Un diagrama de clases del análisis es un artefacto en el que se representan los conceptos del dominio del problema. Representan las cosas del mundo real, no de la implementación automatizada.

En general se siguen directrices muy parecidas a las que se usan en la construcción del modelo conceptual. Este se realiza para cada uno de los casos de uso del sistema y muestra las clases participantes que se clasifican en tres tipos: interfaz, controladoras y entidades.

Clases del Análisis		
Nombre	Características	Representación
Interfaz	Modelan la interacción entre el sistema y los actores. Cada interfaz debe asociarse con al menos un actor y viceversa.	 Clase Interfaz
Controladora	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
Entidad	Modelan la información que posee una vida larga y que a menudo es persistente.	 Clase Entidad

Tabla 30: Clases del análisis.

A continuación se muestra el diagrama de clases del análisis del caso de uso Registrar entrada de materiales, los diagramas restantes se pueden consultar en el **Anexo IV**.

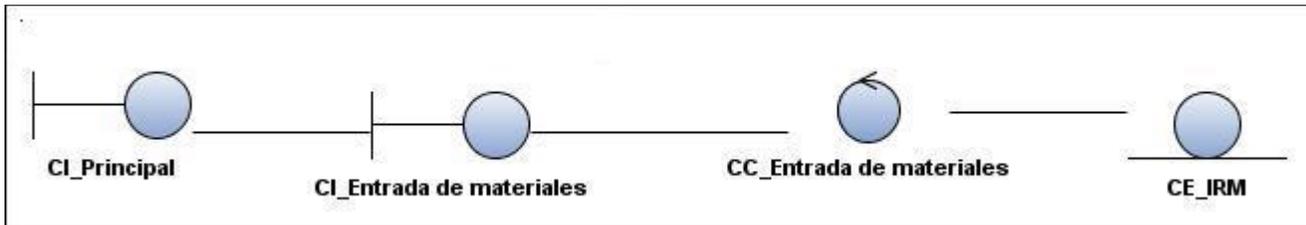


Figura 5: Diagrama de clases del análisis del CUS Registrar entrada de materiales

2.3.2 Diagrama de interacción

Los diagramas de interacción muestran como se comunican los objetos en una interacción (los objetos interactúan para realizar colectivamente los servicios por las aplicaciones). Existen dos tipos de diagramas:

Diagramas de Secuencia: Resaltan la ordenación temporal de los mensajes que en este caso no se realiza el modelado de los mismos debido a la extensión del trabajo.

Diagramas de Colaboración: Resaltan la organización de los objetos que participan en una interacción. A continuación se muestra como ejemplo un diagrama de colaboración y en el **Anexo V** se pueden consultar los diagramas restantes para cada uno de los escenarios de los casos de usos.

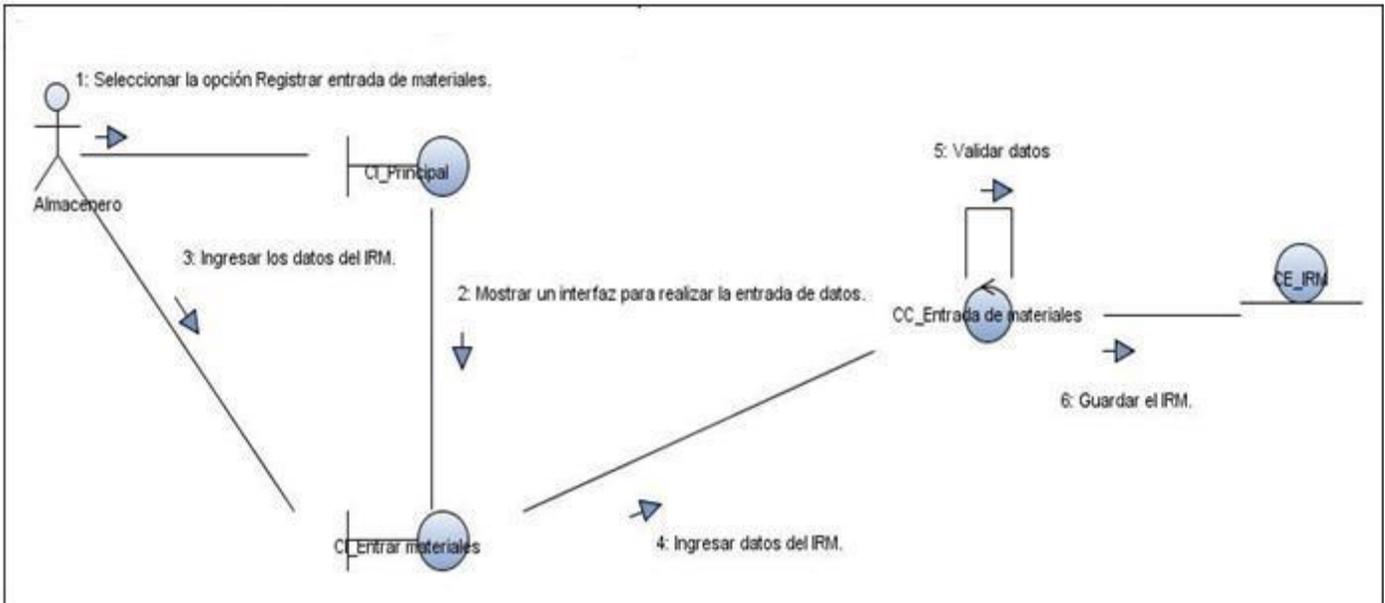


Figura 6: Diagrama de colaboración del CUS Registrar entrada de materiales.

2.4 Diseño

En el diseño se determina la arquitectura general del sistema y su comportamiento dinámico, adaptando la especificación realizada en la etapa anterior. En esta fase se establece el comportamiento dinámico del sistema, es decir, como debe reaccionar ante los acontecimientos.

El resultado obtenido de la etapa de Diseño facilita enormemente la implementación posterior del sistema, pues proporciona la estructura básica del sistema y como los diferentes componentes actúan y se relacionan entre ellos.

2.4.1 Patrones del diseño

Los patrones son soluciones a problemas recurrentes que ocurren una y otra vez en nuestro entorno. Se pueden considerar como recetas para solucionar varias veces un problema del mismo tipo. Los desarrolladores lo usan como una forma de reutilizar la experiencia, clasificando las soluciones con términos de común denominación.

En el diseño de la propuesta de solución se tiene en cuenta un patrón clásico del diseño web conocido como arquitectura MVC (modelo-vista-controlador) que implementa Symfony.

El modelo Vista Controlador es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista son las páginas HTML y el código que provee de datos dinámicos a la página; el modelo es la representación de los datos con que trabaja la aplicación y la lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista.

Modelo: es el objeto que representa los datos del programa. Maneja los datos y controla todas sus transformaciones. El modelo no tiene conocimiento específico de los controladores o de las vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas, y notificar a las vistas cuando cambia el modelo.

Vista: es el objeto que maneja la presentación visual de los datos representados por el modelo. Genera una representación visual del modelo y muestra los datos al usuario. Interactúa con el modelo a través de una referencia al propio modelo.

Controlador: es el objeto que proporciona significado a las órdenes del usuario, actuando sobre los datos representados por el modelo. Cuando se realiza algún cambio, entra en acción, bien sea por cambios en la información del modelo o por alteraciones de la vista. Interactúa con el modelo a través de una referencia al propio modelo.

Por todo lo expuesto anteriormente el patrón MVC ya que al separar la lógica de los datos y la presentación será mucho más sencillo realizar múltiples representaciones de los mismos datos o agregar nuevos tipos de datos según sea requerido. Además, el código de la aplicación será más legible, lo que permitirá un mantenimiento más sencillo y ofrecerá maneras más simples de probar el correcto funcionamiento de ésta.

En la realización del diseño se utilizaron además patrones GRASP, los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos y patrones GOF.

Patrones GRASP implementados

Experto: este patrón se tiene en cuenta para la asignación de responsabilidades a las clases de forma tal que las mismas contengan la información necesaria para poder ejecutar una acción específica. Symfony contribuye a este patrón, ya que a través del uso de la librería externa Doctrine realiza la capa de

abstracción en el modelo encapsulado toda la lógica de los datos y generando las clases con todas las funcionalidades comunes de las entidades. La lógica agregada a las mismas está en correspondencia con lo que plantea el patrón. El uso de este patrón permitirá a los objetos valerse de su propia información para hacer lo que se les pide, favorece la existencia de mínimas relaciones entre las clases, lo que permite contar con un sistema robusto y fácil de mantener.

Creador: este patrón se tiene en cuenta para la asignación de responsabilidades a las clases relacionadas con la creación de objetos, de forma tal que una instancia de un objeto solo pueda ser creada por el objeto que contiene la información necesaria para ello. El uso de este patrón permite crear las dependencias mínimas necesarias entre las clases, lo que favorece al mantenimiento del sistema y ofrece mejores oportunidades de reutilización.

Alta cohesión: este patrón se tiene en cuenta para realizar un diseño que evite contener clases con un alto grado de abstracción, que asuman responsabilidades que podían haber delegado a otros objetos o que tengan responsabilidades muy complejas. Se diseñaron las clases de forma tal que contengan las mínimas responsabilidades necesarias y colaboren con otras para llevar a cabo una tarea. Este patrón permitirá tener clases fáciles de mantener, de entender y reutilizar.

Controlador: este patrón se tiene en cuenta para realizar las asignaciones en cuanto al manejo de los eventos del sistema y definir sus operaciones. Symfony contribuye a la utilización de este patrón ya que define un controlador frontal que es el punto de acceso único a la aplicación y quien maneja las peticiones de los usuarios invocando a las operaciones necesarias para satisfacer las mismas.

Bajo Acoplamiento: el acoplamiento mide la fuerza con que una clase está conectada a otra, de esta forma una clase con bajo acoplamiento debe tener un número mínimo de dependencia con otras clases. El uso de los patrones Experto y Creador contribuyen al bajo acoplamiento entre las clases del sistema. Este patrón se tuvo en cuenta por la importancia que significa realizar un diseño de clases independientes que puedan soportar los cambios de una manera fácil y permitan la reutilización.

Patrones GOF implementados

Decorator: Este patrón permite añadir funcionalidad a una clase dinámicamente. Symfony implementa este patrón a través de la utilización de un *layout* o plantilla global, que almacena el código HTML que es

común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. De esta forma el contenido de las plantillas específicas de un módulo se integra en el *layout*, o si se mira desde el otro punto de vista, el *layout* decora la plantilla.

Singleton: Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Symfony define un objeto que guarda una referencia a todos los objetos del núcleo de Symfony relacionados con una petición dada, y ofrece un método de acceso único para cada uno de ellos.

Facade: Este patrón permite utilizar una interfaz común para un conjunto de interfaces del sistema, haciendo que éste sea más fácil de usar.

2.4.2 Diagramas de clases del diseño

En el diseño se modela el sistema para que soporte todos los requisitos, incluyendo los no funcionales y las restricciones que se le suponen. Se muestra el diagrama del diseño donde se describe la estructura del sistema mostrando sus clases, atributos y las relaciones entre ellos.

Este diagrama es muy utilizado durante el diseño de los sistemas, y a partir de él se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargarán del funcionamiento y la relación entre uno y otro. A continuación se muestra el diagrama de clases del diseño del caso de uso Registrar entrada de materiales, los diagramas restantes se pueden consultar en el **Anexo VI**.

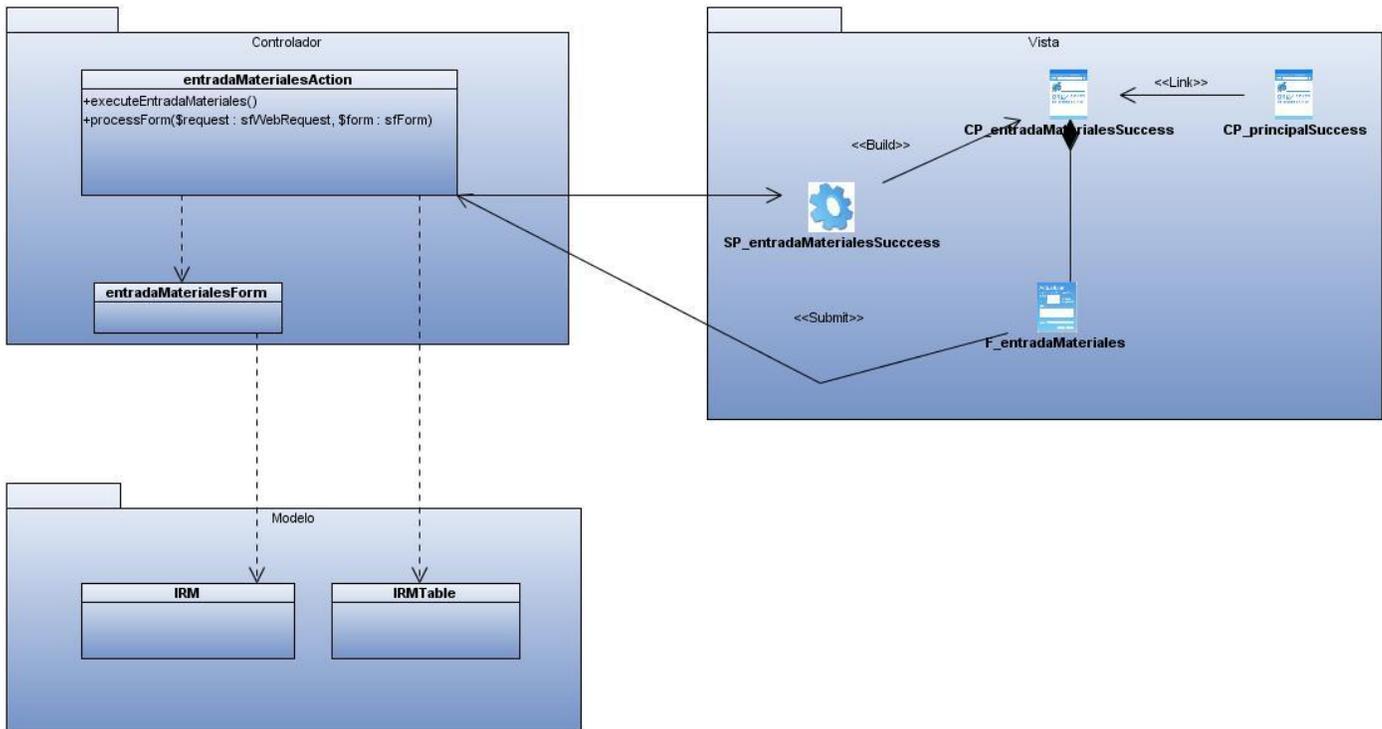


Figura 7: Diagrama de clases del diseño del CUS Registrar entrada de materiales.

2.5 Diseño de la base de datos

La persistencia es la capacidad que tienen los objetos de conservar su estado e identidad entre distintas ejecuciones del programa que los creó o de otros programas que accedan a ellos. En el diseño es donde se definen las clases persistentes puesto que no todas las clases del análisis son persistentes ya que muchas de ellas son clase temporales que dejan de existir cuando el programa termina. A continuación se muestra el modelo de la base de datos que corresponde con el sistema:

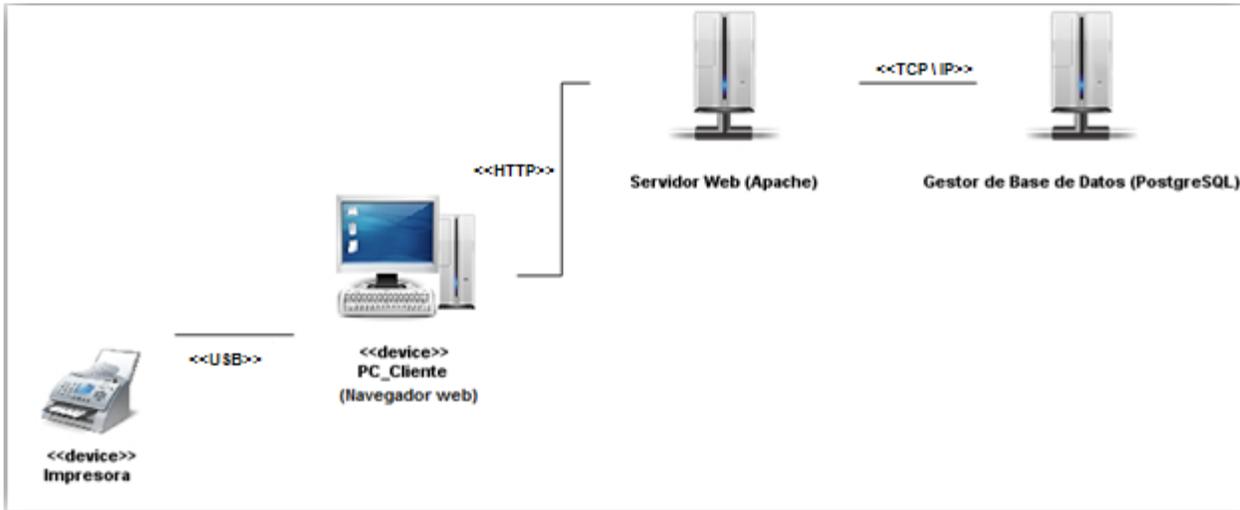


Figura 9: Diagrama de despliegue

2.7 Concepción general de ayuda

El acceso al sistema será realizado por trabajadores que tienen poco o escaso conocimiento informático, que se verán a menudo trabajando en un ambiente bastante dinámico, por lo que deben estar asistidos en cada momento con la información necesaria. La existencia de un manual de usuario donde se explique el funcionamiento, los requerimientos y una lista de preguntas frecuentes con sus respuestas, será un elemento de mucha utilidad, acompañado por una ayuda local de la página donde se encuentre el usuario o accesible en el propio menú de los servicios del sistema.

2.8 Tratamiento de errores

Como fundamento de la aplicación para la prevención de errores provocados por entradas incorrectas por parte de los usuarios se propone que estos sólo realicen las entradas mínimas necesarias dentro del dominio del sistema, una de las técnicas empleadas es conducir a los usuarios solamente por los caminos que concuerden con las funcionalidades del sistema permitidas, inhabilitando el resto de los controles de la interfaz que no necesiten ser utilizados en cada una de las operaciones. Se debe realizar en todo momento un correcto uso de las buenas prácticas de validación de entradas tanto del lado del cliente y como del servidor. Lo que garantizará la integridad del flujo y almacenamiento de la información enviada por el usuario. Se propone además, que exista un mecanismo de réplicas de mensajes que permitan conocer los detalles de las excepciones lanzadas por el sistema en cada momento, realizando el

tratamiento de las mismas e impidiendo que errores conocidos dejen inhabilitadas temporal o radicalmente las funcionalidades del sistema.

2.9 Conclusiones

En este capítulo se finaliza la etapa de análisis y diseño del sistema obteniendo un modelo más detallado de la solución propuesta. Se realizaron los diagramas de clases del análisis y de clases del diseño del sistema así como los diagramas de colaboración para cada una de las funcionalidades descritas, lo que permite una idea más específica del sistema que se propone. Se describe la arquitectura y los patrones que se tuvieron en cuenta para el diseño de los casos de uso. Se muestra además el modelo de la base de datos que satisfacen las necesidades del sistema y que son de gran importancia para la implementación del mismo, así como el diagrama de despliegue.

Conclusiones

Con el desarrollo de este trabajo se arriba a las siguientes conclusiones:

- ✓ Se llevó a cabo un profundo estudio de los sistemas de gestión almacenes en Cuba y en el mundo.
- ✓ Se realizaron entrevistas para conocer cómo se llevan a cabo los procesos de Gestión de Almacenes en la CTR.
- ✓ Se estudiaron las tecnologías, metodologías, lenguajes y herramientas necesarias para realizar el análisis, diseño y posterior desarrollo del sistema.
- ✓ Se realizó un estudio de la arquitectura propuesta.
- ✓ Se obtuvieron los artefactos necesarios, según la metodología de desarrollo de software seleccionada (RUP), para implementar la propuesta de solución.

Por todo lo antes expuesto se concluye que se le ha dado cumplimiento al objetivo general de la investigación: elaborar el análisis y diseño de un sistema para la gestión de almacenes en la Central Termoeléctrica Renté.

Recomendaciones

A partir del estudio realizado en la presente investigación y teniendo en cuenta las ideas que surgieron durante el desarrollo del análisis y diseño del sistema, se realizan las siguientes recomendaciones:

- ✓ Realizar la implementación del módulo teniendo en cuenta la propuesta de diseño obtenida en la presente investigación.
- ✓ Continuar el estudio de los procesos que se llevan a cabo en la CTR con el objetivo de agregarle nuevas funcionalidades al sistema.

Bibliografía

1. **HERRERO, JAIME.** *Una brújula para todos sus artículos.*
2. **Andino, MSc. Esteban Negrín Barroso MSc. Milagro Rodríguez.** *Las TIC y su papel en los procesos de Innovación en el ámbito económico local.* Camagüey : s.n.
3. Definición ABC. *Definición ABC.* [En línea] 15 de Enero de 2010.
<http://www.definicionabc.com/general/gestion.php>.
4. SISTEMAS DE CALIDAD EN LA OBRA PÚBLICA. [aut. libro] TRADECO INFRAESTRUCTURA S.A. DE C.V. Guadalupe : s.n.
5. ADAIA. [En línea] <http://www.leuter.cl/productos/adaia.aspx>.
6. SAP. *SOFTWARE DE GESTIÓN DE ALMACENES.* [En línea]
http://www.sap.com/chile/business_management_software/warehouse_managment.epx.
7. **Stallman, Richard M.** *Software libre para una sociedad libre.* Escrito originalmente en 1996.
8. **Díaz, Camilo Javier Solis Álvarez – Roberth Gustavo Figueroa.** *Metodologías Tradicionales vs. Metodologías Ágiles.*
9. **Rumbaugh, J. y Jacobson, I. y Booch, G.** Pdf: "El Proceso Unificado de Desarrollo de Software". Barcelona : s.n., 2000. págs. Pág. 106- 110.
10. **José H. Canós, Patricio Letelier y M^a Carmen Penadés.** *Métodologías Ágiles en el Desarrollo de Software.* Valencia : s.n.
11. **Orallo, Enrique Hernández.** *El Lenguaje Unificado de Modelado.*
12. **Navarrete, ING. Rosa.** *Sistema de administración de ventas de paquetes turísticos.* 2006.
13. Sitio de descarga. [En línea]
http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.

14. *Symfony la guía definitiva*. 2008. pág. 7.
15. maestro del web. [En línea] <http://www.maestrosdelweb.com/editorial/guia-zend/>.
16. Desarrollo en Web. *Desarrollo en Web*. [En línea] 22 de Octubre de 2008. [Citado el: 10 de Mayo de 2010.] <http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.
17. *Revista de Software Libre ATIX*. **López, Esteban Saavedra**. 2009.
18. Introducción a Apache. *Introducción a Apache*. [En línea] http://linux.ciberaula.com/articulo/linux_apache_intro/.
19. Diferencias entre PHP ASP y otros lenguajes. *Diferencias entre PHP ASP y otros lenguajes*. [En línea] 8 de julio de 2008. http://www.tufuncion.com/diferencias_lenguajes.
20. *Lenguajes de programación*.
21. LENGUAJES WEB. *LENGUAJES WEB*. [En línea] 12 de Octubre de 2009. <http://ming.crearblog.com/lenguajes-web/>.
22. **Artaza, Dr. Diego Lz. de Ipiña Gz. de**. *AJAX: Asynchronous JavaScript and XML (AJAX)*.
23. **Valdes, Damian Adriel Perez**. *maestrosdelweb*. *maestrosdelweb*. [En línea] 3 de Noviembre de 2007. <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
24. PostGreSQL vs. MySQL. *PostGreSQL vs. MySQL*. [En línea] Daniel Pecos. http://danielpecos.com/docs/mysql_postgres/x57.html.
25. Taringa. *Taringa*. [En línea] 14 de Octubre de 2008. <http://www.taringa.net/posts/info/1649223/Que-IDE-usas---JAVA.html>.
26. **Martínez, Jairo Chapela**. *Introducción al entorno de desarrollo Eclipse*. 2007.
27. Netbeans . *Netbeans* . [En línea] <http://dadisokd.net.au.net/index.php/component/content/article/44-software-libre-dadisokd/94-netbeans-dadisokd>.

28. **Acuña, Karennny Brito.** *SELECCIÓN DE METODOLOGÍAS DE DESARROLLO PARA APLICACIONES WEB EN LA FACULTAD DE INFORMÁTICA DE LA UNIVERSIDAD DE CIENFUEGOS.* Cienfuegos : s.n., 2009.

29. GNU Operating System. [En línea] <http://www.gnu.org/home.es.html>.

30. Ivanex. [En línea] <http://ivanex.wikidot.com/metodologia>.

31. Lenguajes de Web. *Leguanjes de Web.* [En línea] <http://ming.crearblog.com/lenguajes-web/>.

Pdf: Almacen 1, 2, 3 Manual de almacenes.

doc: **Título:** Técnicas de Estimación **Tipo de clase:** Clase Teórica-Práctica 2.
<http://eva.uci.cu/mod/resource/view.php?id=22433>

Autor: Ing. Jose Angel Franco Navarro. pdf: UML en acción. Modelando Aplicaciones Web. Cuba

Pdf: Juan Palacios.” Adaptando los Procesos de la Empresa, Flexibilidad con Scrum, Principios de Diseño y implantación de campos de Scrum” (2008).

<http://www.gnu.org/home.es.html>

Trabajos citados

1. **HERRERO, JAIME.** *Una brújula para todos sus artículos.*
2. **Andino, MSc. Esteban Negrín Barroso MSc. Milagro Rodríguez.** *Las TIC y su papel en los procesos de Innovación en el ámbito económico local.* Camagüey : s.n.
3. Definición ABC. *Definición ABC.* [En línea] 15 de Enero de 2010. <http://www.definicionabc.com/general/gestion.php>.
4. SISTEMAS DE CALIDAD EN LA OBRA PÚBLICA. [aut. libro] TRADECO INFRAESTRUCTURA S.A. DE C.V. Guadalupe : s.n.
5. ADAIA. [En línea] <http://www.leuter.cl/productos/adaia.aspx>.
6. SAP. *SOFTWARE DE GESTIÓN DE ALMACENES.* [En línea] http://www.sap.com/chile/business_management_software/warehouse_managment.epx.
7. **Stallman, Richard M.** *Software libre para una sociedad libre.* Escrito originalmente en 1996.
8. **Díaz, Camilo Javier Solis Álvarez – Roberth Gustavo Figueroa.** *Metodologías Tradicionales vs. Metodologías Ágiles.*
9. **Rumbaugh, J. y Jacobson, I. y Booch, G.** Pdf: "El Proceso Unificado de Desarrollo de Software". Barcelona : s.n., 2000. págs. Pág. 106- 110.
10. **José H. Canós, Patricio Letelier y M^a Carmen Penadés.** *Métodologías Ágiles en el Desarrollo de Software.* Valencia : s.n.
11. **Orallo, Enrique Hernández.** *El Lenguaje Unificado de Modelado.*
12. **Navarrete, ING. Rosa.** *Sistema de administración de ventas de paquetes turísticos.* 2006.

13. Sitio de descarga. [En línea] http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_%5Bcuenta_de_Plataforma_de_Java_14715_p/.
14. *Symfony la guía definitiva*. 2008. pág. 7.
15. maestro del web. [En línea] <http://www.maestrosdelweb.com/editorial/guia-zend/>.
16. Desarrollo en Web. *Desarrollo en Web*. [En línea] 22 de Octubre de 2008. [Citado el: 10 de Mayo de 2010.] <http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.
17. *Revista de Software Libre ATIX*. **López, Esteban Saavedra**. 2009.
18. Introducción a Apache. *Introducción a Apache*. [En línea] http://linux.ciberaula.com/articulo/linux_apache_intro/.
19. Diferencias entre PHP ASP y otros lenguajes. *Diferencias entre PHP ASP y otros lenguajes*. [En línea] 8 de julio de 2008. http://www.tufuncion.com/diferencias_lenguajes.
20. *Lenguajes de programación*.
21. LENGUAJES WEB. *LENGUAJES WEB*. [En línea] 12 de Octubre de 2009. <http://ming.crearblog.com/lenguajes-web/>.
22. **Artaza, Dr. Diego Lz. de Ipiña Gz. de**. *AJAX: Asynchronous JavaScript and XML (AJAX)*.
23. **Valdes, Damian Adriel Perez**. maestrosdelweb. *maestrosdelweb*. [En línea] 3 de Noviembre de 2007. <http://www.maestrosdelweb.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>.
24. PostGreSQL vs. MySQL. *PostGreSQL vs. MySQL*. [En línea] Daniel Pecos. http://danielpecos.com/docs/mysql_postgres/x57.html.
25. Taringa. *Taringa*. [En línea] 14 de Octubre de 2008. <http://www.taringa.net/posts/info/1649223/Que-IDE-usas---JAVA.html>.
26. **Martínez, Jairo Chapela**. *Introducción al entorno de desarrollo Eclipse*. 2007.

27. Netbeans . *Netbeans* . [En línea] <http://dadisokd.netau.net/index.php/component/content/article/44-software-libre-dadisokd/94-netbeans-dadisokd>.
28. **Acuña, Karennny Brito.** *SELECCIÓN DE METODOLOGÍAS DE DESARROLLO PARA APLICACIONES WEB EN LA FACULTAD DE INFORMÁTICA DE LA UNIVERSIDAD DE CIENFUEGOS.* Cienfuegos : s.n., 2009.
29. GNU Operating System. [En línea] <http://www.gnu.org/home.es.html>.
30. Ivanex. [En línea] <http://ivanex.wikidot.com/metodologia>.
31. Lenguajes de Web. *Leguanjes de Web.* [En línea] <http://ming.crearblog.com/lenguajes-web/>.

Glosario de términos

Comprobante contable (CC): posibilita la conciliación contable de cada una de las operaciones registradas según la manera de emitirlos definido por la unidad.

Documento de Reclamación (DR): es un documento donde se llenan todos los datos cuando se le va a hacer una reclamación de uno o varios materiales al proveedor.

Documento de Solicitud de Materiales (DSM): es un documento donde se llenan los datos del o los materiales que van a ser solicitados.

Factura: Es un documento que trae el proveedor con la relación del o los materiales que van a ser recepcionados posteriormente.

Informe de Faltante de Materiales (IFM): es el documento donde se llenan los datos del o los materiales que no existen en el almacén.

Informe de materiales Defectuosos (IMD): es el documento donde se llenan los datos del o los materiales que van a ser devueltos a la sección central.

Informe de Recepción de Materiales (IRM): documento donde se recoge todos los datos sobre la entrada de materiales al almacén.

Solicitud de Compra de Materiales (SCM): es un documento donde se llenan los datos del o los materiales que van a ser solicitados por la no existencia de esos materiales en el almacén.

Solicitud de Información de Materiales (SIM): es un documento donde se solicita información de los materiales al almacenero.

Tarjeta de Estiba (TE): tarjeta que existe en el almacén para cada tipo de material, en ella se lleva el control del mismo, registrando cada salida y entrada desde y hacia el almacén.

Vale de Entrega (VE): es un documento que avala las operaciones de salida de los materiales entre las secciones.