

Universidad de las Ciencias Informáticas

Facultad 1



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

Título: Análisis y diseño del Sistema de Gestión de los movimientos de productos de la Unión de Empresas de Recuperación de Materias Primas.

Autoras: Aliuska Pupo Poll.

Yilian Sánchez Contreras.

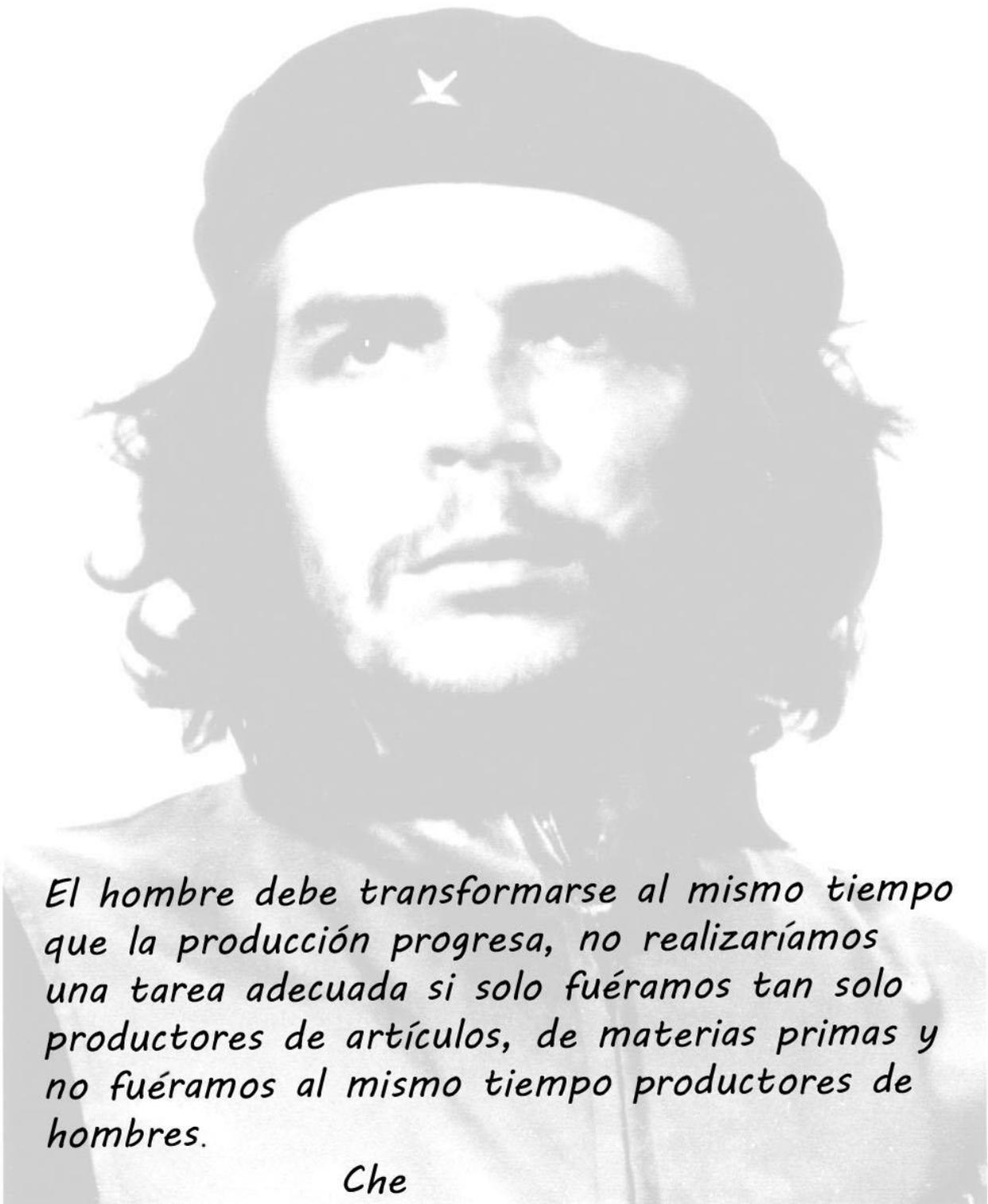
Tutoras: Ing. Marianny Hernández Batista.

Ing. Yudalys Junco Iglesias.

Ciudad de La Habana, Cuba

Junio, 2010

“Año 52 de la Revolución”



El hombre debe transformarse al mismo tiempo que la producción progresa, no realizaríamos una tarea adecuada si solo fuéramos tan solo productores de artículos, de materias primas y no fuéramos al mismo tiempo productores de hombres.

Che

Declaración de autoría

Declaramos que somos las únicas autoras de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) a que hagan el uso que estime pertinente con el mismo.

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

Aliuska Pupo Poll

Yilian Sánchez Contreras

Ing. Marianny Hernández Batista

Ing. Yudalys Junco Iglesia

“A nuestros padres, porque todo lo que somos es el fruto de su sacrificio y amor infinito.”

Aliuska:

“A todos los que han hecho posible el sueño de muchos años.”

*Ante todo quiero agradecer a **mis tutoras**, por apoyarme y ayudarme en todo momento y por ser más que mis tutoras, mis amigas.*

*A **mi mamá, mi nanita** del alma y **mi abuela querida** por llenarme de tanto amor, por darme las fuerzas para seguir adelante, por ser las mejores madres del mundo, sin ustedes nada de esto hubiera sido posible, las amo.*

*A **mi chuchi, mi novio** por ser tan especial, tan único, por su preocupación constante y por todo el amor que siempre me ha transmitido, te amo.*

*A **mi hermanito** del alma, por estar ahí siempre para mí, por su amor incondicional y su confianza en mí, te adoro.*

*A **mis tíos, primos, abuelos, mi cuñada, a Luis y a toda mi familia en general**, por ser como son, por quererme y apoyarme en cada una de mis decisiones, por estar a mi lado en cualquier circunstancia de la vida.*

*A **la yisy** por su apoyo incondicional, sus constantes consejos, por estar a mi lado siempre, por todas las alegrías, tristezas y el tiempo compartido .Mil gracias de todo corazón.*

*A **lia** por ser como una hermana para mí, gracias por ser mi amiga y por todos los momentos inolvidables que pasamos en la vida.*

*A **mis amigos Kendry, Angel y Rey** por permitirme contar con ustedes siempre que los necesité.*

*A todos **mis compañeros de aula** durante estos 5 años.*

Gracias

Yilian:

A **mi papá** por ser como es, por creer en mí en todo momento y por estar siempre ahí para mí.

A **mi mamá** por apoyarme en todo, por ser mi inspiradora en la vida como mujer y como madre, por quererme tanto, y sobre todo por luchar por mí contra todos los obstáculos que se han presentado en mi camino.

A **mi hermana** por ser mi amiga, mi compañera, por darme siempre su apoyo, por sacrificarse por mí y ser la mejor hermana del mundo.

A **mi novio** por ser tan paciente conmigo, por amarme y apoyarme, por los esfuerzos que ha tenido que hacer por mí, pero lo más importante por amarme.

A **mis abuelos** porque a pesar de no estar junto a mí, sé que estarían muy orgullosos de mí, en especial mi abuelita Juana. Y debo agradecerles porque sé que estén donde estén me guían y me apoyan.

A toda **mi familia** porque siempre escuchan mis sueños y me animan a alcanzarlos.

A **mis tutoras** por su ayuda incondicional, por guiarme, sin ellas esto no hubiera sido posible.

A todos aquellos profesores: Que han contribuido en mi formación.

A **la Revolución** por permitirnos alcanzar nuestros sueños.

Resumen

Con el proceso de informatización de la sociedad cubana, el desarrollo de aplicaciones web se ha convertido en una gran potencialidad para la gestión de la información. La Unión de Empresas de Recuperación de Materias Primas es la organización estatal rectora en Cuba de la recuperación, procesamiento y comercialización de desechos útiles en el territorio nacional y en el exterior. A esta organización se le hace cada día más difícil controlar la información que generan los movimientos de productos que se llevan a cabo en las diferentes empresas del país, esto se debe a la poca automatización con la que cuenta esta casa matriz. De ahí que el objetivo del presente trabajo de diploma consiste en realizar el análisis y diseño del sistema de gestión de los movimientos de productos de la Unión de Empresas de Recuperación de Materias Primas, que contribuya a una mejor implementación de las funcionalidades que debe poseer el sistema. Para desarrollar la propuesta se selecciona el Proceso Unificado de Desarrollo de software (RUP) como metodología de desarrollo, el Lenguaje Unificado de Modelado (UML) como lenguaje de modelado, Visual Paradigm como herramienta de Ingeniería de Software Asistida por Ordenador (CASE) a utilizar para el modelado, Symfony como marco de trabajo y como lenguaje de programación PHP. El diseño obtenido será el punto de partida para la posterior implementación del Sistema de Gestión de los movimientos de productos de la Unión de Empresas de Recuperación de Materias Primas.

Palabras claves:

Gestión, Movimientos de productos.

Índice de contenido.

Introducción	1
Capítulo 1: Fundamentación teórica	6
1. Introducción.....	6
1.1 Aplicaciones web.....	6
1.2 Sistema de gestión empresarial.....	7
1.3 Sistemas de Control de Almacenes y Ventas a nivel mundial	7
1.3.1 Trazabilidad 3000	7
1.3.2 Secop PRO.....	8
1.3.3 Gestión 3000.....	8
1.3.4 ¿Por qué no se utilizan estos sistemas?.....	8
1.4 Sistemas de Control de Almacenes y Ventas en Cuba.....	8
1.4.1 Rodas XXI Versión 3.0	8
1.4.2 Siscont5	9
1.4.3 El SABIC	9
1.4.4 ¿Por qué no se utilizan estos sistemas?.....	10
1.5 Sistemas de Control de Almacenes y Ventas en la Universidad de Ciencias Informáticas.....	10
1.5.1 Asset NS.....	10
1.5.2¿Por qué no se utiliza este sistema?	11
1.6 Metodología de desarrollo del software	11
1.6.1 Crystal Methodologies	11
1.6.2 Programación Extrema.....	12
1.6.3 Proceso unificado de desarrollo de software	12
1.6.4 Selección de la metodología a utilizar	13
1.7 Lenguaje de modelado	14

1.7.1 UML.....	14
1.8 Leguajes y tecnologías del lado del cliente	14
1.8.1 XML.....	14
1.8.2 HTML	15
1.8.3 CSS.....	15
1.8.4 JavaScript.	16
1.8.5 Selección del lenguaje del lado del cliente	17
1.9 Lenguajes del lado del servidor	17
1.9.1 ASP	17
1.9.2 JSP.....	17
1.9.3 PHP	18
1.9.4 Selección del lenguaje del lado del servidor	18
1.10 Sistemas gestores de base de datos	19
1.10.1 MySQL	19
1.10.2 PostgreSQL	20
1.11.3 Oracle.....	20
1.10.4 Selección del gestor de base de datos a utilizar.....	21
1.11 Herramientas de modelado	21
1.11.1 Visual Paradigm	22
1.11.2 Rational Rose	22
1.11.3 Selección de la herramienta de modelado a utilizar	23
1.12 Servidor Web.	23
1.12.1 IIS	23
1.12.2 Thttpd	24
1.12.4 Apache	24

1.12.5 Selección del servidor web.....	25
1.13 Herramientas a utilizar.....	25
1.13.1 NetBeans IDE	25
1.13.2 DBDesigner 4.....	25
1.13.3 EMS PostgreSQL Manager 3.....	26
1.14 Marco de Trabajo	26
1.14.1 Symfony 1.4	27
Conclusiones.....	28
Capítulo 2: Características del Sistema -----	29
2. Introducción.....	29
2.1 Descripción del negocio	29
2.1.1 Modelo de negocio	29
2.1.2 Flujo actual de los procesos involucrados en el campo de acción.....	29
2.1.2 Actores del negocio	30
2.1.3 Trabajadores del negocio	30
2.1.4 Diagrama de casos de uso del negocio	31
2.1.5 Diagrama de actividades	31
2.1.6 Descripciones textuales.....	31
2.1.7 Diagrama de clases del modelo de objetos	35
2.2 Especificación de los requisitos del sistema.....	36
2.2.1 Requerimientos Funcionales	36
R1. Autenticar usuario.....	36
R2. Gestionar rol	36
R3. Gestionar permiso	36
R4. Gestionar usuario	37

R5. Gestionar organismo	37
R6. Gestionar empresa.....	37
R7. Gestionar moneda	37
R8. Gestionar tipo de producto	38
R9. Gestionar producto	38
R10. Gestionar venta	38
R11. Registrar compra de producto.....	39
2.2.2 Requerimientos no funcionales	39
1. Usabilidad	39
2. Rendimiento	39
3. Portabilidad	39
4. Seguridad.....	40
5. Apariencia o interfaz externa	40
6. Soporte.....	40
7. Requerimientos de Software	41
8. Requerimientos de Hardware	41
2.3 Modelación del sistema.....	41
2.3.1 Diagrama de casos de uso del sistema	42
2.3.2 Descripción de los casos de uso.....	42
2.4 Estudio de la factibilidad.....	46
2.4.1 Planificación mediante Puntos de Casos de Uso	46
2.4.2 Beneficios tangibles e intangibles	52
Conclusiones.....	53

Capítulo 3: Análisis y diseño del sistema	54
3. Introducción.....	54
3.1 Modelo de Análisis	54
3.1.1 Diagramas de Clases del Análisis	54
3.1.2 Diagramas de Colaboración	59
3.2 Modelo de Diseño.....	61
3.2.1 Estructura de Symfony	61
3.2.2 Patrones	63
3.2.3 Diagramas de Clases del Diseño.....	65
3.3 Diseño de la base de datos	66
3.3.1 Diagrama de clases persistentes.....	66
3.3.2 Modelo de datos (Ver Anexo 6).....	66
3.3.3 Descripción de las tablas de la base de datos.....	66
3.4 Diagrama de despliegue	66
Conclusiones.....	68
Conclusiones generales	69
Recomendaciones	70
Referencias bibliográficas	71
Bibliografía	74
Glosario de términos	75
Anexos	<i>Error! Bookmark not defined.</i>

Índice de figuras.

Figura 1. Diagrama de Casos de Uso del Negocio.....	31
Figura 2. Modelo de objetos del negocio.	35
Figura 3. Diagrama de Casos de Uso del Sistema.....	42
Figura 4. Diagrama de clases del Análisis CU1_AutenticarUsuario.....	54
Figura 5. Diagrama de clases del Análisis CU2_GestionarRol.	55
Figura 6. Diagrama de clases del Análisis CU3_GestionarPermiso.	55
Figura 7. Diagrama de clases del Análisis CU4_GestionarUsuario.	56
Figura 8. Diagrama de clases del Análisis CU5_GestionarOrganismo.....	56
Figura 9. Diagrama de clases del Análisis CU6_GestionarEmpresa.....	57
Figura 10. Diagrama de clases del Análisis CU7_GestionarMoneda.....	57
Figura 11. Diagrama de clases del Análisis CU8_GestionarTipoProducto.....	58
Figura 12. Diagrama de clases del Análisis CU9_GestionarProducto.....	58
Figura 13. Diagrama de clases del Análisis CU10_GestionarVenta.....	59
Figura 14. Diagrama de clases del Análisis CU11_RegistrarCompraProducto.....	59
Figura 15. Diagrama de colaboración "Gestionar Venta. Escenario Crear nuevaVenta".	60
Figura 16. Diagrama de colaboración "Gestionar Venta. Escenario ModificarVenta".	60
Figura 17. Diagrama de colaboración "Gestionar Venta. Escenario EliminarVenta".	60
Figura 18. Diagrama de Clases del Diseño CU_GestionarVenta.....	65
Figura 19. Diagrama de despliegue.	67
Figura 20. Diagrama de actividades "Emitir reportes".	Error! Bookmark not defined.
Figura 21. Diagrama de actividades "Registrar venta".	Error! Bookmark not defined.
Figura 22. Diagrama de actividades "Registrar ventas diarias en valores.	Error! Bookmark not defined.
Figura 23. Diagrama de actividades "Registrar compra".	Error! Bookmark not defined.
Figura 24. Diagrama de colaboración "Autenticar Usuario".	Error! Bookmark not defined.

Figura 25. Diagrama de colaboración “Gestionar Rol. Escenario Crear nuevo Rol”. .. **Error! Bookmark not defined.**

Figura 26. Diagrama de colaboración “Gestionar Rol. Escenario Modificar Rol”. **Error! Bookmark not defined.**

Figura 27. Diagrama de colaboración “Gestionar Rol. Escenario Eliminar Rol”. **Error! Bookmark not defined.**

Figura 28. Diagrama de colaboración “Gestionar Permiso. Escenario Crear nuevo Permiso”. **Error! Bookmark not defined.**

Figura 29. Diagrama de colaboración “Gestionar Permiso. Escenario Modificar Permiso”..... **Error! Bookmark not defined.**

Figura 30. Diagrama de colaboración “Gestionar Permiso. Escenario Eliminar Permiso”. **Error! Bookmark not defined.**

Figura 31. Diagrama de colaboración “Gestionar Usuario. Escenario Crear nuevo Usuario”. **Error! Bookmark not defined.**

Figura 32. Diagrama de colaboración “Gestionar Usuario. Escenario Modificar Usuario”. **Error! Bookmark not defined.**

Figura 33. Diagrama de colaboración “Gestionar Usuario. Escenario Eliminar Usuario”. . **Error! Bookmark not defined.**

Figura 34. Diagrama de colaboración “Gestionar Organismo. Escenario Crear nuevo Organismo”. **Error! Bookmark not defined.**

Figura 35. Diagrama de colaboración “Gestionar Organismo. Escenario Modificar Organismo”..... **Error! Bookmark not defined.**

Figura 36. Diagrama de colaboración “Gestionar Organismo. Escenario Eliminar Organismo”. **Error! Bookmark not defined.**

Figura 37. Diagrama de colaboración “Gestionar Empresa. Escenario Crear nueva Empresa”. **Error! Bookmark not defined.**

Figura 38. Diagrama de colaboración “Gestionar Empresa. Escenario Modificar Empresa”. **Error! Bookmark not defined.**

Figura 39. Diagrama de colaboración “Gestionar Empresa. Escenario Eliminar Empresa”..... **Error! Bookmark not defined.**

Figura 40. Diagrama de colaboración “Gestionar Moneda. Escenario Crear nueva Moneda”.....	Error! Bookmark not defined.
Figura 41. Diagrama de colaboración “Gestionar Moneda. Escenario Modificar Moneda”.....	Error! Bookmark not defined.
Figura 42. Diagrama de colaboración “Gestionar Moneda. Escenario Eliminar Moneda”.....	Error! Bookmark not defined.
Figura 43. Diagrama de colaboración “Gestionar Tipo de Producto. Escenario Crear nuevo Tipo de Producto”.....	Error! Bookmark not defined.
Figura 44. Diagrama de colaboración “Gestionar Tipo de Producto. Escenario Modificar Tipo de Producto”.....	Error! Bookmark not defined.
Figura 45. Diagrama de colaboración “Gestionar Tipo de Producto. Escenario Eliminar Tipo de Producto”.....	Error! Bookmark not defined.
Figura 46. Diagrama de colaboración “Gestionar Producto. Escenario Crear nuevo Producto”.....	Error! Bookmark not defined.
Figura 47. Diagrama de colaboración “Gestionar Producto. Escenario Modificar Producto”.....	Error! Bookmark not defined.
Figura 48. Diagrama de colaboración “Gestionar Producto. Escenario Eliminar Producto”.....	Error! Bookmark not defined.
Figura 49. Diagrama de colaboración “Registrar Compra de Producto”.....	Error! Bookmark not defined.
Figura 50. Diagrama de Clases del Diseño CU1_AutenticarUsuario.....	Error! Bookmark not defined.
Figura 51. Diagrama de Clases del Diseño CU2_GestionarRol.....	Error! Bookmark not defined.
Figura 52. Diagrama de Clases del Diseño CU3_GestionarPermiso.....	Error! Bookmark not defined.
Figura 53. Diagrama de Clases del Diseño CU4_GestionarUsuario.....	Error! Bookmark not defined.
Figura 54. Diagrama de Clases del Diseño CU6_GestionarOrganismo.....	Error! Bookmark not defined.
Figura 55. Diagrama de Clases del Diseño CU5_GestionarEmpresa.....	Error! Bookmark not defined.
Figura 56. Diagrama de Clases del Diseño CU7_GestionarMoneda.....	Error! Bookmark not defined.
Figura 57. Diagrama de Clases del Diseño CU8_GestionarTipoProducto.....	Error! Bookmark not defined.
Figura 58. Diagrama de Clases del Diseño CU9_GestionarProducto.....	Error! Bookmark not defined.

Figura 59. Diagrama de Clases del Diseño CU11_RegistrarCompraProducto. **Error! Bookmark not defined.**

Figura 60. Diagrama de clases persistentes..... **Error! Bookmark not defined.**

Figura 61: Modelo de datos. **Error! Bookmark not defined.**

Índice de tablas.

Tabla 2.1: Actor del negocio y su descripción.	30
Tabla 2.2: Trabajadores y su descripción.	30
Tabla 2.3: Descripción textual del Caso de Uso del Negocio "Emitir reporte ".	32
Tabla 2.4: Descripción textual del Caso de Uso del Negocio "Registrar ventas ".	33
Tabla 2.5: Descripción textual del Caso de Uso del Negocio "Registrar ventas diarias en valores".	34
Tabla 2.6: Descripción textual del Caso de Uso del Negocio "Registrar compra ".	35
Tabla 2.7: Descripción del CU Autenticar Usuario.	43
Tabla 2.8: Descripción del CU Gestionar Rol.	43
Tabla 2.9: Descripción del CU Gestionar Permiso.	43
Tabla 2.10: Descripción del CU Gestionar Usuario.	43
Tabla 2.11: Descripción del CU Gestionar Organismo.	44
Tabla 2.12: Descripción del CU Gestionar Empresa.	44
Tabla 2.13: Descripción del CU Gestionar Moneda.	44
Tabla 2.14: Descripción del CU Gestionar tipo de Producto.	45
Tabla 2.15: Descripción del CU Gestionar Producto.	45
Tabla 2.16: Descripción del CU Gestionar Venta.	45
Tabla 2.17: Descripción del CU Registrar compra de productos.	46
Tabla 2.18: Complejidad de los actores.	47
Tabla 2.19: Factor de peso de los actores sin ajustar.	47
Tabla 2.20: Complejidad de los casos de uso.	48
Tabla 2.21: Factor de peso de los casos de uso sin ajustar.	48
Tabla 2.22: Determinación del factor de Complejidad Técnica.	49
Tabla 2.23: Determinación del factor ambiente.	50
Tabla 2.24: Estimación del esfuerzo total.	51

Tabla 2.25: Descripción detallada del CUS_ Autenticar usuario.	Error! Bookmark not defined.
Tabla 2.26: Descripción detallada del CUS_ Gestionar roles.	Error! Bookmark not defined.
Tabla 2.27: Descripción detallada del CUS_ Gestionar permisos.	Error! Bookmark not defined.
Tabla2.28: Descripción detallada del CUS_ Gestionar usuario.	Error! Bookmark not defined.
Tabla2.29: Descripción detallada del CUS_ Gestionar organismo.	Error! Bookmark not defined.
Tabla 2.30: Descripción detallada del CUS_ Gestionar empresa.	Error! Bookmark not defined.
Tabla 2.31: Descripción detallada del CUS_ Gestionar moneda.	Error! Bookmark not defined.
Tabla 2.32: Descripción detallada del CUS_ Gestionar tipo de producto.	Error! Bookmark not defined.
Tabla 2.33: Descripción detallada del CUS_ Gestionar producto.	Error! Bookmark not defined.
Tabla 2.34: Descripción detallada del CUS_ Gestionar venta.	Error! Bookmark not defined.
Tabla 2.35: Descripción detallada del CUS_ Registrar compra de producto.	Error! Bookmark not defined.
Tabla 3.1: Descripción de la tabla de la BD “Tabla Empresa”.	Error! Bookmark not defined.
Tabla 3.2: Descripción de la tabla de la BD “Tabla Moneda”.	Error! Bookmark not defined.
Tabla 3.3: Descripción de la tabla de la BD “Tabla Tipo de Producto”.	Error! Bookmark not defined.
Tabla 3.4: Descripción de la tabla de la BD “Tabla Organismo”.	Error! Bookmark not defined.
Tabla 3.5: Descripción de la tabla de la BD “Tabla Venta”.	Error! Bookmark not defined.
Tabla 3.6 Descripción de la tabla de la BD “Tabla Producto”.	Error! Bookmark not defined.
Tabla 3.7: Descripción de la tabla de la BD “Nomenclador Medida”.	Error! Bookmark not defined.
Tabla 3.8: Descripción de la tabla de la BD “compra de productos”.	Error! Bookmark not defined.
Tabla 3.9: Descripción de la tabla de la BD “sf_guard_user”.	Error! Bookmark not defined.
Tabla 3.10: Descripción de la tabla de la BD “sf_guard_group”.	Error! Bookmark not defined.
Tabla 3.11: Descripción de la tabla de la BD “sf_guard_permission”.	Error! Bookmark not defined.

Introducción

Con el creciente auge de la informatización en todas las esferas de la sociedad y el desarrollo acelerado de las Tecnologías de la Información y las Comunicaciones (TIC), la mayoría de las empresas e instituciones necesitan informatizar los procesos de trabajo que desarrollan para ganar competitividad. Lamentablemente este proceso no se desarrolla con igual fuerza y rapidez en todas las empresas. “La Unión de Empresas de Recuperación de Materias Primas (UERMP), es la organización estatal rectora en Cuba de la recuperación, procesamiento y comercialización de desechos útiles en el territorio nacional y en el exterior. En la actualidad la Unión pertenece a la rama de la siderurgia, hoy con el nombre de Ministerio de la Industria Sideromecánica (SIME). Cuenta con 26 empresas clasificadas en: 15 empresas recuperadoras (1 en cada provincia y en el municipio especial Isla de la Juventud), 9 empresas especializadas y 2 exportadoras-importadoras” (1).

Los procesos de gestión de los movimientos de productos que se realizan en la UERMP son llevados a cabo por tres áreas de trabajo:

- ✓ Ferroso
- ✓ No ferroso
- ✓ No metálico

Estos procesos ocurren de la siguiente forma:

Ferroso

Los procesos de los movimientos de productos en esta área se gestionan mediante la aplicación Concilia, utilizando el sistema operativo MS-DOS. Esta aplicación mantiene una baja seguridad, implicando la pérdida de información y no cumple con las funcionalidades requeridas por la organización; además, no existen servicios de soporte a la aplicación por parte del proveedor.

No ferroso

Los procesos de gestión de los movimientos de productos en esta área se realizan mediante la aplicación SI – Estadístico 451, que utiliza el sistema operativo MS-DOS. Esta aplicación no cumple con las funcionalidades y la seguridad mínima requerida, posee una interfaz poco agradable al usuario y no existen servicios de soporte a la aplicación por parte del proveedor.

No metálico

En esta área los procesos de los movimientos de productos se efectúan manualmente en las Hojas de Cálculo Microsoft Excel. La información de los productos referentes a esta área es solicitada por teléfono, lo que resulta muy incómodo y extenuante para los encargados de registrar dichos procesos.

De manera general se puede decir que esta casa matriz no cuenta con un sistema informatizado que tenga la capacidad necesaria para manejar y almacenar la información de las empresas, pues las aplicaciones que utilizan poseen una tecnología obsoleta, ya que son ejecutadas sobre MS-DOS, el cual es un sistema operativo que ha quedado en desuso. Además, estas aplicaciones no proporcionan las funcionalidades requeridas por dicha entidad, pues:

- ✓ No realizan una validación de los datos de entrada.
- ✓ No permiten la eliminación y adición de campos, lo que ocasiona la entrada de datos ficticios, proporcionando irregularidades que afectan la contabilidad de esta organización.
- ✓ No brindan una interfaz amigable que ayude al usuario a utilizar el sistema.
- ✓ No cuentan con una base de datos para el control de la información, pues dicha información se maneja en las Hojas de Cálculo Microsoft Excel, lo cual dificulta la consolidación de toda la información que llega diariamente.

Teniendo en cuenta esta situación problémica se plantea como **problema a resolver**: ¿Cómo transformar los procesos de gestión de los movimientos de productos de la Unión de Empresas de Recuperación de Materias Primas, a un lenguaje entendible para los desarrolladores, que permita la posterior implementación del sistema?

Por lo que se define como **objeto de estudio**: los procesos que ocurren en la Unión de Empresas de Recuperación de Materias Primas. Por cuanto se especifica como **campo de acción**: los movimientos de productos de la Unión de Empresas de Recuperación de Materias Primas.

Para dar solución al problema planteado se define como **objetivo de la investigación**: realizar el análisis y diseño del sistema para la gestión de los movimientos de productos de la

Unión de Empresas de Recuperación de Materias Primas. Por lo que se plantea como **objetivos específicos:**

- ✓ Realizar un estudio de los diferentes sistemas de gestión de Control de Almacenes y Ventas existentes a nivel mundial.
- ✓ Seleccionar la metodología y herramientas a utilizar.
- ✓ Diseñar las clases del Sistema de Gestión de los movimientos de productos de la Unión de Empresas de Recuperación de Materias Primas.

Teniendo en cuenta los objetivos, se propone como **idea a defender:** el análisis y diseño del sistema para la gestión de los movimientos de productos de la Unión de Empresas de Recuperación de Materias Primas contribuirá a la futura implementación del Sistema de Gestión de los movimientos de productos de esta casa matriz.

Se tienen como **posibles resultados:**

- ✓ Una selección apropiada de las herramientas a utilizar.
- ✓ El análisis y diseño del sistema para la gestión de los movimientos de productos de la Unión de Empresas de Recuperación de Materias Primas.

Para lograr el cumplimiento de los objetivos se proponen las siguientes **tareas científicas:**

- ✓ Análisis de los sistemas existentes para la gestión de los movimientos de productos de la Unión de Empresas de Recuperación de Materias Primas.
- ✓ Estudio de la documentación existente sobre la gestión de los movimientos de productos de la Unión de Empresas de Recuperación de Materias Primas.
- ✓ Selección de las herramientas y tecnologías para el desarrollo del Sistema de Gestión de los movimientos de productos de la Unión de Empresas de Recuperación de Materias Primas.
- ✓ Entrevista a los responsables de la gestión de los movimientos de productos de la Unión de Empresas de Recuperación de Materias Primas.
- ✓ Levantamiento de requisitos funcionales y no funcionales para la gestión de los movimientos de productos de la Unión de Empresas de Recuperación de Materias Primas.
- ✓ Diseño de las clases del Sistema de Gestión de los movimientos de productos de la Unión de Empresas de Recuperación de Materias Primas.

Para llevar a cabo estas tareas se emplearon métodos teóricos y empíricos de la investigación científica.

Los **métodos teóricos** utilizados para cumplir con las tareas a desarrollar son:

Histórico lógico: Posibilita un mejor análisis histórico de los procesos de trabajo de los departamentos de la empresa, es decir, permite analizar la trayectoria de los procesos, desde su desenvolvimiento hasta las conexiones históricas más importantes. Es utilizado para realizar un estudio de las herramientas y sistemas similares a la propuesta para adquirir mayor experiencia y ser capaces de lograr un mejor análisis y diseño para un nuevo sistema.

Modelación: Partiendo de todo lo investigado y aprendido se realizan los modelos correspondientes a la metodología especificada para el sistema, que ayudaran a dar cumplimiento a las tareas de diseño de los procesos involucrados en la solución.

Analítico-sintético: Permite la separación de un todo en sus partes. Se apoya en que para conocer un fenómeno es necesario descomponerlo en sus partes y luego sintetizar toda la información. Se realizó un análisis de los procesos involucrados en la Unión, para extraer todas las funcionalidades requeridas en el sistema. Su utilización permite comprender mejor toda la información anterior, la cual será de gran utilidad para lograr un adecuado análisis y diseño del sistema.

Los **métodos empíricos** utilizados para obtener información sobre el objeto de estudio son:

Observación: Permite investigar los procesos externamente sin tener que llegar a la esencia de los mismos, lo cual sirve de ayuda para realizar un seguimiento del comportamiento de estos procesos. A través de la observación se permite el avance hacia el objetivo final.

Entrevistas: Permite realizar entrevistas con el fin de precisar el problema a resolver. Se entrevistó al jefe del departamento y a los especialistas de cada área donde se obtuvo la mayor cantidad de información sobre los problemas que presentan las aplicaciones existentes, las posibles soluciones a estos problemas y las nuevas funcionalidades para el sistema.

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

Capítulo 1 “Fundamentación teórica”: Se abordan los elementos teóricos conceptuales que sustentan el problema científico y los objetivos, se describen los procesos que se llevan a cabo en la organización y se realiza un breve estudio de otras estructuras y sistemas similares

al que se desea modelar. Además, se fundamenta la selección de tecnologías, herramientas y metodologías con las cuales se debe desarrollar el sistema.

Capítulo 2: “Características del sistema”: Se describen los requisitos obtenidos como resultado del levantamiento de requisitos, tanto funcionales como no funcionales; se describe el modelo del negocio, los requisitos y casos de uso del sistema; así, como la expansión de los casos de uso, además, se realiza el estudio de la factibilidad del sistema propuesto.

Capítulo 3: “Análisis y diseño del Sistema”: Se realiza el análisis y de diseño del sistema. Se establecen los patrones de diseño, la arquitectura a desarrollar, así, como el diagrama de despliegue de la aplicación.

Capítulo 1: Fundamentación teórica

1. Introducción

El desarrollo y las tendencias actuales en la rama de la informática han logrado que en el mundo ocurran una serie de cambios positivos, los cuales traen consigo una gran optimización de tiempo y de recursos. Las especializaciones del software están impulsando cada vez más el desarrollo de las tecnologías y metodologías que se utilizan con 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. Para diseñar un software se deben tener en cuenta ciertos parámetros como metodologías a usar, lenguajes de programación, gestores de base de datos, patrones arquitectónicos y de diseño.

En el presente capítulo se describen detalladamente los conceptos relacionados con el tema de la investigación. Se expone el resultado del estudio de los sistemas de gestión empresarial utilizados a nivel mundial. Se describen los lenguajes, la metodología de desarrollo y las herramientas propuestas.

1.1 Aplicaciones web

Las aplicaciones web son sistemas informáticos que utilizan los usuarios para acceder a un servidor web mediante una intranet o Internet. En la actualidad se han convertido en sistemas muy usados debido a dos razones: la gran práctica del navegador web como cliente ligero y la facilidad que brindan para actualizar y darle mantenimiento a las aplicaciones sin tener que distribuir ni instalar software en las computadoras de los usuarios.

“...una aplicación web será un sistema web donde la entrada del usuario (entrada de datos y navegación) afecta el estado del negocio. Esta definición intenta establecer que una aplicación web es un sistema de software con estado de negocio, y por tanto puede elaborarse utilizando la metodología de trabajo que propone RUP y utilizando el UML para los modelos” (2).

Una aplicación web genera una serie de páginas dinámicas en un formato estándar común como es Lenguaje de Marcado de Hipertexto (HTML) o el Lenguaje Extensible de Marcado de Hipertexto (XHTML), soportado por los navegadores web. Utilizan lenguajes interpretados del lado del cliente, para añadir elementos dinámicos a la interfaz de usuario. Para añadir más

funcionalidades, los desarrolladores usan lenguajes del lado del cliente, lo que proporciona mayor interacción con el cliente sin necesidad de cargar la página por cada petición.

1.2 Sistema de gestión empresarial

“Los sistemas de gestión empresarial son las herramientas que permiten a una compañía automatizar e integrar la mayor parte de los procesos de su negocio, compartir datos, producir y acceder a la información en tiempo real. Uno de los elementos clave para una organización es la mejora del flujo y procesamiento de la información; además, que el acceso a esta información se realice de manera rápida e interrelacionada. Entre sus ventajas se encuentran: la integración de la información y mejora de la comunicación entre diferentes áreas; la información disponible e inmediata para la toma de decisiones; las mejoras en la productividad y en la gestión de órdenes de compra, reducción de los costos por compras, transporte, logística y mantenimiento; la mejora en los tiempos de respuesta, así, como la rápida adaptación a los cambios, escalabilidad del sistema, seguridad definida por el usuario para el manejo de información y reducción de inventario. Con el fin de lograr mayor productividad y debido a la importancia que posee el manejo de la información en las empresas, se utilizan las herramientas propias de los sistemas de gestión empresarial, que permiten controlar, planificar, organizar y dirigir cada uno de los eslabones de la cadena productiva” (3).

A continuación se seleccionan algunos Sistemas de Control de Almacenes y Ventas a nivel mundial para conocer sus características:

1.3 Sistemas de Control de Almacenes y Ventas a nivel mundial

Muchas empresas en el mundo poseen Sistemas de Control de Almacenes y Ventas, facilitando el manejo de toda la información relacionada. Estos sistemas se adaptan a diversos tipos de establecimientos y facilitan la realización de los procesos de las empresas, entre los sistemas más usados a nivel mundial se encuentran:

1.3.1 Trazabilidad 3000

“Trazabilidad 3000 es un sistema desarrollado por Cea Ordenadores, una empresa creada en 1984 que se dedica a la creación de software en español. Este software facilita la tarea de

controlar en un almacén las entradas y salidas de productos, la facturación y los recibos, los datos de los clientes, los pedidos, proveedores, transportistas e inventarios” (4).

1.3.2 Secop PRO

“Es un pequeño, sencillo, práctico y funcional programa diseñado para llevar un adecuado control del inventario. Incorpora funciones para el registro de productos, ventas y compras, impresión de diversos reportes, posibilidad de realizar copias de seguridad y personalizar algunos aspectos operativos” (5).

1.3.3 Gestión 3000

“Creado por la empresa Cea Ordenadores. Gestión 3000 es una herramienta de gestión de facturación que abarca campos como: compras, ventas, control de almacenes, TPV y contabilidad. El programa cuenta con diversos módulos por separado, entre ellos: facturas, gestión de almacén, control de clientes, proveedores y control de ventas. Mediante estos módulos un personal calificado o una pequeña empresa pueden controlar todos los aspectos de la facturación y contabilidad del negocio” (6).

1.3.4 ¿Por qué no se utilizan estos sistemas?

Estos sistemas son altamente funcionales a la hora de cumplir sus objetivos específicos, pero al utilizarlos en la UERMP no sería conveniente porque poseen funcionalidades que no se necesitan. Además, no se pueden ejecutar en el sistema operativo Linux, porque son sistemas propietarios, es decir, hay que pagar licencia para hacer uso de estos sistemas.

1.4 Sistemas de Control de Almacenes y Ventas en Cuba

En Cuba también se encuentran sistemas que controlan los medios en los almacenes y las compras-ventas realizadas en las empresas; entre ellos se pueden mencionar:

1.4.1 Rodas XXI Versión 3.0

“El potente y versátil Sistema Integral Económico- Administrativo Rodas XXI ha sido creado por CITMATEL, para la automatización de la gestión empresarial. Permite aprovechar al máximo las posibilidades que le brinda el más moderno equipamiento informático. Contiene diferentes módulos que pueden usarse integrados o independientes entre ellos: inventarios,

contabilidad, facturación y finanzas. Este sistema controla los movimientos de todos los productos; lleva el historial de su ciclo de vida. Se pueden realizar todo tipo de operaciones de entradas y salidas de productos de los almacenes con facilidad en el momento que se desee, generando el documento asociado al movimiento de forma automática. Es posible trabajar con varios almacenes y a su vez cada uno de ellos de forma independiente” (7).

1.4.2 Siscont5

El Software Contable Financiero (Siscont), es un sistema que se adhiere a las definiciones y conceptos del Ministerio de la Industria Básica, aunque, por las acciones contables y financieras que permite, puede ser utilizado en otras entidades nacionales. Contiene varios módulos, entre ellos: inventarios, ventas y pedidos, facturación y activos fijos. Puede ser explotado en régimen monousuario y multiusuario. “El sistema está preparado para controlar el saldo de cada material en dos monedas, a partir de los procedimientos vigentes en el país en cuanto a política monetaria. Permite el tratamiento de la contabilidad de forma transaccional, por resúmenes diarios o de forma personalizada según se defina por parámetros. Incorpora tratamiento de lotes a los productos y dos tipos de valoración de las cuentas *FIFO (First In First Out)* y Promedio Ponderado” (8).

1.4.3 EI SABIC

“El Sistema Automatizado para la Banca Internacional de Comercio (SABIC), es un sistema diseñado y desarrollado por la Dirección de Sistemas Automatizados del Banco Central de Cuba, para satisfacer las necesidades de procesamiento de datos de bancos e instituciones no bancarias, utilizando los medios técnicos de computación disponibles en el mercado. Este sistema ha sido adaptado a los requerimientos de las operaciones propias del Banco Central y ha sido desarrollado para que los empleados que hagan uso de él puedan tramitar sus operaciones y realizar sus consultas sin necesidad de acudir a los archivos ni a la actividad manual. De esta forma se aumenta la seguridad, eficiencia del trabajo y la productividad de los trabajadores.

El SABIC permite la contabilidad en tiempo real (mantiene actualizados los ficheros contables) y la contabilización multimoneda (permite registrar los activos y pasivos en las monedas orígenes, sin tener que realizar en el momento del registro las conversiones de

monedas, lo cual aumenta la exactitud de la información sobre la posición financiera de la institución). Además, las operaciones contables se pueden realizar a través de transacciones tipificadas que generan los asientos contables de forma automática. Teniendo en cuenta lo dinámico que resulta la actividad bancaria, este sistema está concebido modularmente: un módulo central, módulos de transacciones, listados y procesos. Esta modularidad tiene la intención de facilitar la adaptabilidad y el mantenimiento del sistema para garantizar su evolución, es decir, se puede modificar cualquier módulo sin que los demás se vean afectados” (9).

1.4.4 ¿Por qué no se utilizan estos sistemas?

A pesar de que algunas empresas cubanas utilizan varios sistemas nacionales y extranjeros, para el control y la gestión de los recursos materiales, además, realizan procesos de ajuste de inventario, baja de los medios materiales e inventario físico, se determina que estos sistemas no son la mejor opción para el sistema propuesto; ya que no se ajustan a las normativas de la UERMP, pues han sido creados para resolver los problemas de determinadas entidades.

1.5 Sistemas de Control de Almacenes y Ventas en la Universidad de Ciencias Informáticas

En la Universidad de Ciencias Informáticas (UCI) también se utilizan sistemas de Control de Almacenes y Ventas. Entre ellos se encuentra Asset NS.

1.5.1 Asset NS

Sistema de Gestión Integral (Asset NS), es un sistema integral modular concebido para el control de la actividad económica empresarial. Permite realizar, controlar y contabilizar todas las transacciones relacionadas con el proceso de compra – venta de productos y servicios, los cobros, pagos y anticipos asociados a los mismos, recursos humanos, nóminas, los activos fijos y las herramientas de la entidad. Es un sistema que facilita el uso de la parametrización para adaptarse a las exigencias de cada cliente, en particular en la emisión de varios reportes que tendrán la forma y el contenido que el usuario les defina. Este sistema posibilita el control de inventarios de múltiples almacenes y generación automática de comprobantes de operaciones. Ofrece un control estricto de las existencias, reservas y disponibilidad de los

productos, así, como las cuentas por cobrar y pagar debidas a la facturación y recepción de los productos.

1.5.2 ¿Por qué no se utiliza este sistema?

Después de realizar un estudio del sistema, se llega a la conclusión de que no cumple con todas las funcionalidades requeridas por el cliente, por lo que se hace necesaria una nueva aplicación que cumpla con las necesidades de la UERMP.

1.6 Metodología 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 es el paso inicial para el desarrollo de un sistema. Además de una notación, de un proceso y de herramientas, las metodologías proporcionan guías para estimar costos, manejo del proyecto en las tareas y entregas, así, como medidas y métricas, formas definidas y dirección en las entregas de la construcción, políticas y procedimientos para garantizar la calidad del software y descripciones de los roles.” (10).

Actualmente se cuenta con una buena cantidad de metodologías tradicionales o robustas como RUP y metodologías ágiles como Programación Extrema (XP), Crystal Methodologies y Scrum , pero sólo se analizarán las más conocidas, para así poder elegir la más adecuada para realizar el análisis y diseño del sistema.

1.6.1 Crystal Methodologies

“Se trata de un conjunto de metodologías para el desarrollo de un software, caracterizadas por estar centradas en las personas que componen el equipo y la reducción al máximo del número de artefactos producidos. Han sido desarrolladas por Alistair Cockburn. En esta metodología, el desarrollo del software se considera un juego cooperativo de invención y comunicación, limitado por los recursos a utilizar; confía en la autorregulación y libera el proceso de codificación, además, las revisiones se realizan al final de cada iteración. De sus principales características se puede decir, que cuando el número de personas del proyecto aumenta, también aumenta la necesidad de coordinar; el factor más significativo es la comunicación

entre el cliente y los que están desarrollando el sistema, además, cuando el potencial de daño se incrementa, la tolerancia a variaciones se ve afectada” (11). De sus desventajas se puede decir que “Crystal Methodologies como todas las metodologías ágiles están pensadas para proyectos pequeños y reducidos de tiempo, presenta requisitos volátiles o basados en nuevas tecnologías y hace menos énfasis en la arquitectura del software” (12).

1.6.2 Programación Extrema

XP (de sus siglas en inglés *Extreme Programming*), es una de las metodologías más exitosas en la actualidad, utilizada para proyectos de corto plazo y equipo pequeño. La metodología consiste en una programación rápida o extrema cuya particularidad es simplificar el desarrollo del software y reducir el costo del proyecto. Los principios básicos de la Programación Extrema son: la retroalimentación rápida, asumir la simplicidad, el cambio incremental, adherirse al cambio y trabajo de alta calidad (desde trabajo excelente hasta trabajo increíblemente sobresaliente). Aunque en XP existe una comunicación y retroalimentación muy profunda entre el cliente y los desarrolladores del equipo y se hace más simple el desarrollo y codificación de los módulos del sistema, presenta desventajas, entre las que se encuentran que: genera una documentación pobre, en muchos casos el cliente solicita que además del producto de software sea entregado un manual de usuario y esta metodología no define como generarla; como se realizan constantes pruebas de unidad, a medida que avanza el proyecto se van haciendo más complejas, lo que podría hacer que se retrase; además, al ser el código propiedad de todos y tener el programador libertad de realizar cualquier cambio, aumenta la posibilidad de colisión entre los programadores.

1.6.3 Proceso unificado de desarrollo de software

RUP (de sus siglas en inglés *Rational Unified Process*), es un proceso de amplio marco que ofrece las mejores prácticas para el software y los sistemas de entrega y la gestión de proyectos. Constituye una de las metodologías más utilizadas para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es una metodología de desarrollo de software basada en casos de uso para describir lo que se espera del software y está orientada a la arquitectura del sistema, documentándose lo mejor posible y basándose en UML como lenguaje de modelado. Posee 3 características principales como son: **dirigido por casos de**

uso: los casos de uso capturan requerimientos funcionales y representan piezas de funcionalidad, que brindan un resultado de valor al usuario; **centrado en una arquitectura:** comprende los aspectos estáticos y dinámicos más importantes del sistema; **iterativo e incremental:** el trabajo se divide en piezas pequeñas o mini proyecto, cada uno proveyendo un subproducto incremental. Además, posee grandes ventajas como: la reducción de riesgos basado en la retroalimentación temprana; las pruebas continuas e iterativas promueven una mejor evaluación del estado del proyecto, los patrocinadores reciben evidencia concreta del avance del proyecto, además, se pueden acomodar mejor los cambios (requerimientos, tácticos y tecnológicos) y los problemas más complejos se atacan primero. En cuanto a sus desventajas, si se quiere aplicar RUP a un proyecto pequeño, sería tedioso realizar todas las fases en un período corto de tiempo, no se podría aplicar mucho lo de iteratividad en cada fase y por el grado de complejidad puede no resultar muy adecuado su uso.

1.6.4 Selección de la metodología a utilizar

Lo más importante para elegir una metodología para el desarrollo de un software, es determinar el alcance que tendrá, saber como es el proyecto que se va a desarrollar y cual es el tiempo con que se dispone. A partir del estudio realizado de las metodologías más usadas se decide trabajar con RUP. La misma fue escogida por ser una metodología robusta y bien definida, probada en diferentes proyectos en la UCI; realiza un levantamiento exhaustivo de requerimientos, determina cuales son los defectos del producto desde su fase inicial; realiza un análisis y diseño lo más completo posible; se tiene en cuenta la opinión del cliente, además, permite guiar las actividades del proyecto y contar con la documentación adecuada a lo largo del desarrollo del software. Se considera que RUP es la óptima para modelar el sistema debido a que representa grandes potencialidades para el análisis, implementación y documentación de sistemas orientados a objetos; 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.

1.7 Lenguaje de modelado

1.7.1 UML

UML (de sus siglas en inglés *Unified Modeling Language*), es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir. Se usa para entender, diseñar, hojear, configurar, mantener y controlar la información sobre tales sistemas. Se caracteriza por presentar tecnología orientada a objetos, viabilidad en la corrección de errores, así, como un desarrollo iterativo e incremental, participación del cliente en todas las etapas del proyecto; permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos. Puede conectarse con lenguajes de programación (ingeniería directa e inversa).

En este lenguaje existe un equilibrio entre expresividad y simplicidad, por lo que no es difícil de aprender y de utilizar. Cubre con las cuestiones relacionadas con el tamaño propio de los sistemas complejos y críticos, además, es un lenguaje muy expresivo que cubre todas las vistas necesarias para desarrollar y luego desplegar los sistemas. Es importante resaltar que UML es un "lenguaje" para especificar y no para describir métodos o procesos. "UML en otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en una gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso unificado de desarrollo de software), pero no especifica en sí mismo que metodología o proceso usar" (13).

Se utiliza UML como lenguaje de modelo, porque es el lenguaje que utiliza la metodología RUP para la realización del modelado de sus procesos.

1.8 Leguajes y tecnologías del lado del cliente

1.8.1 XML

XML (de sus siglas en inglés *Extensible Markup Language*), es un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium*. XML es una simplificación y adaptación del Lenguaje de Marcado Generalizado (SGML) y permite definir la gramática de lenguajes específicos. XML es realmente una manera de definir lenguajes para diferentes

necesidades; propone un estándar de datos que puede codificar el contenido, la semántica y los esquemas de una gran variedad de casos, desde los más simples a los más complejos. “Entre sus ventajas se encuentran que es extensible: después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se puede continuar utilizando sin complicación alguna; el analizador es un componente estándar, no es necesario crear un analizador específico para cada versión de lenguaje XML, esto posibilita el empleo de cualquiera de los analizadores disponibles, de esta manera, se evitan bugs (errores) y se acelera el desarrollo de aplicaciones; si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla; además, XML mejora la compatibilidad entre aplicaciones. Entre sus desventajas se puede decir que el navegador o el “visualizador” que el usuario utiliza, no cuenta con un *Parser* capaz de visualizar el contenido del documento” (14).

1.8.2 HTML

“HTML (de sus siglas en inglés *Hypertext Markup Language*), es el lenguaje utilizado para representar documentos en la WWW (*World Wide Web*). Además de texto normal incluye elementos multimedia (gráficos, vídeo, audio) y existen links (enlaces) que permiten saltar a otras partes del documento o a otro sitio cualquiera de Internet. La descripción se basa en especificar en el texto la estructura lógica del contenido (títulos, párrafos de texto normal, enumeraciones, definiciones, citas.), así, como los diferentes efectos que se quieren dar (cursiva, negrita, o un gráfico determinado) y dejar que luego la presentación final de dicho hipertexto se realice por un programa especializado” (15). Otra característica muy importante de este lenguaje es que es portable, es decir, se pueden visualizar las páginas con cualquier sistema operativo y por supuesto también crearlas. Entre sus desventajas se encuentran que “en HTML existe un determinado número de etiquetas, por lo que no se pueden crear más; es muy limitado, ya que las opciones para realizar la estructura no son muy numerosas y no se sabe con seguridad como se va a visualizar el documento” (16).

1.8.3 CSS

“CSS (de sus siglas en inglés *Cascading Style Sheets*), es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con

HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación, además, es imprescindible para crear páginas web complejas. Dicho lenguaje se utiliza para definir el aspecto de todos los contenidos, es decir, el color, tamaño y tipo de letra de los párrafos de texto, la separación entre titulares y párrafos, así, como la tabulación con la que se muestran los elementos de una lista” (17). De sus ventajas se puede decir que: obliga a crear documentos HTML y XHTML bien definidos y con significado completo; mejora la accesibilidad del documento; reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes. Entre sus desventajas, se puede decir que: es incompatible entre diferentes navegadores, por lo que si algo se ve excelente en un navegador es posible que en otro no se vea igual, o que varíe la visualización incluso en las diferentes versiones de un mismo navegador.

1.8.4 JavaScript.

JavaScript es un lenguaje de scripts desarrollado por Netscape para incrementar las funcionalidades del lenguaje HTML. Es un lenguaje interpretado ya que no necesita de ningún compilador, solamente de un navegador y es orientado a objetos. JavaScript es sumamente utilizado en el mundo del desarrollo web por ser muy versátil y potente, tanto para la realización de pequeñas tareas como para la gestión de complejas aplicaciones. Con su utilización se puede evitar un poco la sobrecarga del lado del servidor, ya que funciones como la validación de formularios (que antes se hacían del lado del servidor), se pueden realizar con este lenguaje y de una manera rápida. De sus ventajas se puede decir que: no requiere tiempo de compilación, los scripts pueden desarrollarse en un período de tiempo relativamente corto; posee características de interfaz, que son gestionadas por el navegador y por el código HTML; los programas JavaScript pueden ser pequeños y compactos, no requieren mucha memoria ni tiempo adicional de transmisión; es independiente de la plataforma de hardware o sistema operativo, siempre y cuando exista un navegador con soporte JavaScript. De sus desventajas se puede decir que: en JavaScript el número de métodos integrados es insuficiente para gestionar documentos y ventanas, además, el código del script debe descargarse completamente antes de poderse ejecutar y no es posible ocultar el código fuente.

1.8.5 Selección del lenguaje del lado del cliente

Con las características antes expuestas, se seleccionan como lenguajes del lado del cliente: HTML y JavaScript. HTML puede utilizar algunas funcionalidades de JavaScript, para la realización de una aplicación con la calidad requerida y para validar que la entrada de los datos sea correcta. Se propone utilizar CSS, pues es la mejor forma de separar el contenido y su presentación, además, es imprescindible para crear páginas web.

1.9 Lenguajes del lado del servidor

1.9.1 ASP

ASP (de sus siglas en inglés *Active Server Pages*). ASP es un lenguaje más lento y pesado que PHP. Algunas de las ventajas de dicho lenguaje consisten en que debido a que usa principalmente VBScript, es relativamente simple tratar con el lenguaje si ya se conoce como programar en Visual Basic. El soporte de este lenguaje también se encuentra habilitado por defecto en el servidor IIS, facilitando su instalación y ejecución. De desventajas se puede decir que: “los componentes integrados en ASP son bastante limitados, de modo que si necesita usar características "avanzadas", como interactuar con servidores *FTP*, necesita comprar componentes adicionales; el mayor inconveniente es que se trata de un sistema propietario que es usado nativamente solo por IIS (*Microsoft Internet Information Server*), poca flexibilidad y escasa seguridad que estos equipos llevan” (18).

1.9.2 JSP

JSP (de sus siglas en inglés *Java Server Pages*), está basado en Java, orientado a objetos y multiplataforma. Desarrollado por Sun Microsystems Inc., con gran experiencia y actualizaciones habituales y con un proyecto de desarrollo libre a partir de la versión 1.2. JSP hace gran énfasis en los componentes y no tanto en los *scripts*, permitiendo obtener un buen rendimiento y escalabilidad. Es necesario que disponga de, por ejemplo, *TomCat* para poder ejecutarlo. Dos de las grandes desventajas de JSP es: por una parte, una excesiva complejidad y por otra (paradójicamente) una aproximación elemental; la excesiva complejidad deriva de enfocar cualquier problema como un EJB (*Enterprise Java Bean*), lo que muchas veces lleva a consumir grandes cantidades de tiempo y dinero; la aproximación

elemental radica en que muchas empresas usan JSP como ASP, como una manera de hacer sus páginas dinámicas pero sin integrarlas a todos los módulos *Java*.

1.9.3 PHP

PHP (de sus siglas en inglés *Hypertext Processor*), es un lenguaje sencillo, de sintaxis cómoda y similar a la de otros lenguajes como *C* o *C++*, es multiplataforma, orientado a objetos y dispone de una gran cantidad de librerías que facilitan muchísimo el desarrollo de las aplicaciones. Es un lenguaje ideal tanto para el que comienza a desarrollar aplicaciones web como para el desarrollador experimentado y está alcanzando niveles de uso que convierten su conocimiento en algo indispensable para los profesionales del desarrollo en internet. PHP es un lenguaje basado en herramientas con licencia de software libre, es decir, no hay que pagar licencias, no está limitado en su distribución y se puede ampliar con nuevas funcionalidades si así se quiere. Es capaz de combinarse con gestores de bases de datos tales como: MySQL, PostgreSQL y Oracle, lo que permite la creación de una aplicación web robusta que arroja resultados muy interesantes y prometedores para aquellas páginas que deseen ser activas y dinámicas. La ejecución e interpretación de PHP es completamente en el servidor web: en este se encuentra almacenado el script por lo que el cliente solo puede recibir el resultado de la ejecución.

PHP puede ser capaz de interactuar con servidores web muy populares como ISAPI y Apache por existir también en versión CGI (*Common Gateway Interface*). El uso de PHP presenta ventajas como: es capaz de leer y manipular datos desde diversas fuentes, incluyendo datos que pueden ingresar los usuarios desde formularios HTML; cuenta con una biblioteca sumamente amplia por defecto de funciones; no requiere definición de tipos de variables ni manejo detallado del bajo nivel. De sus desventajas se puede decir que: “PHP hace que sea muy difícil mantener el código de terceras personas, además, existen muchas versiones de PHP con incompatibilidades entre sí, y en PHP 4 es difícil depurar los errores” (19).

1.9.4 Selección del lenguaje del lado del servidor

Después del estudio realizado de los diferentes lenguajes de programación más usados, se determina emplear PHP para el desarrollo de la aplicación debido a que PHP es *OpenSource*, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas

que no funcionan, además, no está forzado a pagar actualizaciones anuales para tener una versión que funcione; es un lenguaje que corre en cualquier plataforma utilizando el mismo código fuente; PHP actualmente se puede ejecutar bajo Apache, IIS, además, puede conectarse con muchos gestores de base de datos tales como: MySQL, PostgreSQL y Oracle, destacando su conectividad con MySQL; es capaz de leer y manipular datos desde diversas fuentes, incluyendo los datos que pueden ingresar los usuarios desde formularios HTML; tiene la capacidad de expandir su potencial utilizando su enorme cantidad de módulos, además, posee una amplia documentación en su web oficial de Internet.

1.10 Sistemas gestores de base de datos

Los sistemas gestores de base de datos permiten almacenar y posteriormente acceder a los datos de forma rápida y estructurada. Entre los más utilizados están: PostgreSQL, MySQL y Oracle.

1.10.1 MySQL

“MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Funciona en diferentes plataformas. Posee un sistema de privilegios y contraseñas que es muy flexible, seguro y permite verificación basada en el *host*. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor. MySQL soporta a grandes bases de datos. Un índice puede usar prefijos de una columna para los tipos de columna *CHAR*, *VARCHAR*, *BLOB*, o *TEXT*. MySQL Server tiene soporte para comandos SQL, que permite chequear, optimizar y reparar tablas” (20). Entre sus ventajas se encuentran: su velocidad a la hora de realizar las operaciones, que hace a MySQL uno de los gestores que ofrecen mayor rendimiento; su bajo consumo lo hace apto para ser ejecutado en una máquina con escasos recursos sin ningún problema; las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación; tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está; el conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet, en servicios de foro y de buscadores de aplicaciones. Entre sus desventajas se puede decir que: MySQL carece de soporte para transacciones,

rollback y subconsultas, el hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo, para aquellos programadores que provienen de otros gestores que sí poseen esta característica; MySQL no es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.

1.10.2 PostgreSQL

PostgreSQL es un sistema de gestión de base de datos objeto-relacional (*ORDBMS*), es una derivación libre (*OpenSource*) del proyecto POSTGRES. Esta versión incluye características de la orientación a objetos, como puede ser herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, pero a pesar de esto PostgreSQL no es un sistema de gestión de base de datos puramente orientado a objetos. Una de sus principales características es que soporta distintos tipos de datos, además de los del tipo base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes, cadenas de *bits* y permite la creación de tipos propios (21).

Entre sus ventajas se encuentran: la escalabilidad; es capaz de ajustarse al número de CPU (Unidades Centrales de Procesamiento) y a la cantidad de memoria que posee el sistema de forma óptima, siendo capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta; implementa el uso de *rollback*, subconsultas y transacciones; tiene la capacidad de comprobar la integridad referencial, así, como almacenar procedimientos en la propia base de datos; el único costo asociado a él, es el de conocerlo, pues su código fuente está disponible bajo la más liberal de las licencias del *OpenSource*: la licencia *BSD*, que permite usarlo, modificarlo y distribuirlo en productos comerciales o no comerciales, sin costo alguno. Entre sus desventajas se puede decir que: es lento y tiene un límite de 8K por fila, aunque, se puede aumentar a 32K, con una disminución considerable del rendimiento.

1.11.3 Oracle

“Oracle es un sistema gestor de base de datos relacional, que se destaca por ofrecer soporte de transacciones, escalabilidad, además de ser multiplataforma. Por estas características se le considera como uno de los más completos. Las últimas versiones han sido certificadas para poder trabajar sobre plataforma Linux” (22). Entre sus principales ventajas se pueden

mencionar que: posee igual interacción en todas las plataformas (Windows, Unix, Macintosh y Mainframes); más del 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas operativos; Oracle soporta base de datos de todos los tamaños, desde severas cantidades de *bytes* y *gigabytes* en tamaño; tiene soporte de transacciones, la realización de *backups* (copias de seguridad) es fácil, es eficaz y eficiente además, tiene una amplia gama de herramientas para operar con la base de datos tanto como usuario que como administrador. Su principal desventaja es que es un software privativo por lo que hay que pagar un enorme precio (que es de varios miles de euros), para utilizarlo; además de criticársele la política de suministro de *parches* de seguridad desde el 2005, lo cual ha incrementado el nivel de exposición de los usuarios en lo que a seguridad respecta.

1.10.4 Selección del gestor de base de datos a utilizar

Después del estudio realizado de los diferentes gestores de base de datos más usados, se determina utilizar PostgreSQL, porque aunque, Oracle es el mejor gestor de base de datos en la actualidad, es un software propietario y con un costo muy elevado. Por el contrario PostgreSQL es libre, es un gestor de base de datos objeto-relacional altamente extensible, soporta operadores y tipos de datos definidos por el usuario; es capaz de manejar complejas rutinas y reglas de modo que su avanzada funcionalidad se pone de manifiesto con las consultas SQL declarativas. Además, tiene ciertas características orientadas a objetos, puede manejar un ilimitado número de base de datos y permite una fácil gestión de los usuarios.

1.11 Herramientas de modelado

Herramienta Case

Las Herramientas de Ingeniería de Software Asistida por Ordenador (CASE) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software, reduciendo el costo de las mismas en términos de tiempo y dinero. Estas herramientas ayudan en todos los aspectos del ciclo de vida de desarrollo del software, en tareas como: el proceso de realizar un diseño del proyecto, cálculo de costos, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación y detección de errores.

1.11.1 Visual Paradigm

Es un software privativo, para modelado en UML. Esta herramienta tiene características gráficas muy cómodas, que facilitan la realización de los diagramas de modelado que sigue el estándar de UML; los mismos son: diagramas de clases, casos de uso, comunicación, secuencia, estado, actividad, componentes. Entre sus principales ventajas se encuentran: que presenta entorno de creación de diagramas para UML 2.0; su diseño es centrado en casos de uso y enfocado al negocio, lo cual genera un software de mayor calidad; posee uso de un lenguaje estándar común a todo el equipo de desarrollo, que facilita la comunicación, las capacidades de ingeniería directa e inversa, modelo y código, que permanece sincronizado en todo el ciclo de desarrollo; Visual Paradigm *está* disponible para múltiples plataformas; soporta aplicaciones web, varios idiomas, es fácil de instalar y actualizar y presenta compatibilidad entre ediciones.

“Visual Paradigm es una herramienta CASE profesional que soporta el ciclo de vida del desarrollo de un software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue; UML ayuda a una rápida construcción de mejores aplicaciones y a un menor costo; permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación, además, incorpora el soporte para trabajo en equipo, lo cual permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios realizados por sus compañeros. Entre sus desventajas se puede decir que: Visual Paradigm no permite su uso en proyectos comerciales y muestra muchas otras funcionalidades no disponibles como *gancho* para las versiones de pago” (23).

1.11.2 Rational Rose

Es una de las más poderosas herramientas de modelado visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo. Cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, la construcción de los componentes, la transición a los usuarios y la certificación de las distintas fases. La interfaz de Rational Rose está formada por un navegador, que permite acceder a las distintas vistas del modelo, una ventana de documentación que posibilita

manejar los documentos del elemento seleccionado en cualquiera de los diagramas, una barra de herramientas *Standard* para acceder rápidamente a las acciones comunes a ejecutar para cada uno de los diagramas del modelo, una barra de herramientas Diagrama en la que muestra el conjunto de herramientas disponibles para el diagrama activo, una ventana de Diagrama la cual permite desplegar y editar cualquiera de los diagramas UML y una ventana Registro que permite registrar todas las órdenes ejecutadas y errores que se producen durante la ejecución. “De sus desventajas se puede decir que su entorno gráfico no es muy amigable para el usuario, no es un software libre y además, no se puede crear el entorno del sistema para los diagramas de casos de uso” (24).

1.11.3 Selección de la herramienta de modelado a utilizar

Se propone utilizar la herramienta CASE Visual Paradigm porque además de cumplir perfectamente con todos los estándares UML, es una herramienta que la UCI ha pagado por su licencia y tiene como política que sea la herramienta CASE a utilizar.

1.12 Servidor Web.

“Un servidor web brinda contenido estático a un navegador, carga un archivo y lo envía a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante HTTP” (25).

1.12.1 IIS

IIS (de sus siglas en inglés *Internet Information Server*), es una serie de servicios para los ordenadores que funcionan con Windows. Los servicios que ofrece son: FTP, SMTP y HTTP/HTTPS. Estos servicios convierten a un ordenador en un servidor de Internet o Intranet, es decir, que en las computadoras que tienen estos servicios instalados se pueden publicar páginas web, tanto locales como remotas. “El servidor web se basa en varios módulos que le dan capacidad para procesar distintos tipos de páginas, por ejemplo, Microsoft incluye los de *Active Server Pages (ASP)* y *ASP.NET*; también, pueden ser incluidos los de otros fabricantes, como *PHP*. Sus principales desventajas están en que: este servidor solo funciona con el sistema operativo Windows, no es un servidor web flexible, debido a las limitaciones de Microsoft; el código fuente es propietario; es difícil controlar como se analizan las URL;

además, si se utilizan dos servidores, las configuraciones de los servidores deben coincidir exactamente” (26).

1.12.2 Thttpd

Thttpd es un servidor web de código libre, disponible para la mayoría de las variantes de Unix. Se caracteriza por ser simple, pequeño, portátil, rápido y seguro, ya que utiliza los requerimientos mínimos de un servidor HTTP. Esto lo hace ideal para servir grandes volúmenes de información estática. Entre sus ventajas se puede decir que: es simple, ya que sólo maneja lo necesario para poner en práctica el protocolo HTTP; es pequeño, porque también tiene un pequeño tamaño de período de explotación y es muy cuidadoso sobre la asignación de memoria; es portátil, pues se compila limpiamente sobre la mayoría de sistemas operativos; es rápido, porque en el empleo típico, es sobre todo más rápido que los mejores servidores "destacados" (Apache); es seguro, porque se extiende a grandes longitudes para proteger el servidor web contra ataques de otros sitios, el administrador puede decidir restringir la transferencia de archivos de imagen JPEG a en la mayor parte de 20 *KB* por segundo; los promedios de carga se caen debido a la reducción de la transferencia gráfica. “Entre sus desventajas, se puede decir que: Thttpd no está optimizado para trabajar con varios threads (núcleos), además, no presenta un buen soporte de tecnologías” (27)

1.12.4 Apache

Apache es un servidor de páginas web de código abierto, multiplataforma y modular, se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation. Sus mensajes de errores son altamente configurables; se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular, este diseño permite a los administradores de sitios web, elegir que características van a ser incluidas en el servidor seleccionando que módulos se van a cargar, ya sea al compilar o al ejecutar el servidor. Sus principales ventajas están en que: trabaja sobre múltiples plataformas; incluye módulos que se cargan de forma dinámica; brinda soporte para base de datos, además de poseer soporte SSL para transacciones seguras; incluye soporte para *hosts* virtuales; es un servidor de código abierto, además de ser rápido y eficiente. “Su principal desventaja es que debido a la cantidad de módulos y funcionalidades que presenta es más lento que otros servidores” (28).

1.12.5 Selección del servidor web

Se escoge el servidor web Apache porque es una de las plataformas de servidores web de código abierto, más poderosa del mundo. Es una tecnología gratuita, multiplataforma, altamente configurable y muy sencilla de utilizar e implementar, por lo que cualquier persona con experiencia en programación puede implementar un módulo para Apache, con el objetivo de realizar una función determinada, esto significa que hay una gran cantidad de módulos disponibles para su utilización, con disímiles funcionalidades. Apache, presenta base de datos de autenticación, reescritura de las URL, comprobación de la ortografía de las URL y gran cantidad de manuales *on-line*.

Después de un estudio y selección de las metodologías, los lenguajes y tecnologías que se utilizarán para la realización del diseño del sistema, se proponen las herramientas con las que el sistema se debe desarrollar.

1.13 Herramientas a utilizar

1.13.1 NetBeans IDE

Se escoge trabajar con NetBeans IDE que es una aplicación de código abierto ("*OpenSource*"), diseñada para el desarrollo de aplicaciones fácilmente portables entre las distintas plataformas, haciendo uso de la tecnología Java. NetBeans IDE dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, colaboración entre varias personas, creación de aplicaciones compatibles con teléfonos móviles, resaltado de sintaxis y por si fuera poco sus funcionalidades son ampliables mediante la instalación de *packs*. NetBeans IDE soporta el lenguaje PHP 5.3 y el marco de trabajo Symfony. Las desventajas de usar la plataforma NetBeans es que no tiene compatibilidad integrada para puntales y etiquetas de *struts*; además de tener un rendimiento más lento que Eclipse.

1.13.2 DBDesigner 4

DBDesigner es un sistema totalmente visual de diseño de base de datos, que combina características y funciones profesionales con un diseño simple, muy claro y fácil de usar, a fin

de ofrecer un método efectivo para gestionar la base de datos. Permite administrar la base de datos, diseñar tablas, hacer peticiones SQL manuales; presenta soporte para la función *drag-and-drop*. El programa dispone además de una interfaz profesional y de detallados manuales de uso. DBDesigner 4 es un sistema que integra diseño de base de datos, modelado, creación y mantenimiento en un entorno único, sin fisuras. Combina características profesionales y una interfaz de usuario clara y sencilla para ofrecer la manera más eficiente para gestionar la base de datos.

1.13.3 EMS PostgreSQL Manager 3

EMS SQL Manager para PostgreSQL es una poderosa herramienta gráfica para la administración y desarrollo de PostgreSQL Database Server (servidor de base de datos PostgreSQL). PostgreSQL Manager soporta todas las nuevas características de PostgreSQL incluyendo *tablespaces* (espacios de tablas), *named arguments* (argumentos nombrados) en funciones. Ofrece una gran variedad de herramientas para usuarios avanzados, tales como: *Visual Database Designer* (diseñador visual de base de datos), *Visual Query Builder* (constructor visual de consultas) y un poderoso editor de objetos binarios (*BLOB*), para satisfacer todas sus necesidades. PostgreSQL Manager cuenta con una nueva y avanzada interfaz gráfica de usuario con un sistema asistente bastante descriptivo, tan claro en su uso que ni un principiante se podrá confundir. Entre sus características principales se encuentran: que presenta una nueva interfaz gráfica de usuario último modelo, presenta ágil navegación y administración de base de datos, además de tener una administración sencilla de todos los objetos PostgreSQL, por lo que se propone como herramienta a utilizar en el desarrollo del sistema.

1.14 Marco de Trabajo

En el desarrollo de software, un marco de trabajo es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. “Típicamente, un marco de trabajo puede incluir: soporte de programas, bibliotecas y un lenguaje interpretado, para ayudar a desarrollar y unir los diferentes componentes de un proyecto” (29).

Un marco de trabajo, simplifica el desarrollo de una aplicación mediante la automatización de algunos de los patrones utilizados, para resolver las tareas comunes. Además, un marco de

trabajo proporciona estructura al código fuente, donde el desarrollador es obligado a crear código más legible y más fácil de mantener, lo que permite la programación de aplicaciones, ya que encapsula operaciones complejas en instrucciones sencillas.

1.14.1 Symfony 1.4

Se propone para la realización del sistema, utilizar Symfony, que es un marco de trabajo que tiene como objetivo fundamental, automatizar los patrones más utilizados en la elaboración de sistemas web; además, obliga a los programadores a clarificar el código; establece un estándar de código legible; encapsula operaciones complejas en simples líneas de código, ahorrando mucho más tiempo a la hora de mostrar datos directamente de la base de datos. Symfony es uno de los marcos de trabajos más completos para el lenguaje PHP, con la implementación de ORM, patrones de diseño, herramientas de pruebas, conceptos de Modelo/Vista/Controlador y totalmente orientado a objetos. “Entre sus características se encuentran que es: **escalable**: Symfony es infinitamente escalable si se disponen de los recursos necesarios, además, es posible complementarlo con un centenar de *plugin* desarrollados por la comunidad; **probado y testeado**: Symfony ha sido probado con éxito durante varios años en aplicaciones muy diferentes; **documentado**: se trata del marco de trabajo desarrollado con PHP mejor documentado” (30).

Conclusiones

Con el desarrollo del capítulo se realizó una amplia investigación sobre el estado de los Sistemas de Control de Almacenes y Ventas de productos, así, como los conceptos fundamentales para el desarrollo del presente trabajo, tanto en el ámbito nacional como internacional, llegando a la conclusión de que ninguno puede utilizarse en la UERMP, pues las funcionalidades que poseen son propias de los sistemas para los cuales fueron desarrollados. Además, se hizo un análisis profundo de todas las tecnologías, lenguajes, metodologías y herramientas a utilizar para el análisis y diseño del sistema. Para poder realizar estas tareas se tuvieron en cuenta algunos métodos de la investigación como: el histórico-lógico y el analítico-sintético. Después de realizado el análisis se llega a la conclusión que se utilizará como lenguaje para el desarrollo del sistema: PHP; como lenguaje de modelado: UML; como sistema gestor de base de datos: PostgreSQL; como metodología de desarrollo: RUP; como herramienta de modelado: Visual Paradigm y como marco de trabajo Symfony 1.4.

Capítulo 2: Características del Sistema

2. Introducción

En el presente capítulo se abordan las características que el sistema debe tener, se describe el modelo del negocio, así, como se especifica como se realizó el levantamiento de requisitos tanto funcionales como no funcionales. Se realiza el diagrama de casos de uso del sistema; así, como la expansión de los casos de uso del sistema. Además, se realiza el estudio de la factibilidad de la propuesta.

2.1 Descripción del negocio

2.1.1 Modelo de negocio

El Modelamiento del negocio en la etapa de concepción de un proyecto, es una de las actividades más importantes en el desarrollo de un sistema, pero lamentablemente muchas veces no se lleva a cabo con la profundidad necesaria, provocando esto, que no exista una total comprensión de los procesos a informatizar.

2.1.2 Flujo actual de los procesos involucrados en el campo de acción

El flujo comienza cuando el parte diario de las compras realizadas en cada empresa recuperadora del país y el parte de ventas realizadas por la unión, son recibidos por el puesto de dirección de la Unión. El Operador del puesto de dirección es el responsable de recoger la información y organizarla, así, como de emitir los reportes de recuperación y ventas. El Operador realiza copias de los reportes y las envía a los especialistas de cada área (Ferroso, No ferroso y No metálico), al Jefe del área de control, al Director de la UERMP y al Vicedirector de producción del SIME. En las áreas de trabajo los especialistas pueden ser: Especialistas de ventas, Especialistas de compras. Estos especialistas trabajan con diferentes sistemas, entre ellos: el Concilia, SI-Estadístico 451, así, como las Hojas de Cálculo Microsoft Excel. Cuando las copias de los reportes son enviados a las áreas de trabajo, el especialista de compra tiene la responsabilidad de llenar el modelo de salida: recuperación*casa de compra y el especialista de venta tiene la responsabilidad de llenar el modelo de salida correspondiente a las ventas realizadas por la unión: ventas*clientes. Después de llenar cada

uno de los modelos, los especialistas envían la información al Revisor técnico para que verifique los datos registrados. Cuando el Jefe del área de control recibe los reportes, registra las ventas en valores, almacenando la información en los archivos de ventas de la UERMP.

2.1.2 Actores del negocio

Un actor del negocio es: cualquier individuo, grupo, organización o máquina que interactúa con el negocio; este siempre permanece fuera de las fronteras del negocio.

Actor	Descripción
1. Operador del puesto de dirección.	Es el actor encargado de almacenar, organizar y emitir los reportes de la recuperación y las ventas de materias primas.

Tabla 2.1: Actor del negocio y su descripción.

2.1.3 Trabajadores del negocio

Representa a personas o sistemas dentro del negocio, que son las que realizan las actividades que están comprendidas dentro de un caso de uso. Estos trabajadores están dentro de la frontera del negocio, son los que en un futuro se convertirán en usuarios del sistema que se quiere construir.

Trabajadores	Descripción
Especialista de compra.	Recibe copia de los reportes y es el responsable de llenar el modelo de recuperación*casa de compra.
Especialista de venta.	Recibe copia de los reportes y es el responsable de llenar el modelo de ventas*clientes.
Director de la Unión.	Recibe copia de los reportes.
Vicedirector de producción del SIME.	Recibe copia de los reportes.
Jefe del área de control.	Registra las ventas en valores.
Revisor técnico.	Encargado de verificar los datos registrados en los modelos de salida.

Tabla 2.2: Trabajadores y su descripción.

2.1.4 Diagrama de casos de uso del negocio

El modelado de casos de uso produce dos artefactos: los diagramas de casos de uso y las descripciones de los mismos. Los diagramas de casos de uso muestran el comportamiento del sistema a partir de los usuarios que interactúan con el sistema. En otras palabras, describe gráficamente quien utiliza el sistema y la forma en que los usuarios esperan interactuar con él. En la figura 1 se muestra el diagrama de casos de uso correspondiente al actor del negocio.

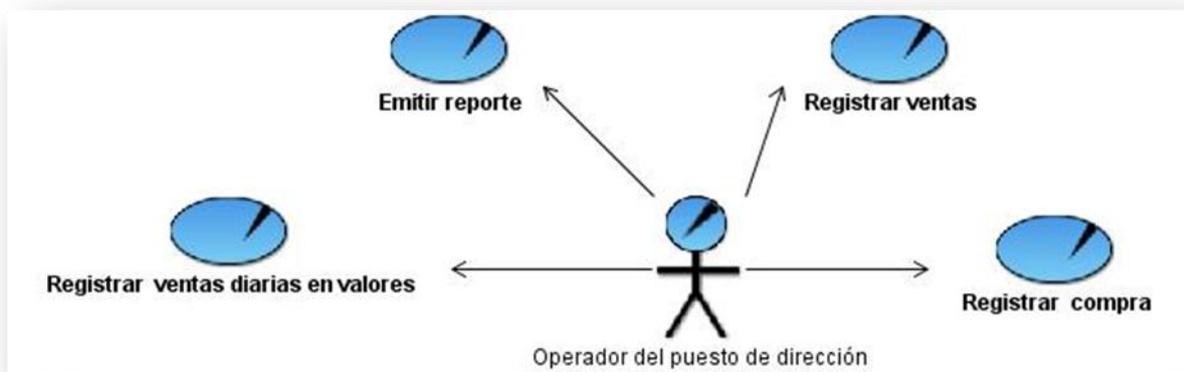


Figura 1. Diagrama de Casos de Uso del Negocio.

2.1.5 Diagrama de actividades

Un diagrama de actividades puede considerarse como un caso especial de un diagrama de estados, en el cual casi todos los estados son estados de acción (identifican una acción que se ejecuta al estar en él) y casi todas las transiciones al término de dicha acción (ejecutada en el estado anterior). Un diagrama de actividades puede dar detalles a un caso de uso, un objeto o un mensaje en un objeto. Permiten representar transiciones internas al margen de las transiciones o eventos externos. ([Ver Anexo 1](#))

2.1.6 Descripciones textuales

Caso de Uso:	Emitir reporte
Actor:	Operador del puesto de dirección
Trabajadores:	Especialista de compra, Director de la unión y el Vicedirector de producción del SIME.

Capítulo 2: Características del sistema

Resumen:	El caso de uso inicia cuando el Operador del puesto de dirección de la Unión recibe los partes diarios de cada una de las empresas recuperadoras de materias primas del país y realiza copias, enviándoselas a cada uno de los trabajadores. Estos se encargan de controlar las informaciones referentes a las compras de materias primas realizadas por cada una de las empresas. El caso de uso culmina cuando el Operador realiza un resumen de los tres reportes, conformando el reporte general a emitir.	
Precondiciones:	El Operador del Puesto de dirección recibe los partes diarios con anterioridad.	
Flujo Normal de Eventos		
Sección 1		
Acción del Actor	Respuesta del Negocio	
<ol style="list-style-type: none"> 1. El Operador del puesto de dirección recibe reportes de las empresas. 2. Realiza copias de reporte consolidado. 3. Emite reporte. 	4. Los trabajadores reciben el reporte emitido.	
Pos condiciones	Los trabajadores reciben el parte emitido.	

Tabla 2.3: Descripción textual del Caso de Uso del Negocio "Emitir reporte".

Caso de Uso:	Registrar ventas	
Actor:	Operador del puesto de dirección	
Trabajadores:	Especialista de venta, Revisor técnico.	
Resumen:	El caso de uso inicia cuando el Operador del puesto de dirección emite los reportes. El Especialista de venta se encarga de registrar las ventas realizadas por la unión, el Revisor técnico verifica que los datos al registrar las ventas sean los correctos. El caso de uso finaliza cuando el Especialista de venta registra las ventas realizadas.	
Precondiciones:	El Especialista de venta recibe los reportes emitidos.	
Flujo Normal de Eventos		
Sección 1		
Acción del Actor	Respuesta del Negocio	

Capítulo 2: Características del sistema

1. El Operador del puesto de dirección envía los reportes al Especialista de venta de cada área de trabajo.	2. El Especialista de venta reúne la información referente a las ventas.
	3. El Especialista de venta actualiza el modelo de salida ventas*clientes.
	4. El Especialista de venta envía el modelo de salida al Revisor técnico para que verifique que los datos actualizados en el modelo de salida ventas*clientes sean los correctos.
	5. El Revisor técnico verifica los datos.
	6. El Especialista de venta registra las ventas realizadas.
Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	5.1 Si el Revisor técnico encuentra errores le regresa el modelo al Especialista de venta para que arregle los errores.
Pos condiciones	Las ventas queden registradas.

Tabla 2.4: Descripción textual del Caso de Uso del Negocio "Registrar ventas".

Caso de Uso:	Registrar ventas diarias en valores
Actor:	Operador del puesto de dirección.
Trabajador:	Jefe del área de control.
Resumen:	El caso de uso inicia cuando el Operador del puesto de dirección emite los reportes. El Jefe del área de control se encarga de almacenar las ventas realizadas en valores. El caso de uso finaliza cuando el Jefe del área de control registra las ventas.
Precondiciones	El especialista de compra recibe el reporte emitido.
Flujo Normal de Eventos	
Sección 1	
Acción del Actor	Respuesta del Negocio
1. El Operador del puesto de dirección envía los	2. El Jefe del área de control reúne la información

Capítulo 2: Características del sistema

reportes al Jefe de área de control.	referente a las ventas.
	3. El Jefe de área de control almacena las ventas realizadas en el modelo: ventas diarias en valores.
	4. El Jefe del área de control registra las ventas en valores.
Pos condiciones	Las ventas queden registradas en valores.

Tabla 2.5: Descripción textual del Caso de Uso del Negocio "Registrar ventas diarias en valores".

Caso de Uso:	Registrar compras	
Actor:	Operador del puesto de dirección.	
Trabajador:	Especialista de compra, Revisor técnico.	
Resumen:	El caso de uso inicia cuando el Operador del puesto de dirección emite los reportes. El Especialista de compra se encarga de registrar las compras realizadas, el Revisor técnico verifica que los datos al registrar las compras sean los correctos. El caso de uso finaliza cuando el Especialista de compra registra las compras realizadas.	
Precondiciones	El especialista de compra recibe el reporte emitido.	
Flujo Normal de Eventos		
Sección 1		
Acción del Actor	Respuesta del Negocio	
1. El Operador del puesto de dirección envía los reportes al Especialista de compra de cada área de trabajo.	2. El Especialista de compra reúne la información referente a las compras.	
	3. El Especialista de compra actualiza el modelo de salida recuperación* casa de compra.	
	4. El Especialista de compra envía el modelo de salida al Revisor técnico para que verifique que los datos actualizados en el modelo de salida recuperación* casa de compra sean los correctos.	
	5. El Revisor técnico verifica los datos.	
	6. El Especialista de compra registra las compras realizadas.	

Flujos Alternos	
Acción del Actor	Respuesta del Negocio
	5.1 Si el Revisor técnico encuentra errores le regresa el modelo al Especialista de compra para que arregle los errores.
Pos condiciones	Las compras queden registradas.

Tabla 2.6: Descripción textual del Caso de Uso del Negocio "Registrar compra".

2.1.7 Diagrama de clases del modelo de objetos

El modelo de objeto representa la relación que tiene cada trabajador del negocio con las entidades del mismo. Las entidades son contenedoras de información persistente del sistema.

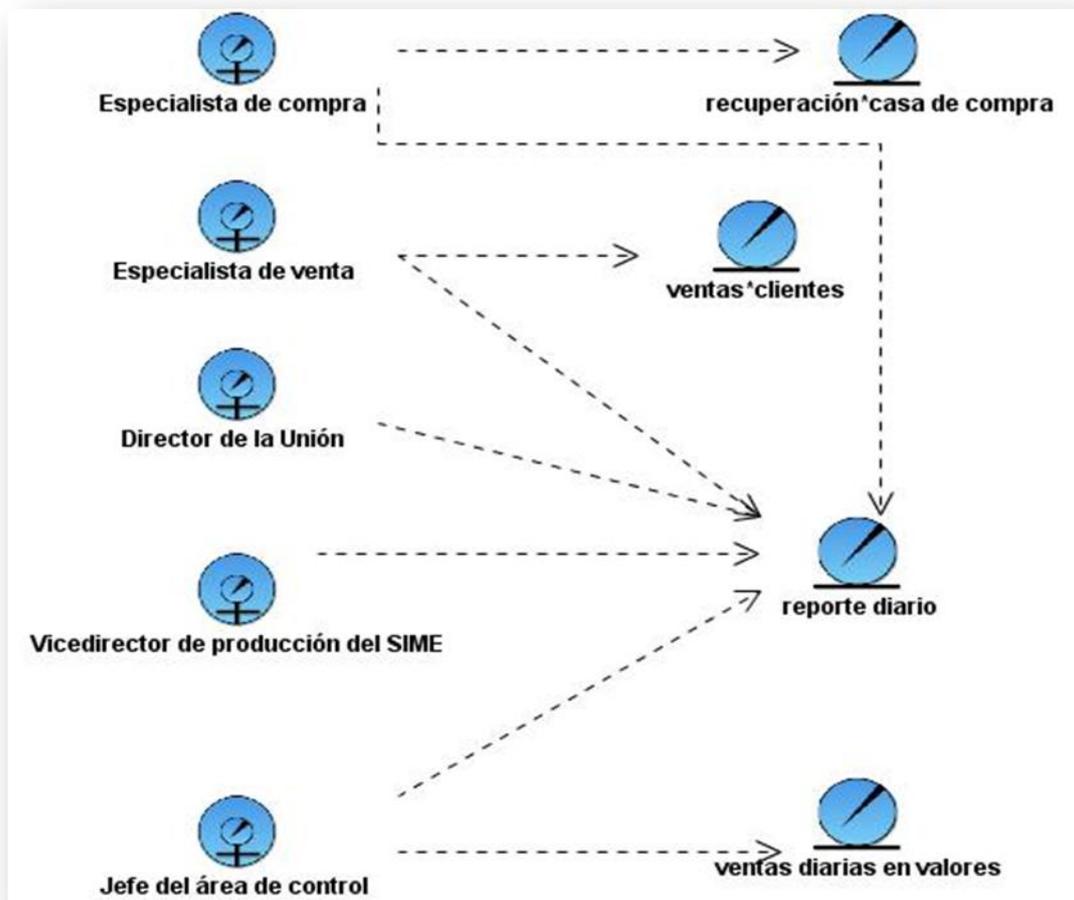


Figura 2. Modelo de objetos del negocio.

2.2 Especificación de los requisitos del sistema

2.2.1 Requerimientos Funcionales

Son capacidades o condiciones que el sistema debe cumplir. En la realización de los casos de uso del negocio, se obtienen las actividades que serán objeto de automatización. Estas actividades no son exactamente los requerimientos funcionales, pero sí son el punto de partida para identificar que debe hacer el sistema. Los requerimientos funcionales se mantienen invariables sin importar con qué propiedades o cualidades se relacionen.

Dentro de los requerimientos funcionales del sistema propuesto se encuentran:

R1. Autenticar usuario

1.1 Introducir los datos del usuario (usuario, contraseña).

R2. Gestionar rol

2.1 Crear rol

2.1.1 Actualizar el listado de los roles existentes.

2.2 Modificar rol

2.2.1 Mostrar el listado de los roles existentes.

2.2.2 Actualizar el listado de los roles existentes.

2.3 Eliminar rol

2.3.1 Mostrar el listado de los roles existentes.

2.3.2 Actualizar el listado de los roles existentes.

R3. Gestionar permiso

3.1 Crear permiso

3.1.1 Actualizar el listado de los permisos existentes.

3.2 Modificar permiso

3.3.1 Mostrar el listado de los permisos existentes.

3.3.2 Actualizar el listado de los permisos existentes.

3.3 Eliminar permiso

3.3.1 Mostrar el listado de los permisos existentes.

3.3.2 Actualizar el listado de los permisos existentes.

R4. Gestionar usuario

4.1 Crear usuario

4.1.1 Actualizar el listado de los usuarios existentes.

4.2 Modificar usuario

4.2.1 Mostrar el listado de los usuarios existentes.

4.2.2 Actualizar el listado de los usuarios existentes.

4.3 Eliminar usuario

4.3.1 Mostrar el listado de los usuarios existentes.

4.3.2 Actualizar el listado de los usuarios existentes.

R5. Gestionar organismo

5.1 Crear organismo

5.1.1 Actualizar el listado de los organismos existentes.

5.2 Modificar organismo

5.2.1 Mostrar el listado de los organismos existentes.

5.2.2 Actualizar el listado de los organismos existentes.

5.3 Eliminar organismo

5.3.1 Mostrar el listado de los organismos existentes.

5.3.2 Actualizar el listado de los organismos existentes.

R6. Gestionar empresa

6.1 Crear empresa

6.1.1 Actualizar el listado de las empresas existentes.

6.2 Modificar empresa

6.2.1 Mostrar el listado de las empresas existentes.

6.2.2 Actualizar el listado de las empresas existentes.

6.3 Eliminar empresa

6.3.1 Mostrar el listado de las empresas existentes.

6.3.2 Actualizar el listado de las empresas existentes.

R7. Gestionar moneda

7.1 Crear moneda

7.1.1 Actualizar el listado de las monedas existentes.

7.2 Modificar moneda

- 7.2.1 Mostrar el listado de las monedas existentes.
- 7.2.2 Actualizar el listado de las monedas existentes.

7.3 Eliminar moneda

- 7.3.1 Mostrar el listado de las monedas existentes.
- 7.3.2 Actualizar el listado de las monedas existentes.

R8. Gestionar tipo de producto

8.1 Crear tipo de producto

- 8.1.1 Actualizar el listado de los tipos de productos existentes.

8.2 Modificar tipo de producto

- 8.2.1 Mostrar el listado de los tipos de productos existentes.
- 8.2.2 Actualizar el listado de los tipos de productos existentes.

8.3 Eliminar tipo de producto

- 8.3.1 Mostrar el listado de los tipos de productos existentes.
- 8.3.2 Actualizar el listado de los tipos de productos existentes.

R9. Gestionar producto

9.1 Crear producto

- 9.1.1 Actualizar el listado de los productos existentes.

9.2 Modificar producto

- 9.2.1 Mostrar el listado de los productos existentes.
- 9.2.2 Actualizar el listado de los productos existentes.

9.3 Eliminar producto

- 9.3.1 Mostrar el listado de los productos existentes.
- 9.3.2 Actualizar el listado de los productos existentes.

R10. Gestionar venta

10.1 Crear venta

- 10.1.1 Actualizar el listado de las ventas existentes.

10.2 Modificar venta

- 10.2.1 Mostrar el listado de las ventas existentes.
- 10.2.2 Actualizar el listado de las ventas existentes.

10.3 Eliminar venta

10.3.1 Mostrar el listado de las ventas existentes.

10.3.2 Actualizar el listado de las ventas existentes.

R11. Registrar compra de producto

11.1 Introducir los datos del producto (código, nombre del comprador, producto, cantidad, medida, precio de compra, moneda).

2.2.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades, como las características que hacen al producto atractivo, usable, rápido o confiable; por ejemplo, pudiera desearse que el sistema responda dentro de un intervalo de tiempo especificado o que obtenga los resultados de los cálculos con un nivel de precisión dado.

Dentro de los requerimientos no funcionales del sistema propuesto, se encuentran:

1. Usabilidad

- ✓ El sistema podrá ser usado por los encargados que trabajan en cada área.
- ✓ La aplicación deberá poseer una interfaz y navegación asequible y funcional tanto para usuarios expertos como para los que no tienen conocimientos profundos en informática.
- ✓ Mantener informado al usuario del resultado de las operaciones.

2. Rendimiento

- ✓ El sistema debe requerir un consumo mínimo de recursos.
- ✓ Debe tener tiempos de respuesta rápidos garantizando de esta forma la agilidad del sistema.
- ✓ Después de instalado el software, el mismo debe ser capaz de soportar gran cantidad de usuarios conectados simultáneamente.

3. Portabilidad

- ✓ El sistema debe ser Multiplataforma.

4. Seguridad

El sistema debe permitir el acceso a cada uno de sus servicios, al personal requerido para esto. La seguridad puede ser tratada en tres aspectos diferentes, estos son:

- ✓ **Confidencialidad:** La información manejada por el sistema debe estar protegida de acceso no autorizado y divulgación.
- ✓ **Integridad:** La información manejada por el sistema será objeto de cuidadosa protección contra la corrupción y estados inconsistentes, de la misma forma será considerada igual a la fuente o autoridad de los datos. Pueden incluir también mecanismos de chequeo de integridad y realización de auditorías.
- ✓ **Disponibilidad:** A los usuarios autorizados se les garantizará el acceso a la información. Los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

5. Apariencia o interfaz externa

- ✓ El sistema debe poseer una interfaz web amigable y sencilla, fácil para la interacción del usuario. Con colores en su interfaz que distingan el logo de la recuperación sin saturación de colores ni imágenes.
- ✓ El sistema proporcionará claridad y correcta organización de la información, permitiendo la interpretación correcta e inequívoca de esta, facilitando el uso de la aplicación.
- ✓ El sistema implementará la ejecución de acciones de una manera rápida, minimizando los pasos a ejecutar en cada proceso.
- ✓ En la interfaz de presentación se mostrará el logo de la recuperación en Cuba.

6. Soporte

- ✓ La aplicación tendrá un período de tiempo en prueba para ver su funcionamiento, para luego brindar servicio de instalación.
- ✓ Se contará con un manual de usuario para que el cliente pueda trabajar con la aplicación.

7. Requerimientos de Software

Para Windows: Computadora personal con sistema operativo Microsoft Windows 95 o superior.

Para Linux: Computadora personal con sistema operativo Linux, distribución Ubuntu 6.5 o superior.

Navegador Mozilla, Netscape, Opera o Internet Explorer 5.0 o superior.

Servidor Web: Apache 2.0.

8. Requerimientos de Hardware

PC cliente con 128 MB de RAM (Mínimo).

PC servidor con 512MB de RAM (Mínimo).

Todas las máquinas deben estar conectadas a la red.

2.3 Modelación del sistema

La modelación de un sistema, muestra la forma en que el sistema tiene que funcionar. Al diagramar las relaciones que hay entre las actividades del sistema, la modelación de sistema facilita la comprensión de las relaciones entre las diversas actividades y el impacto que tienen entre sí. Muestra los procesos como parte de un gran sistema, cuyo objetivo es responder a una necesidad específica del cliente.

2.3.1 Diagrama de casos de uso del sistema

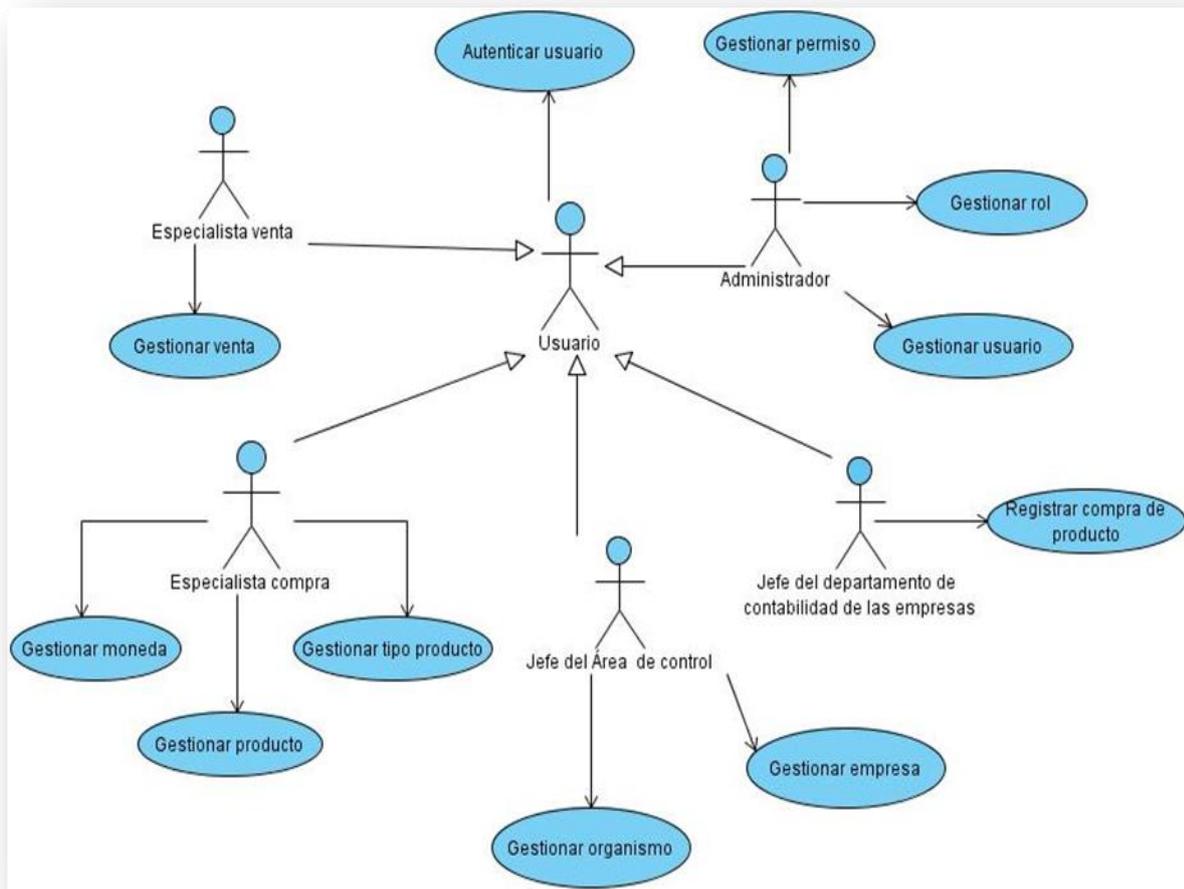


Figura 3. Diagrama de Casos de Uso del Sistema

2.3.2 Descripción de los casos de uso

A continuación se expone una breve descripción de los casos de uso del sistema, para ver la descripción extendida de estos casos de uso del sistema ([Ver Anexo 2](#)).

CU-1	Autenticar usuario
Actores	Usuario
Descripción	El caso de uso inicia cuando el Usuario introduce su usuario y contraseña en el formulario. El sistema verifica si los datos introducidos son correctos. El caso de uso termina cuando se asignan los privilegios de acuerdo al rol.

Capítulo 2: Características del sistema

Referencia	RF 1(RF 1.1)
-------------------	--------------

Tabla 2.7: Descripción del CU Autenticar Usuario.

CU-2	Gestionar rol
Actores	Administrador
Descripción	El caso de uso inicia cuando el Administrador necesita realizar una determinada acción sobre un rol. El sistema permite escoger las opciones de crear, modificar o eliminar un rol. El caso de uso culmina al terminar las acciones correspondientes a la opción seleccionada.
Referencia	RF 2 (RF 2.1, RF 2.2, RF 2.3)

Tabla 2.8: Descripción del CU Gestionar Rol.

CU-3	Gestionar permiso
Actores	Administrador
Descripción	El caso de uso inicia cuando el Administrador necesita realizar una determinada acción sobre un permiso. El sistema permite escoger las opciones de crear, modificar o eliminar un permiso. El caso de uso culmina al terminar las acciones correspondientes a la opción seleccionada.
Referencia	RF 3 (RF 3.1, RF 3.2, RF 3.3)

Tabla 2.9: Descripción del CU Gestionar Permiso.

CU-4	Gestionar usuario
Actores	Administrador
Descripción	El caso de uso inicia cuando el Administrador necesita realizar una determinada acción sobre un usuario. El sistema permite escoger las opciones de crear, modificar o eliminar un usuario. El caso de uso culmina al terminar las acciones correspondientes a la opción seleccionada.
Referencia	RF 4(RF 4.1, RF 4.2, RF 4.3)

Tabla 2.10: Descripción del CU Gestionar Usuario.

Capítulo 2: Características del sistema

CU-5	Gestionar organismo
Actores	Jefe del área de control
Descripción	El caso de uso inicia cuando el Jefe del área de control necesita realizar una determinada acción sobre un organismo. El sistema permite escoger las opciones de crear, modificar o eliminar un organismo. El caso de uso culmina al terminar las acciones correspondientes a la opción seleccionada.
Referencia	RF 5 (RF 5.1, RF 5.2, RF 5.3)

Tabla 2.11: Descripción del CU Gestionar Organismo.

CU-6	Gestionar empresa
Actores	Jefe del área de control
Descripción	El caso de uso inicia cuando el Jefe del área de control necesita realizar una determinada acción sobre una empresa. El sistema permite escoger las opciones de crear, modificar o eliminar una empresa. El caso de uso culmina al terminar las acciones correspondientes a la opción seleccionada.
Referencia	RF 6 (RF 6.1, RF 6.2, RF 6.3)

Tabla 2.12: Descripción del CU Gestionar Empresa.

CU-7	Gestionar moneda
Actores	Especialista de compra.
Descripción	El caso de uso inicia cuando el Especialista de compra necesita realizar una determinada acción sobre una moneda. El sistema permite escoger las opciones de crear, modificar o eliminar una moneda. El caso de uso culmina al terminar las acciones correspondientes a la opción seleccionada.
Referencia	RF 7 (RF 7.1, RF 7.2, RF 7.3)

Tabla 2.13: Descripción del CU Gestionar Moneda.

Capítulo 2: Características del sistema

CU-8	Gestionar tipo de producto
Actores	Especialista de compra.
Descripción	El caso de uso inicia cuando el Especialista de compra necesita realizar una determinada acción sobre un tipo de producto. El sistema permite escoger las opciones crear, modificar o eliminar un tipo de producto. El caso de uso culmina al terminar las acciones correspondientes a la opción seleccionada.
Referencia	RF 8 (RF 8.1, RF 8.2, RF 8.3)

Tabla 2.14: Descripción del CU Gestionar tipo de Producto.

CU-9	Gestionar producto
Actores	Especialista de compra.
Descripción	El caso de uso inicia cuando el Especialista de compra necesita realizar una determinada acción sobre un producto. El sistema permite escoger las opciones de crear, modificar o eliminar un producto. El caso de uso culmina al terminar las acciones correspondientes a la opción seleccionada.
Referencia	RF 9 (RF 9.1, RF 9.2, RF 9.3)

Tabla 2.15: Descripción del CU Gestionar Producto.

CU-10	Gestionar venta
Actores	Especialista de venta.
Descripción	El caso de uso inicia cuando el Especialista de venta necesita realizar una determinada acción sobre una venta. El sistema permite escoger las opciones de crear, modificar o eliminar una venta. El caso de uso culmina al terminar las acciones correspondientes a la opción seleccionada.
Referencia	RF 10 (RF 10.1, RF 10.2, RF 10.3)

Tabla 2.16: Descripción del CU Gestionar Venta.

CU-11	Registrar compra de productos
Actores	Jefe del departamento de contabilidad de las empresas recuperadoras.
Descripción	El caso de uso inicia cuando el Jefe del departamento de contabilidad de las empresas recuperadoras necesita registrar las compras de productos realizadas por las empresas. El sistema permite registrar las compras de productos realizadas. El caso de uso culmina al terminar las acciones correspondientes a la opción seleccionada.
Referencia	RF 11 (RF 11.1)

Tabla 2.17: Descripción del CU Registrar compra de productos.

2.4 Estudio de la factibilidad

La especificación de los requerimientos mediante Casos de, ha probado ser uno de los métodos más efectivos para capturar la funcionalidad de un sistema.

Este hecho se puede apreciar en algunas metodologías actuales ampliamente difundidas, entre ellas RUP, en la cual se propone especificar la funcionalidad de los sistemas mediante la utilización de Casos de Uso. Se realiza un estudio de factibilidad con el objetivo de demostrar si se debe continuar o no con el proyecto. Además, se realiza una estimación del esfuerzo que representaría el desarrollo del proyecto utilizando el método estadístico Análisis de Puntos de Casos de Uso.

2.4.1 Planificación mediante Puntos de Casos de Uso

La estimación mediante el Análisis de Puntos de Casos de Uso es un método propuesto originalmente por *Gustav Karner* de *Objector* y posteriormente refinado por muchos otros autores. Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores (31).

Paso 1: Cálculo de Puntos de Casos de Uso 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.

UAW se calcula 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.

Los criterios se muestran en la siguiente tabla:

Tipo de Actor	Descripción	Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Applications Programming Interface).	1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3

Tabla 2.18: Complejidad de los actores.

En este caso son 5 actores de tipo complejo, Especialista de compra, Especialista de venta, Jefe del Área de control, Jefe del departamento de contabilidad de las empresas y el Administrador.

TIPO	PESO	VALOR	PESO*VALOR
Simple	1	0	0
Medio	2	0	0
Complejo	3	5	15
RESULTADO:			15

Tabla 2.19: Factor de peso de los actores sin ajustar.

Luego se calcula UUCW. Este valor se determina 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. Una Transacción está representada por uno o más pasos del flujo de eventos principal del Caso de Uso, pudiendo existir más de una Transacción dentro del mismo caso de uso.

Tipo de Caso de Uso	Descripción	Peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones.	5
Medio	El Caso de Uso contiene de 4 a 7 transacciones.	10
Complejo	El Caso de Uso contiene más de 8 transacciones.	15

Tabla 2.20: Complejidad de los casos de uso.

TIPO	PESO	VALOR	PESO*VALOR
Simple	5	10	50
Medio	10	0	0
Complejo	15	0	0
RESULTADO:			50

Tabla 2.21: Factor de peso de los casos de uso sin ajustar.

Cálculo de Puntos de Casos de Uso sin ajustar

$$UUCP = UAW + UUCW$$

$$UUCP = 15 + 50$$

$$UUCP = 65$$

Paso 2: Calculando los Puntos de Casos de Uso ajustados

$$UCP = UUCP * TCF * EF$$

Donde:

UCP: Puntos de Casos de Uso ajustados.

UUCP: Puntos de Casos de Uso sin ajustar.

Capítulo 2: Características del sistema

TCF: Factor de complejidad técnica.

EF: Factor ambiente.

Calculando TCF: Factor de complejidad técnica.

Este coeficiente 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 significa un aporte muy importante.

En la siguiente tabla se muestran los factores, el peso de cada uno de ellos y el valor asignado:

Factor	Descripción	Peso	Valor asignado	(Peso x Valor)
T1	Sistema distribuido	2	3	6
T2	Tiempo de respuesta	1	3	3
T3	Eficiencia del usuario final	1	4	4
T4	Procesamiento interno complejo	1	0	0
T5	El código debe ser reutilizable	1	4	4
T6	Facilidad de instalación	0.5	4	2
T7	Facilidad de uso	0.5	4	2
T8	Portabilidad	2	5	10
T9	Facilidad de cambio	1	3	3
T10	Concurrencia	1	3	3
T11	Incluye objetivos especiales de seguridad	1	1	1
T12	Provee acceso directo a terceras partes	1	0	0
T13	Se requieren facilidades especiales de entrenamiento a usuarios	1	0	0
Total	Σ (Peso x Valor)			38

Tabla 2.22: Determinación del factor de Complejidad Técnica.

$$TCF = 0.6 + 0.01 * \sum (P_i * V_i)$$

$$TCF = 0.6 + (0.01 * 38)$$

$$TCF = 0.98$$

Calculando EF: Factor de ambiente

Capítulo 2: Características del sistema

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del factor de ambiente. El cálculo del mismo es similar al cálculo del factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

Factor	Descripción	Peso	Valor asignado	(Peso x Valor)
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	4	6
E2	Experiencia en la aplicación.	0.5	3	1.5
E2	Experiencia en orientación a objetos	1	4	4
E4	Capacidad del analista líder.	0.5	4	6
E5	Motivación.	1	4	4
E6	Estabilidad de los requerimientos.	2	3	6
E7	Personal part-time.	-1	2	-2
E8	Dificultad del lenguaje de programación.	-1	4	-4
Total	Σ (Peso x Valor)			21.5

Tabla 2.23: Determinación del factor ambiente.

$$EF = 1.4 - 0.03 * \sum (P_i * V_i)$$

$$EF = 1.4 - (0.03 * 21.5)$$

$$EF = 0.755$$

Calculando los Puntos de Casos de Uso ajustados

$$UCP = UUCP * TCF * EF$$

$$UCP = 65 * 0.98 * 0.755$$

$$UCP = 48.0935$$

Paso 3: Calculando el esfuerzo del flujo de trabajo:

$$E = UCP * CP$$

Donde: **E**: Esfuerzo estimado en horas-hombres

UCP: Puntos de Casos de Uso ajustados

CF: Factor de conversión.

Para calcular CF se tiene en cuenta lo siguiente: **CF = 20 horas-hombres** (si Total EF \leq 2)

CF = 28 horas-hombres (si Total EF = 3 ó Total EF = 4)

CF = abandonar o cambiar proyecto (si Total EF \geq 5)

Por tanto:

$$E = UCP * CP$$

$$E = 48.0935 * 20$$

$$E = 961.87$$

Paso 4: Calcular el esfuerzo de todo el proyecto

Para calcular el esfuerzo de todo el proyecto se utiliza la siguiente tabla que representa el porcentaje estimado que representa cada actividad en el proyecto. Teniendo el valor del esfuerzo de la actividad de implementación se procede a completar la tabla según operaciones matemáticas. horas-hombres.

ACTIVIDAD	% ESFUERZO	VALOR ESFUERZO
ANÁLISIS	10	240.4675
DISEÑO	20	480.935
IMPLEMENTACIÓN	40	961.87
PRUEBA	15	360.7012
SOBRECARGA	15	360.7012
TOTAL	100%	2404.675

Tabla 2.24: Estimación del esfuerzo total.

Como la jornada laboral de un día de trabajo es de 8 horas y en un mes se trabaja un aproximado de 24 días, entonces, una persona en 1 mes trabaja 192 horas, por tanto:

$$E = E \text{ (horas-hombres)} / 192 \text{ horas-meses}$$

Quedaría, $E = 2404.675 \text{ horas-hombres} / 192 \text{ horas-meses} = 12.53 = 12 \text{ meses-hombres}$.

Calculando el tiempo de desarrollo:

$$\text{Tiempo de desarrollo} = E / \text{cantidad de hombres}$$

$$\text{Tiempo de desarrollo} = 12 / 2 = 6 \text{ meses}$$

2.4.2 Beneficios tangibles e intangibles

Los beneficios que reporta, contribuirán a la mejora de la gestión de la recuperación de materias primas que se llevan a cabo en la Unión, además, simplifica las dificultades de los procesos involucrados en la organización, constituye una propuesta de desarrollo para la futura implementación del sistema. Los beneficios que reporta fundamentalmente son de orden social, pues está dirigido a la gestión de los movimientos de productos de la UERMP, beneficiando a todos los trabajadores involucrados en este proceso. Una vez implementada la propuesta de análisis y diseño, contribuirá a disminuir el tiempo invertido en los procesos de la recuperación, procesamiento y comercialización de desechos útiles.

Conclusiones

En este capítulo se realizó la descripción de los procesos del negocio, se identificaron los requisitos funcionales y no funcionales que debe cumplir el sistema, lo que permitió una correcta identificación de los actores y casos de uso del sistema. También se realizó la estimación del esfuerzo, lo cual se utiliza para determinar el tiempo de duración del proyecto.

Capítulo 3: Análisis y diseño del sistema

3. Introducción

En el presente capítulo se traducen los requerimientos a una especificación que describe como implementar el sistema. El análisis consiste en obtener una visión del sistema, de modo que solo se interesa por los requerimientos funcionales. El diseño es un refinamiento del análisis que tiene en cuenta los requerimientos no funcionales, o sea, como cumple el sistema sus objetivos. El diseño debe ser suficiente para que el sistema pueda ser implementado sin ambigüedades.

3.1 Modelo de Análisis

El modelo de análisis es utilizado fundamentalmente por los desarrolladores para comprender como debe darse forma al sistema, es decir, como debe ser diseñado e implementado. Este modelo sirve como una primera aproximación al diseño (32).

El modelo de análisis contiene las clases y sus objetos organizados en paquetes que colaboran. Entiéndase por clases de análisis las clases interfaz, entidad y control, las cuales describen las realizaciones de los casos de uso. La clase interfaz modela la interacción entre el sistema y sus actores, la clase entidad modela información que posee larga vida y que es a menudo persistente y la clase control coordina la realización de uno o unos pocos casos de uso, coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.

3.1.1 Diagramas de Clases del Análisis

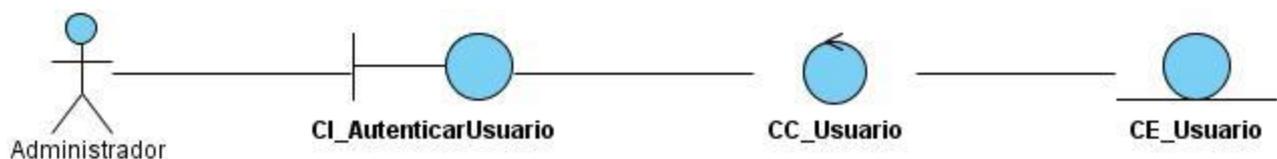


Figura 4. Diagrama de clases del Análisis CU1_AutenticarUsuario.

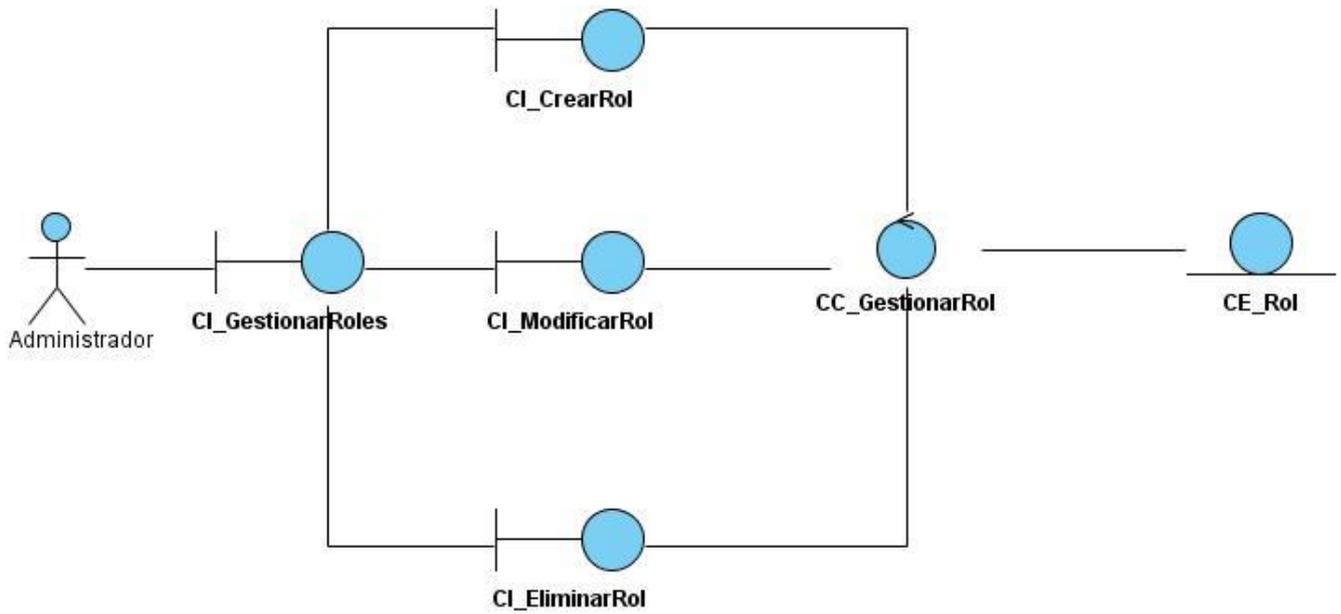


Figura 5. Diagrama de clases del Análisis CU2_GestionarRol.

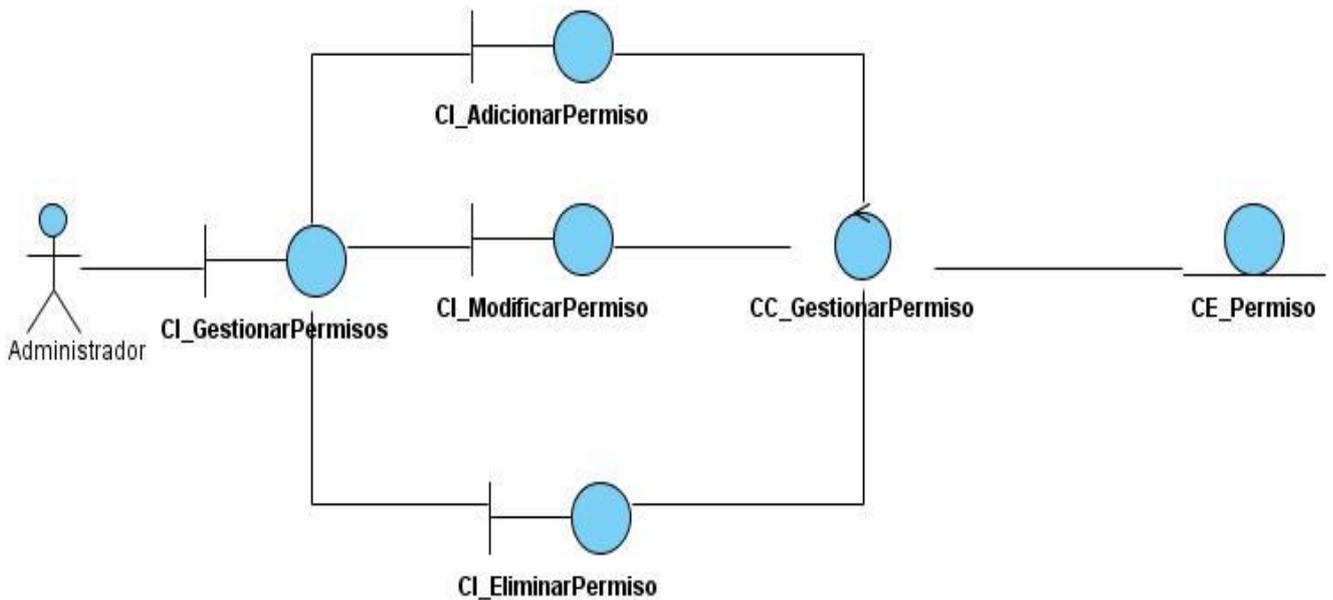


Figura 6. Diagrama de clases del Análisis CU3_GestionarPermiso.

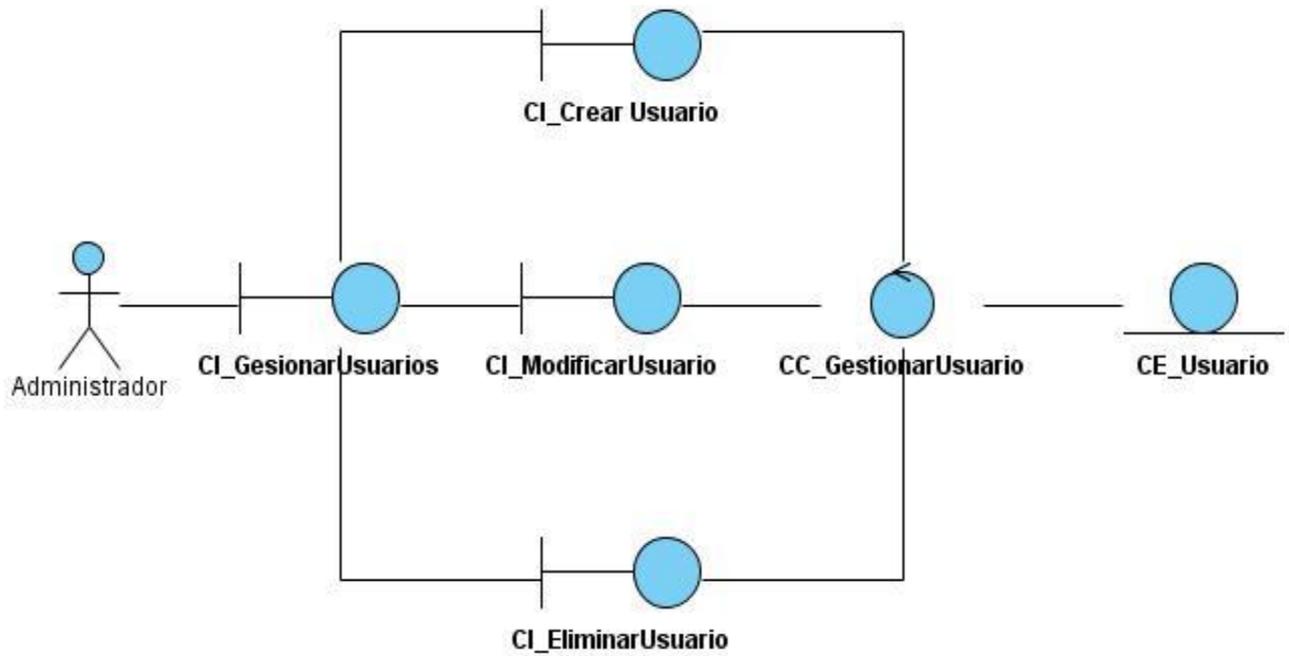


Figura 7. Diagrama de clases del Análisis CU4_GestionarUsuario.

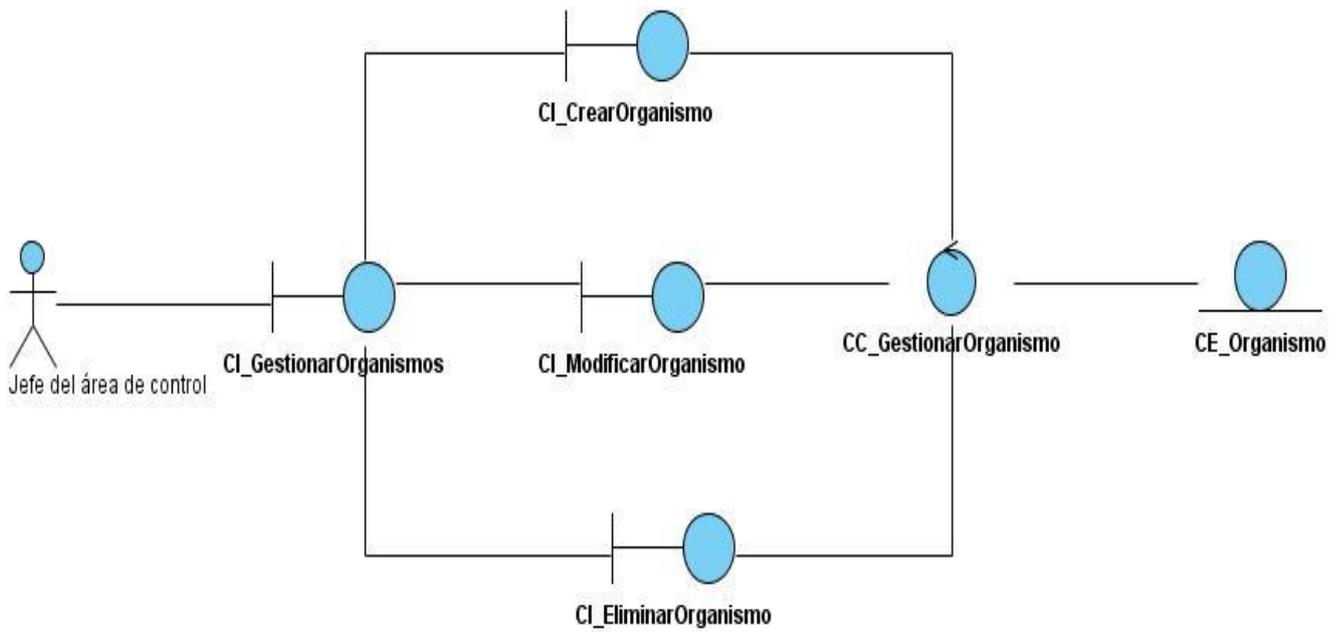


Figura 8. Diagrama de clases del Análisis CU5_GestionarOrganismo.

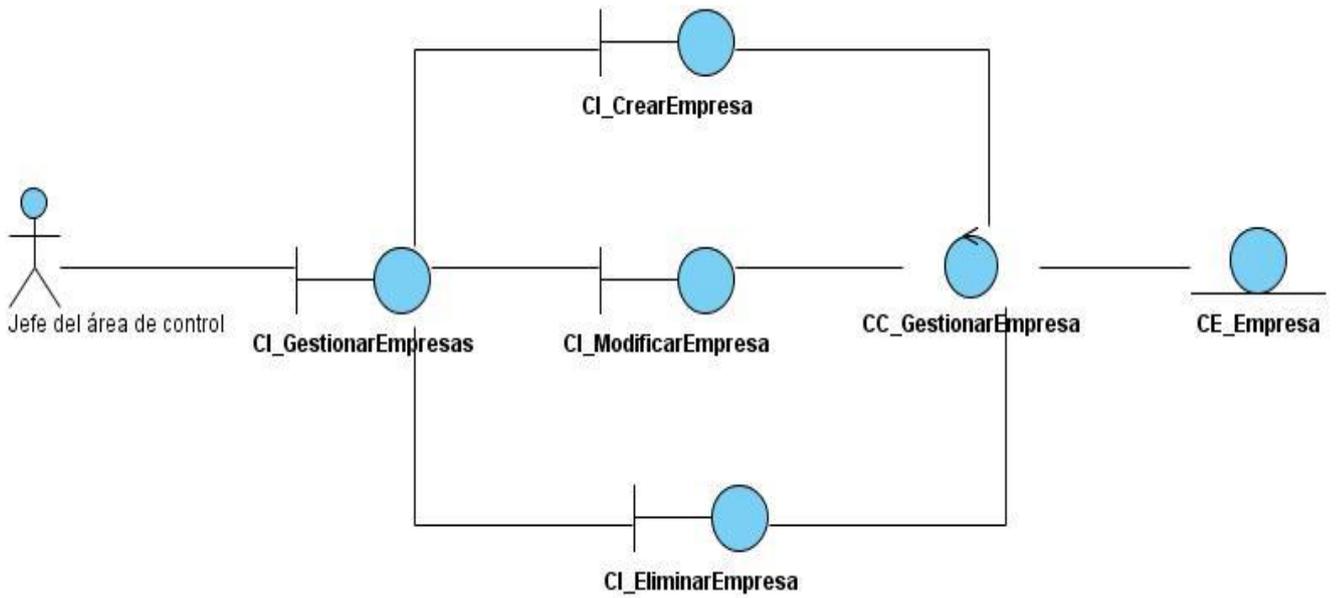


Figura 9. Diagrama de clases del Análisis CU6_GestionarEmpresa.

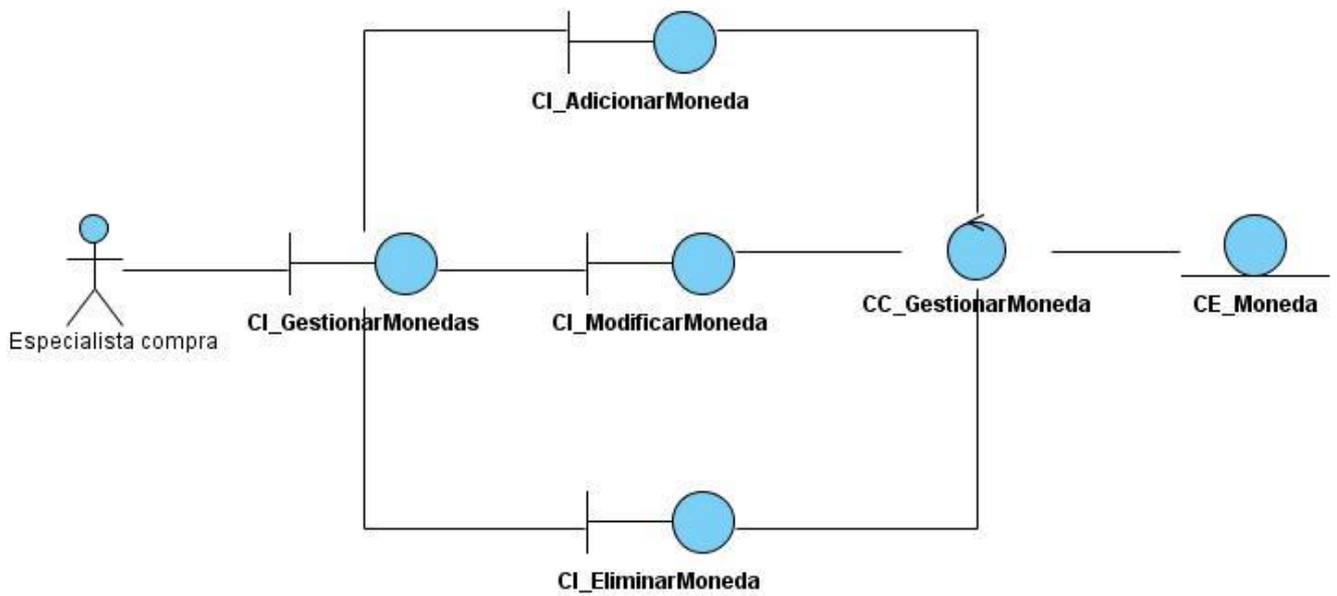


Figura 10. Diagrama de clases del Análisis CU7_GestionarMoneda.

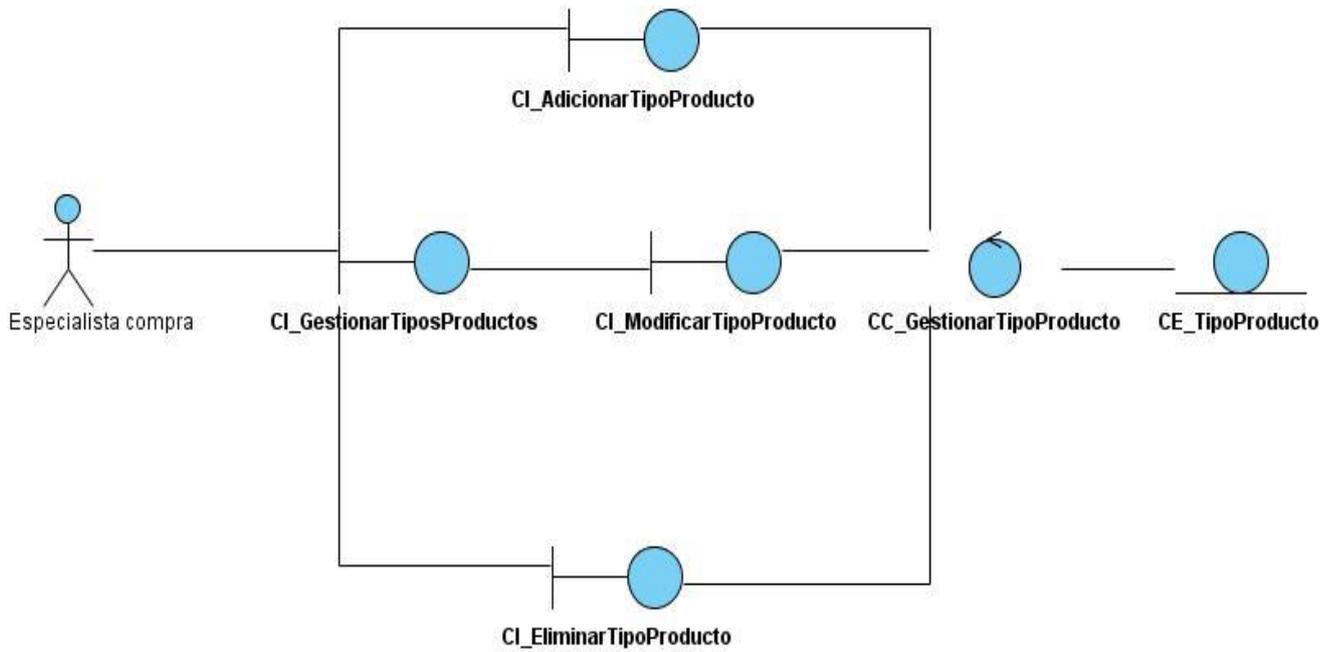


Figura 11. Diagrama de clases del Análisis CU8_GestionarTipoProducto.

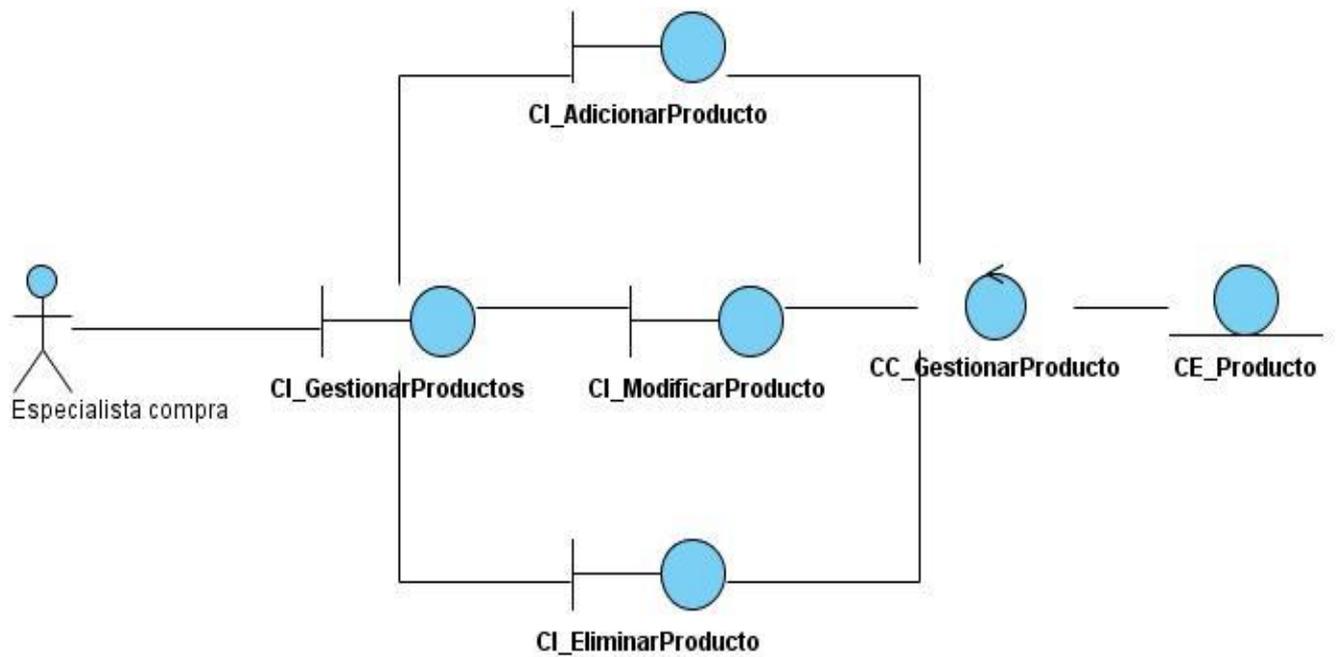


Figura 12. Diagrama de clases del Análisis CU9_GestionarProducto.

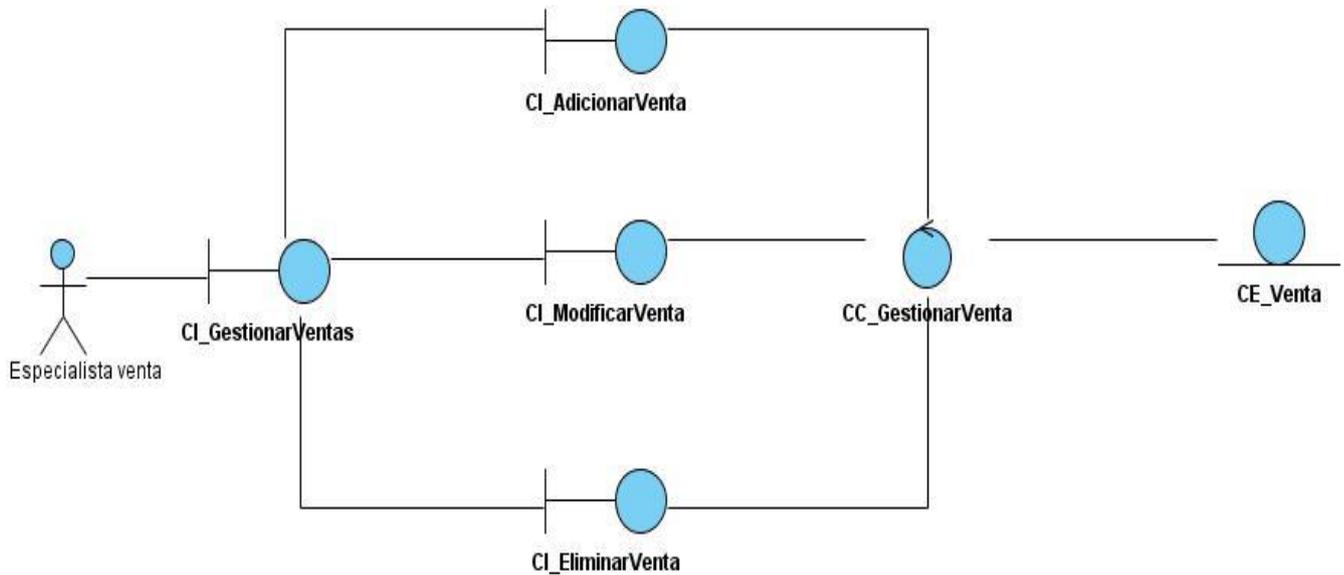


Figura 13. Diagrama de clases del Análisis CU10_GestionarVenta.

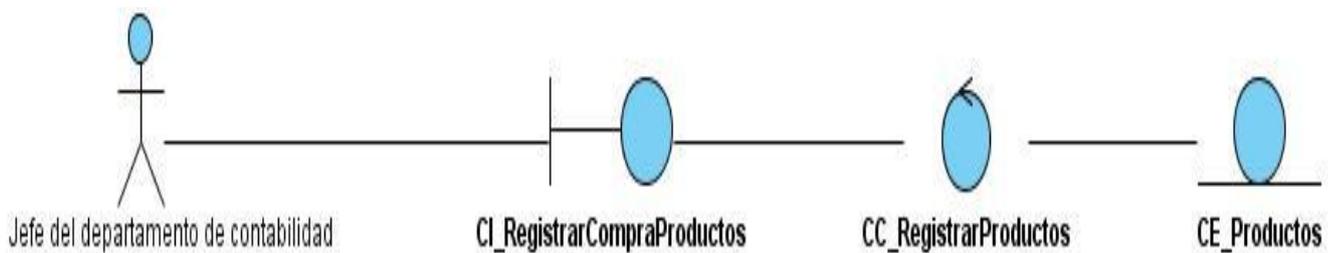


Figura 14. Diagrama de clases del Análisis CU11_RegistrarCompraProducto.

3.1.2 Diagramas de Colaboración

Los diagramas de colaboración son usados para modelar la dinámica del sistema a través de los objetos de las diferentes clases del análisis, sus relaciones y los mensajes que se puedan enviar entre ellos. “Un diagrama de colaboración destaca la organización estructural de los objetos que envían y reciben mensajes” (33).

A continuación se muestra el diagrama de colaboración por escenario del caso de uso Gestionar Ventas. Los otros diagramas de colaboración de los casos de usos para ilustrar el comportamiento del sistema se encuentran en : ([Ver Anexo 3](#))

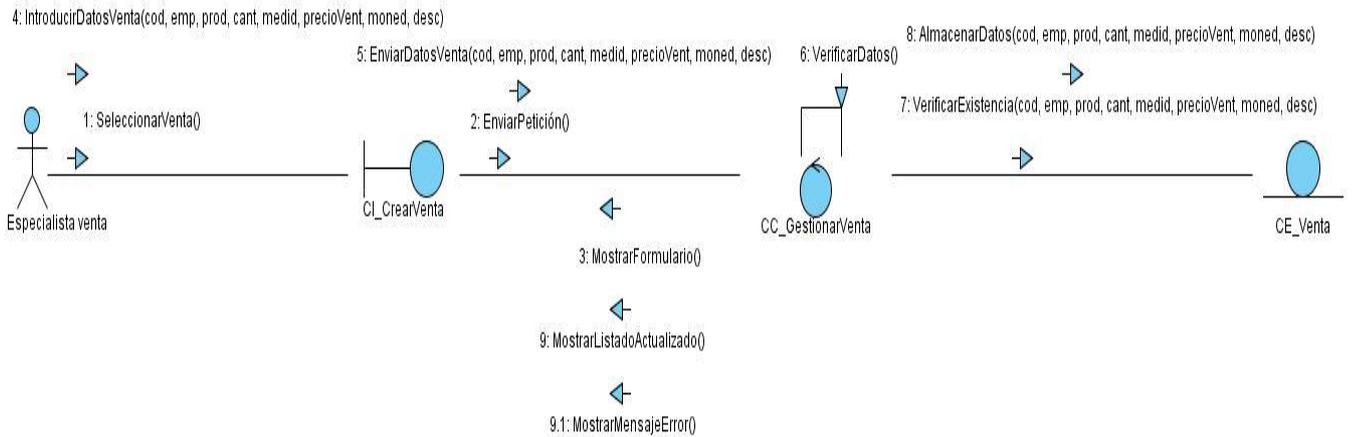


Figura 15. Diagrama de colaboración "Gestionar Venta. Escenario Crear nuevaVenta".

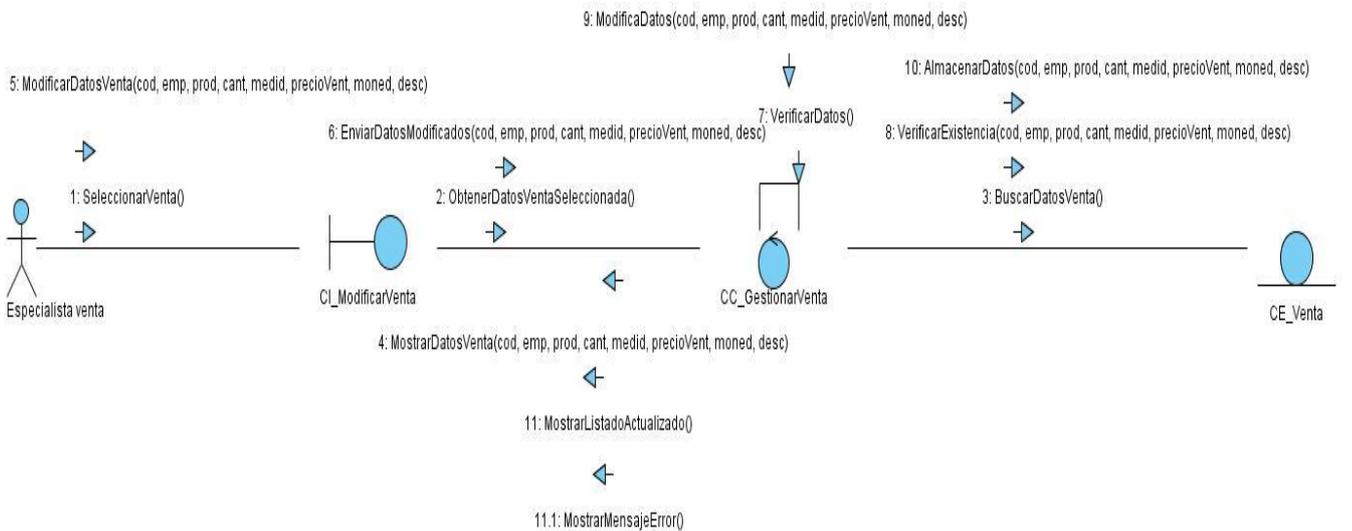


Figura 16. Diagrama de colaboración "Gestionar Venta. Escenario ModificarVenta".

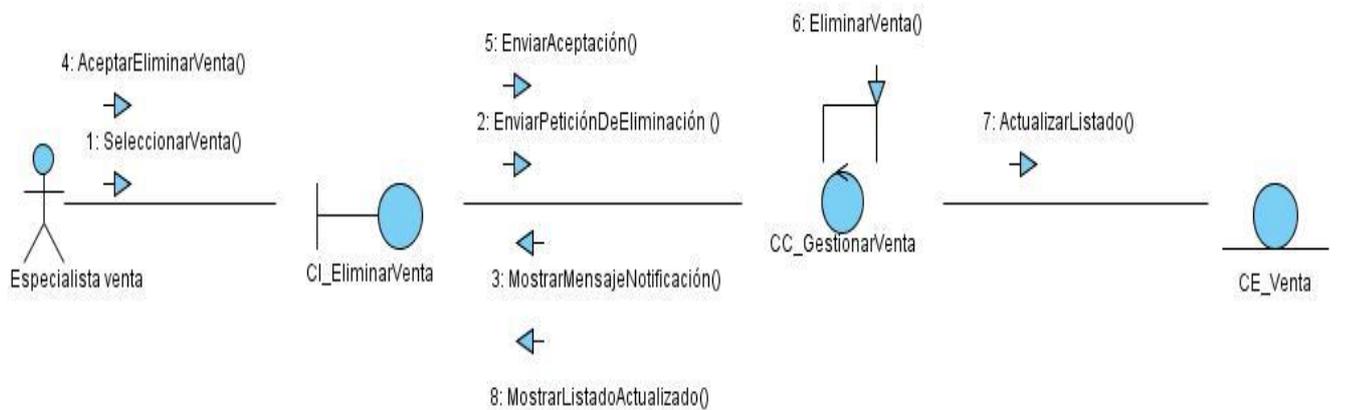


Figura 17. Diagrama de colaboración "Gestionar Venta. Escenario EliminarVenta".

3.2 Modelo de Diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso, sirve como una abstracción del modelo de implementación y su código fuente.

El modelo de diseño es utilizado como una entrada esencial a las actividades de implementación y pruebas (33).

El modelo de diseño es un modelo de objetos que describe la realización de los casos de uso, centrándose en como los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, realizan sus funciones en el sistema a considerar. Esta es la entrada fundamental en las actividades de implementación, para lograr un mejor entendimiento en cuanto al diseño. A continuación se explica en que se basa el flujo de trabajo Symfony.

3.2.1 Estructura de Symfony

Symfony está basado en el patrón clásico del diseño web conocido como arquitectura Modelo Vista Controlador (MVC), pero este implementa todas las ventajas de dicha arquitectura. Este modelo separa la vista (interfaz) y el modelo (base de datos) mediante el controlador que es el encargado de procesar las interacciones del usuario y realizar los cambios apropiados en el modelo o en la vista, además, Symfony hace otra división en el modelo.

Modelo: Symfony divide el modelo en una capa de acceso a datos y otra de abstracción de datos. La abstracción indica que se quiere de la base de datos y la capa de acceso hace las consultas necesarias para obtener la información, de esta forma, si se cambia de base de datos, solamente se cambiaría la capa de acceso y la capa de abstracción podría seguir haciendo las mismas operaciones.

Vista: en la presentación de la mayoría de las páginas existen varios elementos comunes como son: la cabecera, la navegación, el pie de página y la plantilla global conocido como *layout*, cambiando tan solo el interior o contenido de la página. Así, estos tres elementos quedan separados.

Controlador: el trabajo del controlador se repite para muchas acciones. Symfony crea un controlador frontal, único en la aplicación, que está encargado de realizar labores comunes,

como son: el manejo de las peticiones del usuario, el manejo de la seguridad y cargar la configuración de la aplicación, además, es el único punto de entrada a la aplicación.

Symfony utiliza el ORM (Mapeo Objeto Relacional), para gestionar el acceso a la base de datos, esto significa que se puede guardar, obtener, modificar y borrar información en una base de datos sin crear sentencias SQL a mano y sin tener que descender hasta los detalles más técnicos de cada base de datos; además, la otra gran ventaja de los ORM es que se puede cambiar de una base de datos a otra, simplemente cambiando una opción en un archivo de configuración, no es necesario preocuparse por las conexiones de las bases de datos y escribir SQL, tan solo es necesario definir la base de datos en formato XML u obtener la definición desde una base de datos ya existente.

En la realización del diagrama de clases del diseño se tuvo en cuenta la utilización del marco de trabajo Symfony por lo que se representó el mismo en tres capas: Modelo, Vista y Controlador. El paquete Modelo está integrado por las clases que tienen los datos que persisten en la base de datos; el paquete Vista incluye las páginas clientes, las páginas servidoras, los formularios y el paquete Controlador tiene la clase Acciones de su nombre en inglés Actions, dicha clase actúa como controlador frontal y contiene las acciones que se implementan en el caso de uso.

Con la utilización del marco de trabajo Symfony se pueden encontrar diferentes tecnologías que favorecen y facilitan el uso de este marco de trabajo en la elaboración de cualquier aplicación web. Entre ellas se pueden observar:

AJAX (de su nombre en inglés *Asynchronous JavaScript and XML*): un nuevo avance de la tecnología con el objetivo de incrementar el intercambio de datos con el servidor. Con la ayuda de prototipos, se hace de una forma más simple poder actualizar los datos de la aplicación sin necesidad de refrescarla y utilizando pocas líneas de código.

Doctrine: un ORM es una técnica de programación que permite convertir datos, entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y el utilizado en una base de datos relacional, es decir, las tablas de la base de datos pasan a ser clases y los registros objetos que se pueden manejar con facilidad, por lo que favorece la vinculación de un desarrollador a programar en Symfony.

3.2.2 Patrones

El planteamiento de formalizar soluciones a distintas situaciones, de modo que puedan ser entendidas por otros profesionales, es a lo que se llama patrones. Por tanto, un patrón no es más que la descripción detallada de una solución adecuada a un problema concreto. Un patrón es un modelo que podemos seguir para realizar algo. Los patrones surgen de la experiencia de seres humanos de tratar de lograr ciertos objetivos. Capturan la experiencia existente y probada para promover buenas prácticas. Los patrones permiten y han permitido en diferentes áreas del conocimiento rehusar la esencia de la solución de un problema al enfrentar nuevos problemas similares.

Patrones de Casos de Uso

Un patrón de caso de uso es un diseño generalmente probado en un modelo de casos de uso, junto a una descripción del contexto en el cual será usado y las consecuencias que tendrá su aplicación en el modelo. Los patrones de casos de uso capturan buenas prácticas en el modelo de casos de uso, son usados: como plantillas para saber como estructurar el modelo de casos de uso, en el caso de los estructurales, o para saber como deben ser organizadas las descripciones de los casos de uso, en el caso de los descriptivos.

Entre los patrones de casos de uso se encuentran los siguientes:

CRUD: Consiste en un caso de uso llamado Información CRUD o Administrar Información, modelando todas las operaciones que pueden ser realizadas en un segmento de información, como es: crear, leer, actualizar y eliminar. Este patrón es utilizado para la realización de los casos de usos del sistema.

Symfony es capaz de fusionar buenas prácticas de trabajo por sí mismo, de forma que los desarrolladores no tengan que preocuparse por implementar varios de los patrones arquitectónicos y de diseño más utilizados en la actualidad, ya que el mismo marco de trabajo los implementa.

Patrones de Diseño

Patrones GRASP

Creador: todos los módulos del sistema tienen una clase actions.class.php que contiene las acciones definidas para los módulos y es en ella misma donde se ejecutan las funciones que hacen al sistema funcional. En esta clase las acciones se encargan de crear los objetos de las

clases que representan las entidades, evidenciando de este modo que la clase `actions.class.php` es el “creador” de las entidades.

Controlador: todas las peticiones web son manejadas por un solo controlador frontal (`frontend_dev.php`), que es el punto de entrada único de toda la aplicación en un entorno determinado. Cuando el controlador frontal recibe una petición, utiliza el sistema de enrutamiento para asociar el nombre de una acción y el nombre de un módulo con las URL entradas por el usuario.

Experto: Doctrine es la librería externa que utiliza Symfony para realizar su capa de abstracción al modelo de datos, encapsulando toda la lógica de los datos y generando las clases con funcionalidades comunes de las entidades, por tanto, cada clase creada por Doctrine a partir de una entidad es experta en manejar su información.

Alta Cohesión: Symfony permite la asignación de responsabilidades con alta cohesión, por ejemplo, la clase `actions.class.php` tiene la responsabilidad de definir las acciones sobre las plantillas y colabora con otras para realizar diferentes operaciones y crear objetos, está formada por diferentes funcionalidades que se encuentran estrechamente relacionadas, proporcionando que el software sea flexible frente a grandes cambios.

Patrones GOF

Singlenton (Instancia Única): garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Es el caso del controlador frontal, donde hay una llamada a la función `sfContext::getInstance()` que garantiza que siempre se acceda a la misma instancia.

Decorator (Decorador): añade a una clase funcionalidad dinámicamente. El archivo `layout.php`, que también se denomina plantilla global, almacena el código HTML que es común a todas las páginas de la aplicación, para no tener que repetirlo en cada página. El contenido de la plantilla se integra en el `layout` decorando la misma.

Abstrac Factory (Fábrica Abstracta): se utiliza este patrón al trabajar con objetos de distintas familias, de manera que no se mezclen entre sí, haciendo transparente el tipo de familia concreta que se esté usando. Cuando el marco de trabajo necesita, por ejemplo, crear un nuevo objeto, busca en la definición de la factoría el nombre de la clase que se debe utilizar para esta tarea.

3.2.3 Diagramas de Clases del Diseño

A continuación se muestra el diagrama de clases del diseño del caso de uso Gestionar Venta.

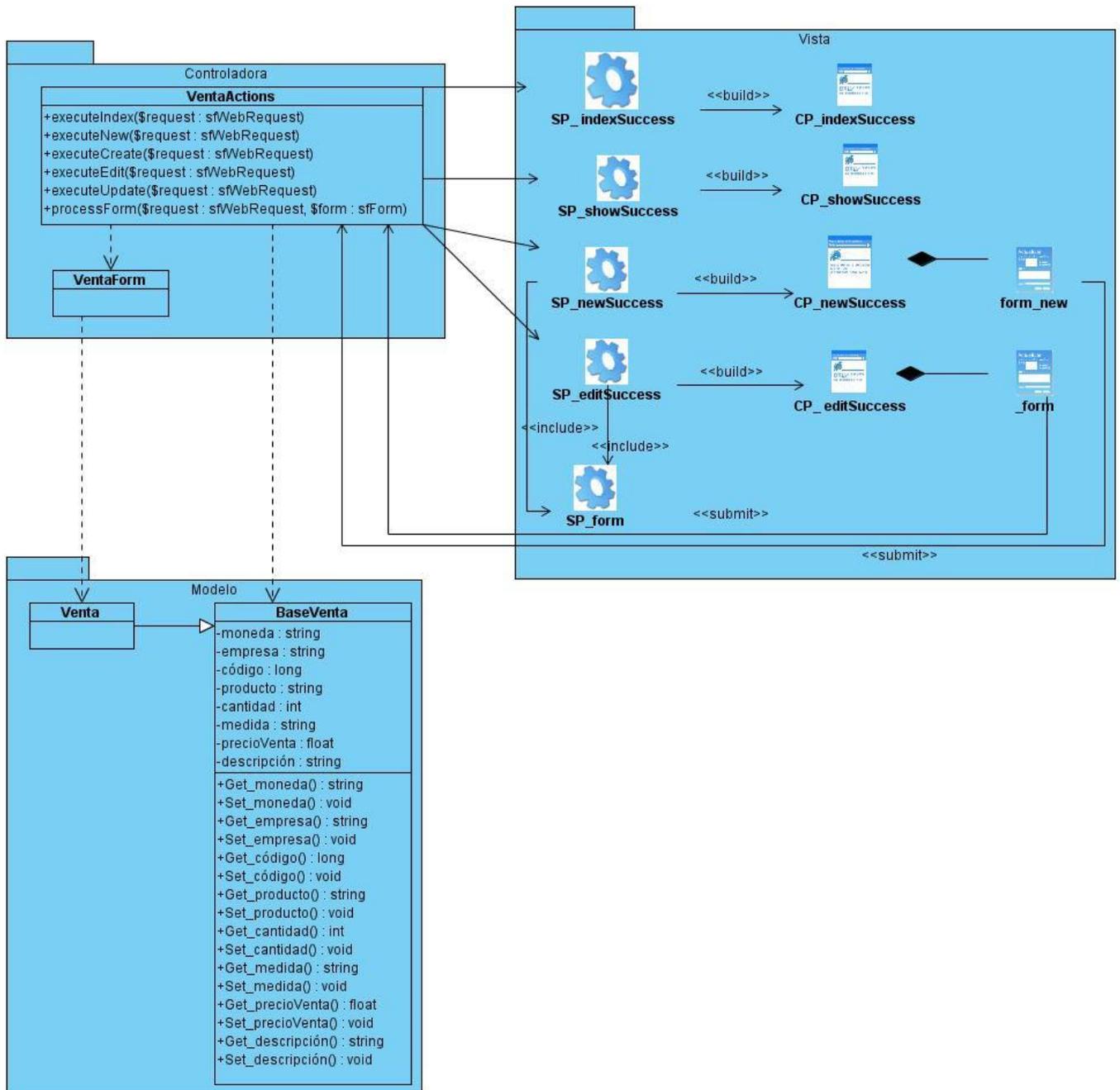


Figura 18. Diagrama de Clases del Diseño CU_GestionarVenta.

El resto de los diagramas de clases del diseño de los casos de uso están: ([Ver Anexo 4](#))

3.3 Diseño de la base de datos

El objetivo principal del diseño de bases de datos es generar tablas que modelan los registros en los que se guarda la información del sistema. Posteriormente se ilustra el modelo lógico y físico de la base de datos.

3.3.1 Diagrama de clases persistentes

Un diagrama de clases muestra un conjunto de clases, interfaces, colaboraciones y sus relaciones. Gráficamente un diagrama de clases es una colección de vértices y arcos. Un diagrama de clases persistentes es justo un tipo de diagrama que muestra un conjunto de objetos que deben ser almacenados en algún repositorio como una base de datos relacional. ([Ver Anexo 5](#)).

3.3.2 Modelo de datos

En el modelo de datos se propone la utilización del **plugin: SfDoctrineGuardPlugin-4.0.1.gz**. Este **plugin** añade características de seguridad y gestión de roles y permisos muy por encima de las que trae Symfony por defecto ([Ver Anexo 6](#)).

3.3.3 Descripción de las tablas de la base de datos

Luego de ilustrado el diagrama de clases persistentes y el modelo de datos, se procede a describir cada una de las tablas de la base de datos, donde se detallan los atributos, el tipo y una breve descripción correspondiente a cada tabla. ([Ver Anexo 7](#))

3.4 Diagrama de despliegue

“Un diagrama de despliegue es un grafo de nodos unidos por conexiones de comunicación. Un nodo es un objeto físico en tiempo de ejecución que representa un recurso computacional, generalmente con memoria y capacidad de procesamiento” (35).

En el diagrama de despliegue se visualiza la relación física de los componentes de la aplicación. Permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware.

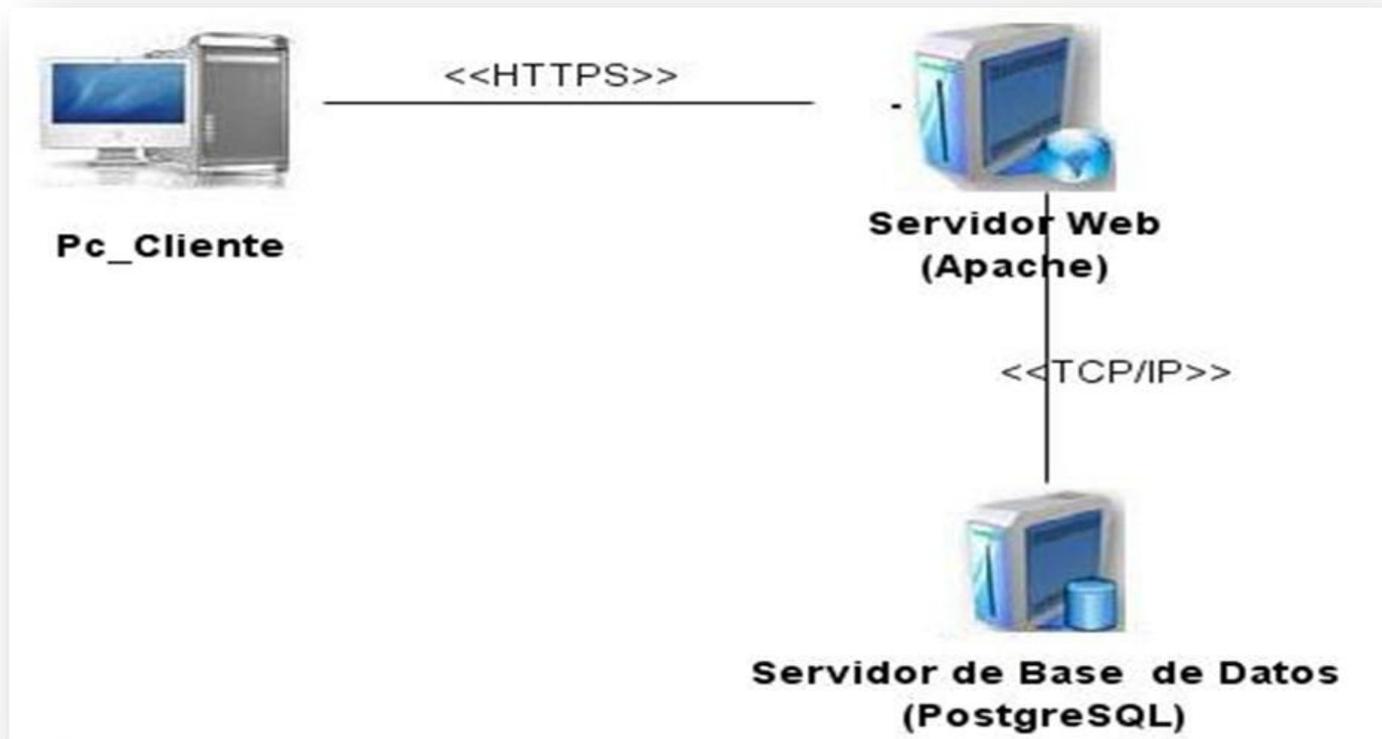


Figura 19. Diagrama de despliegue.

Conclusiones

En este capítulo se realizó el análisis y diseño de cada uno de los casos de uso del sistema además, se mostraron los diagramas correspondientes: los de clases del análisis y de clases del diseño, así, como los diagramas de colaboración para cada una de las funcionalidades descritas. Se describió la arquitectura y los patrones que se tuvieron en cuenta para el diseño de los casos de uso. Además, se realizó el modelo de datos correspondiente a la base de datos, para responder a las necesidades de la aplicación, así, como el diagrama de despliegue. Se llega a la conclusión que el análisis y diseño es muy importante para el desarrollo de cualquier aplicación, pues permite al desarrollador tener una idea clara acerca de lo que se va a hacer para satisfacer las necesidades del cliente.

Conclusiones generales

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

1. Se llevó a cabo un profundo estudio de los Sistemas de Control de Almacenes y Ventas de productos, utilizados en Cuba y en el mundo.
2. Se realizaron entrevistas para conocer cómo se llevan a cabo los movimientos de los productos de la UERMP.
3. Se estudiaron las tecnologías, metodologías, lenguajes y herramientas necesarias para realizar el análisis, diseño y posterior implementación del sistema.
4. 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 ha realizado un análisis y diseño cumpliendo todos los requerimientos que permitirá la futura implementación del Sistema de Gestión de los Movimientos de Productos para la UERMP.

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 hacen necesarias las siguientes recomendaciones:

1. Implementar la propuesta de análisis y diseño del Sistema de Gestión de los Movimientos de productos de la UERMP.
2. Incluir en próximas iteraciones otras funcionalidades en el sistema.

Referencias bibliográficas

1. Historia de la UERMP. UERMP, Ciudad Habana : UERMP. Historia de la UERMP. Ciudad Habana, Cuba.
2. Aplicaciones web. [En línea] <http://www.aplicacionesweb.net/>.
3. Institution, The British Standards. ¿Qué son los sistemas de Gestión? [En línea] <http://www.bsigroup.com.mx/es-mx/Auditoria-y-Certificacion/Sistemas-de-Gestion/De-un-vistazo/Que-son-los-sistemas-de-gestion/>.
4. Trazabilidad 3000. [En línea] <http://www.ceaordenadores.com>.
5. Secop pro. [En línea] <http://www.sistemaspaez.com>.
6. Gestión 3000. [En línea] <http://www.ceaordenadores.com>.
7. Rodas XXI. [En línea] <http://www.rodasxxi.cu>.
8. Siscont. [En línea] <http://siscont.tm.minbas.cu/Docs/Documentos/Manuales/Inventario%20SISCONT5.pdf>.
9. La contabilidad en una nueva tecnología. [En línea] <http://www.betsime.disaic.cu/secciones/>.
10. Pressman, Roger S. *Ingeniería de Software, Un enfoque práctico*. 2001.
11. Canós, José H, Letelier, Patricio y Penadés, M^a Carmen. *Metodologías Ágiles en el Desarrollo de Software*. 2001 : s.n.
12. Crystal Methodologies. [En línea] <http://www.itbuilder.com.mx/blogs/fabiola.soto/post/Metodologias-Agiles.aspx>.
13. Larman, Craig. *UML y Patrones*. 1999.
14. XML. [En línea] <http://manuales.dgsca.unam.mx/xml/Desventajas.htm>.
15. Santamaría, Fesabid. [En línea] http://fesabid98.florida-uni.es/Comunicaciones/f_santamaria/f_santamaria.htm.

16. HTML. [En línea] <http://html.rincondelvago.com/internet-y-html.html>.
17. Pérez, Eguíluz. [En línea] <http://www.librosweb.es/ajax/index.html>.
18. ASP. [En línea] <http://www.programacion.com/asp>.
19. PHP. [En línea] <http://www.forosdelweb.com> › ... › PHP.
20. Programación Web. [En línea]
<http://www.programacionweb.net/articulos/articulo/?num=184>.
21. Quiñones. Introduccion a PostgreSQL. [En línea]
http://www.postgresql.org.pe/articles/introduccion_a_postgresql.pdf.
22. Oracle. [En línea] <http://www.desarrolloweb.com/articulos/840.php>.
23. Visual Paradimg. [En línea] <http://www.webmastersmalaga.com/category/software/>.
24. Rational Rose. [En línea] <http://www.scribd.com/doc/25374125/Estudio-de-Herramientas-CASE-de-Soporte-a-UML-y-UML2>.
25. Extensiones para Web. Entorno Virtual de Aprendizaje. [En línea]
<http://teleformacion.uci.cu/mod/resource/view.php?id=21363..>
26. IIS. [En línea] <http://www.dwaynewhitten.com/info306/iis.ppt>.
27. Thttpd. [En línea] dnolivieri.net/ACSO/AllDocs/01-Grupo1-Webservers2.ppt.
28. Apache. [En línea] dnolivieri.net/ACSO/AllDocs/01-Grupo1-Webservers2.ppt.
29. Gutiérrez. Lenguajes y Sistemas Informáticos. [En línea]
http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf .
30. Symfony. [En línea]
http://librosweb.es/symfony_1_0/capitulo1/symfony_en_pocas_palabras.html.
31. Jacobson, I. y Booch, G. y Rumbaugh. El Proceso Unificado de Desarrollo de software. 2000.

32. Rumbaugh, James, Jacobson, Ivar y Booch, Grady. El Proceso Unificado de Desarrollo de Software, *Vol. I*. 2004.
33. Modelo de Análisis. Continuación del FT Análisis y Diseño. 2008-2009.
34. Corporation. Rational Software. 2003.
35. Vilas, A. F. Diagrama de despliegue. 2001.

Bibliografía

(s.f.). Recuperado el 2010, de

<http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node11.html>

(s.f.). Recuperado el 2010, de

<http://www.scansource.com/MX/Manufacturer%20Partners/POS.aspx>

(s.f.). Recuperado el 2010, de

<http://www.dccia.ua.es/dccia/inf/asignaturas/GPS/archivos/Uml.PDF>

(2010). Obtenido de <http://www.webestilo.com/php/>

Aplicaciones web. (s.f.). Recuperado el 2010, de <http://www.aplicacionesweb.net/>

Desarrollo web. (s.f.). Recuperado el 2010, de <http://www.desarrolloweb.com/manuales/27/>

Lenguajes del lado del servidor. (s.f.). Recuperado el 2010, de

http://www.adelat.org/media/docum/nuke_publico/lenguajes_del_lado_servidor_o_cliente.html

Sistemas de control. (s.f.). Recuperado el 2010, de

http://www.freedownloadmanager.org/es/downloads/sistema_de_control_de_existencias_gratis/

Sistemas de gestión. (s.f.). Recuperado el 2010, de <http://definanzas.com/2008/06/02/erps-sistemas-de-gestion-empresarial/>

Sistemas de gestión empresarial. (2010). Obtenido de

<http://www.slideshare.net/jcfdezmxmanag/sistemas-de-gestin-empresarial>

Glosario de términos

TPV: es el acrónimo de **terminal punto de venta** (de sus siglas en inglés "*POS terminal*" o "*Point of sale terminal*").

CITMATEL: tecnologías de la Información y Servicios Telemáticos

FIFO: método contable diseñado para valorar inventarios y asuntos financieros que involucran dinero que una compañía asocia con inventario de bienes producidos, materias primas, partes o componentes.

Netscape: es una empresa de software famosa por ser la creadora del navegador web *Netscape Navigator*. Fue comprada por AOL (America Online) en 1999.

Sun Microsystems: es una empresa informática recientemente comprada por *Oracle Corporation* antes era parte de Silicon Valley, fabricante de semiconductores y software. Las siglas SUN se derivan de «**Stanford University Network**», proyecto que se había creado para interconectar en red las bibliotecas de la Universidad de Stanford

TomCat: también llamado Jakarta TomCat, funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la *Apache Software Foundation*. TomCat implementa las especificaciones de los servlets y de Java Server Pages (JSP) de *Sun Microsystems*.

Servlets: son objetos que corren dentro del contexto de un contenedor de servlets (ej.: TomCat) y extienden su funcionalidad.

Promedio ponderado con este método se determina un promedio, sumando los valores existentes en el inventario con los valores de las nuevas compras, para luego dividirlo entre el número de unidades existentes en el inventario (incluyendo tanto los inicialmente existentes, como los de las nuevas compras).

Parser: o procesador o analizador sintáctico, lee el documento XML, es la herramienta principal de cualquier aplicación XML. Mediante el Parser no solamente se puede comprobar si los documentos son bien formados o válidos, sino que también se puede incorporarlos a las aplicaciones, de manera que estas puedan manipular y trabajar con documentos XML.

FTP (de sus siglas en inglés de *File Transfer Protocol* - Protocolo de Transferencia de Archivos): en informática, es un protocolo de red para la transferencia de archivos entre

sistemas conectados a una red TCP (*Transmission Control Protocol*), basado en la arquitectura cliente-servidor.

Rollback: es una operación que devuelve a la base de datos algún estado previo. Los *Rollbacks* son importantes para la integridad de la base de datos, a causa de que significan que la base de datos puede ser restaurada a una copia limpia incluso después de que se han realizado operaciones erróneas.

Parche: consta de cambios a un programa, que se aplican para corregir errores, agregarle funcionalidad y actualizarlo.

HTTP: (*HyperText Transfer Protocol*, o Protocolo de Transferencia de Hipertexto).

URL: son las siglas de **Localizador de Recurso Uniforme** (de sus siglas en inglés *Uniform Resource Locator*), es la dirección global de documentos y de otros recursos en la *World Wide Web*.

Drag and drop: arrastrar y soltar, es una expresión informática que se refiere a la acción de mover con el ratón objetos de una ventana a otra o entre partes de una misma ventana.

Struts: es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma J2EE (*Java 2, Enterprise Edition*). *Struts* se desarrollaba como parte del proyecto Jakarta de la *Apache Software Foundation*, pero actualmente es un proyecto independiente conocido como Apache Struts.

ventas*cliente: Modelos de salidas de la UERMP.

recuperación*casa: Modelos de salidas de la UERMP.

Plugin: (complemento) es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica. Esta aplicación adicional es ejecutada por la aplicación principal e interactúan por medio de un API. También se le conoce como **plugin** (de sus siglas en inglés "enchufable"), **add-on** (agregado), **complemento**, **conector** o **extensión**.