

Universidad de las Ciencias Informáticas

Facultad 3



Título: Informatización del proceso Vales de devolución del subsistema Inventario-Cedrux

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

Autor: Javier Ernesto Acebal Alarcón

Tutor(es): MsC. Tahirí Rivero Álvarez

Ing. Annia Verdecia Boza

Ing. Carlos Alberto Giralt Torriente

La Habana

junio de 2013

PENSAMIENTO



La mente que se abre a una nueva idea jamás volverá a su tamaño original

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaro que yo, **Javier Ernesto Acebal Alarcón**, soy el único autor de este trabajo y autorizo a la Facultad 3 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste, firmamos la presente declaración de autoría en La Habana a los _____ días del mes de _____ del año _____.

Javier Ernesto Acebal Alarcón

Tahirí Rivero Álvarez

Annia Verdecia Boza

Carlos Alberto Giralt Torriente

DATOS DE CONTACTO

MsC. Tahirí Rivero Álvarez:

Ingeniera en Ciencias Informáticas, Universidad de las Ciencias Informáticas.

Categoría Docente: Instructora.

Años de graduado: 5.

Analista.

Ing. Annia Verdecia Boza:

Ingeniera en Ciencias Informáticas, Universidad de las Ciencias Informáticas.

Analista.

Ing. Carlos Alberto Giralt Torriente:

Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas.

Arquitecto de base de datos.

AGRADECIMIENTOS

No importa cuánto se viva sino cómo se vive, si se vive bien y se muere joven, se puede haber contribuido más que una persona hasta los ochenta años preocupada sólo de sí misma. La amistad y el amor son las más grandes contribuciones que podemos hacerle a una persona. Por eso deseo agradecer a todas aquellas personas que de una forma u otra siempre pude contar con ellos.

Agradecer a mis padres por apoyarme en todos los momentos de mi vida, por brindarme todo su amor y cariño, por haber estado siempre presente para mí. Los quiero mucho.

A mi hermano del alma que no sabe cuánto me ha dolido estar lejos de él y no ver cómo ha ido creciendo y hacerse un hombre. Te amo Brother.

A toda mi familia que siempre estuvo al tanto de todos los momentos por los que transite en la universidad. A Rober que sabe cuánto lo quiero y prefiero decírselo y no escribirlo, tú los sabes hermano.

Especialmente agradecer a la persona que ha sido mi gran ejemplo a seguir, quien me aportó grandes conocimientos y a quien le debo todo lo que soy. Donde quiera que estés, Te Quiero Papa.

Especialmente agradecer a El Chino, Leidys y Juan Carlos que sin su ayuda no hubiera sido posible realizar este trabajo.

A las grandes amistades que he conocido, con los que he vivido momentos que nunca olvidare, a Gustavo, Juan José y todos mis compañeros de aula y edificio.

DEDICATORIA

A mí mami y papi por ser los mejores padres del mundo y a mi abuela que no me alcanza con dos vidas para agradecerle y demostrarle cuanto la quiero.

RESUMEN

En la actualidad, las organizaciones se ven en la necesidad de transmitir información de alta calidad que esté disponible en el momento oportuno y de agilizar los procesos económicos, basándose en las tecnologías y la información, para mejorar la toma de decisiones. En Cuba, muchas empresas optan por el uso de sistemas de gestión empresarial; sin embargo, estos sistemas no responden totalmente a las funcionalidades requeridas por las entidades para el óptimo control de sus procesos. En la Universidad de las Ciencias Informáticas (UCI) se ha creado el Centro de Informatización de la Gestión de Entidades (CEIGE) que cuenta con el producto CEDRUX, sistema de planificación empresarial que dentro de sus funcionalidades gestiona los procesos contables, los que deben estar debidamente documentados, descritos y modelados. El presente trabajo de diploma comprende el análisis, diseño e implementación del proceso Vales de devolución del subsistema Inventario-CedruX, con vista a satisfacer al cliente. En ese sentido, se realiza un estudio del arte apuntando a los sistemas relacionados con la gestión de Vales de devolución. La implementación de la solución se realizó basándose en el Modelo de Desarrollo propuesto por la subdirección de producción de CEIGE. En el contenido del trabajo se hace además, una descripción de las herramientas, lenguajes y tecnologías necesarios para el desarrollo de la solución. Con la implementación del componente Devolución, totalmente funcional, se cumplen las necesidades plasmadas por el cliente, componente validado mediante las pruebas que se le aplicaron a la implementación de las clases y a las interfaces.

PALABRAS CLAVES: CedruX, devolución e inventario.

Tabla de contenido

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Introducción	5
1.2 Proceso Vales de devolución	5
1.2.1 Sistemas informáticos que implementan los Vales de devolución	6
1.2.2 Valoración de los sistemas estudiados.....	8
1.3 Modelo de Desarrollo	9
1.4 Conclusiones.....	22
CAPÍTULO 2: PROPUESTA DE SOLUCIÓN.....	23
2.1 Introducción.....	23
2.2 Modelo de negocio del proceso Vales de devolución.....	23
2.2.1 Mapa de procesos del negocio.....	23
2.2.2 Diagrama del proceso de negocio.....	24
2.2.3 Descripción del proceso del negocio.....	25
2.2.4 Modelo conceptual.....	27
2.3 Ingeniería de requisitos asociado al proceso vale de devolución.....	28
2.3.1 Requisitos funcionales.....	28
2.3.2 Prototipo de interfaz de usuario principal.....	31
2.4 Diseño de la solución.....	32
2.4.1 Diagrama de clase del diseño.....	32
2.4.2 Diagrama de interacción.....	34
2.4.3 Modelo de datos.....	36
2.5 Implementación de la solución.....	38
2.5.1 Diagrama de componentes.....	38
2.6 Conclusiones parciales.....	39

Capítulo 3: VALIDACIÓN PROPUESTA DE SOLUCIÓN.....	40
3.1 Introducción	40
3.2 Validación del modelo de negocio e ingeniería de requisitos.....	40
3.3 Validación del diseño propuesto	40
3.4 Validación de la aplicación.	44
3.4.1 Prueba de Caja Blanca.	44
3.5 Verificación y validación funcional.	48
3.5.1 Pruebas de caja negra.	48
3.5.2 Diseño de los casos de pruebas.....	49
3.5.3 Análisis de la integridad del sistema	50
3.6 Conclusiones.....	50
CONCLUSIONES	52
RECOMENDACIONES	53
BIBLIOGRAFÍA	54
ANEXOS.....	57

ÍNDICE DE FIGURAS

FIGURA 1 ESTILO ARQUITECTÓNICO MODELO-VISTA-CONTROLADOR.....	15
FIGURA 2 MAPA DE PROCESO DE NEGOCIO.	23
FIGURA 3 PROCESO DE NEGOCIO: DEVOLUCIÓN DE MATERIALES.	24
FIGURA 4 MODELO CONCEPTUAL: DEVOLUCIÓN DE MATERIALES	27
FIGURA 5 PROTOTIPO DE INTERFAZ DE USUARIO.	32
FIGURA 6 DIAGRAMA DE CLASES DEL DISEÑO.....	33
FIGURA 7 EVIDENCIA DEL PATRÓN EXPERTO.....	34
FIGURA 8 EVIDENCIA DEL PATRÓN CREADOR.....	34
FIGURA 9 EVIDENCIA DEL PATRÓN CONTROLADOR	35
FIGURA 10 DIAGRAMA DE CLASES DEL DISEÑO.....	36
FIGURA 11 MODELO DE DATOS.....	37
FIGURA 12 DIAGRAMA DE COMPONENTE.	38
FIGURA 13 GRÁFICA TAMAÑO OPERACIONAL DE CLASE.	41
FIGURA 14 GRÁFICA RELACIÓN ENTRE CLASE.	43
FIGURA 15 MÉTODO CREARDEVOLUCIONACTION.....	45
FIGURA 16 GRAFO DE FLUJO ASOCIADO AL MÉTODO.....	45

FIGURA 17 GESTIONAR VALE DE DEVOLUCIÓN	57
FIGURA 18 LISTAR VALES DE ENTREGA	58
FIGURA 19 CREAR VALE DE DEVOLUCIÓN.....	59
FIGURA 20 MODIFICAR VALE DE DEVOLUCIÓN	60
FIGURA 21 GESTIONAR PRODUCTOS DEL VALE DE DEVOLUCIÓN.....	61
FIGURA 22 LISTAR PRODUCTOS DEL VALE DE ENTREGA	62
FIGURA 23 PRODUCTOS DEL VALE DE DEVOLUCIÓN.....	63
FIGURA 24 IMPRIMIR VALE DE DEVOLUCIÓN.....	64
FIGURA 25 IMPRIMIR LISTADO DE VALES DE DEVOLUCIÓN	64

Índice de Tablas

TABLA 1 PATRÓN EL NOMBRE REVELA LA INTENCIÓN.	13
TABLA 2 PATRÓN COMPLETAR UNA ÚNICA META.	14
TABLA 3 PATRÓN ESCENARIO MÁS FRAGMENTADOS	14
TABLA 4 DESCRIPCIÓN DEL PROCESO DE NEGOCIO: DEVOLUCIÓN DE MATERIALES.....	27
TABLA 5 DESCRIPCIÓN DEL REQUISITO ADICIONAR VALE DE DEVOLUCIÓN.....	31
TABLA 6 DESCRIPCIÓN DE LA TABLA "MOD_DOCUMENTO.DAT_ENTREGA_DEVOLUCION".	37
TABLA 7 RESULTADOS DE LA EVALUACIÓN DE LA RELACIÓN ATRIBUTO/RELACIÓN.	44
TABLA 8 CASO DE PRUEBA PARA EL CAMINO BÁSICO 1.....	47
TABLA 9 CASO DE PRUEBA PARA EL CAMINO BÁSICO 2.....	47
TABLA 10 REQUISITOS A PROBAR.....	50
TABLA 11 DESCRIPCIÓN DE VARIABLE.	50
TABLA 12 JUEGOS DE DATOS A PROBAR.....	50

INTRODUCCIÓN

En el entorno competitivo actual, el flujo de información en toda la empresa es esencial para gestionar el negocio con éxito. El desarrollo de sistemas automatizados de gestión empresarial ha tomado un gran auge a nivel mundial, dándole paso a herramientas que posibilitan que el control logístico dentro de las empresas se realice con mayor efectividad. Un ejemplo de estas herramientas son los Sistemas de Planificación de Recursos Empresariales (ERP, por sus siglas en inglés). Estos sistemas son definidos como sistemas de gestión de información que integran y automatizan muchas de las prácticas de negocio asociadas con los aspectos operativos o productivos de una empresa (1). El propósito de un software de este tipo, es apoyar a los clientes de la empresa, dar tiempos rápidos de respuesta a sus problemas, así como un eficiente manejo de la información que permita la toma de decisiones y minimizar los costos (2). Lo más destacable de un ERP es que unifica y ordena toda la información de la empresa en un solo lugar, de este modo, cualquier suceso queda a la vista de forma inmediata y se acortan los ciclos productivos, lo que brinda que aumente el control en la empresa y se incremente la calidad de servicios y productos (3).

El proceso de informatización de la sociedad cubana, ha propiciado el aumento del uso de herramientas informáticas en los principales sectores del país. En busca del fortalecimiento de la gestión de entidades, se crea el proyecto denominado ERP-Cuba, con el desarrollo del producto Cedrux como solución informática, con el objetivo de unificar los requisitos de funcionalidad, interoperabilidad y seguridad de un sistema económico y financiero. Este sistema integral se desarrolla por el Centro de Informatización de la Gestión de Entidades (CEIGE) ubicado en la Universidad de las Ciencias Informáticas (UCI). El centro productivo dirige sus resultados hacia la esfera de la gestión de empresas y entidades. Tiene como misión el desarrollo de productos y servicios asociados para la gestión integral de entidades, contribuyendo a la formación integral de profesionales que respondan a las necesidades del progreso científico-técnico y socioeconómico del país (4).

El sistema Cedrux cuenta con el subsistema Inventario, el cual se encarga de controlar las existencias de productos, así como los movimientos de entrada y salida que se le realizan a los mismos en el almacén. Además, es responsable de llevar a cabo la gestión de Vales de devolución, proceso de gran importancia para el subsistema ya que a través del mismo se maneja la información referente a los productos devueltos al almacén por alguna entidad y el motivo de dicha devolución. Actualmente el proceso se

realiza de forma manual en las entidades y no se encuentra informatizado en el sistema Cedrux, lo que provoca el aumento en gasto de recursos materiales, pérdida de información y disminuye la productividad de los trabajadores.

Dada la situación antes expuesta condicionó la definición del siguiente **problema a resolver**: La gestión de los procesos de Inventario en Cedrux al no posibilitar la realización del proceso gestionar vale de devolución está afectando la satisfacción del cliente.

Objeto de estudio:

Los procesos de devolución.

Objetivo:

Desarrollar un componente para los procesos de Inventario en Cedrux, que posibilite realizar la gestión de vales de devolución de manera que satisfaga las necesidades del cliente.

Objetivos específicos:

- Establecer un marco teórico relativo a los procesos de recepción y Vales de devolución.
- Desarrollar el proceso referente a los Vales de devolución que necesita el subsistema.
- Validar la solución de la gestión de Vales de devolución en Cedrux.

Campo de acción:

Los Vales de devolución en los sistemas de gestión empresarial.

Tareas de la investigación:

- Resumir las especificaciones de soluciones similares, como propuestas para la inclusión de las mismas al nuevo sistema.
- Realizar la descripción del proceso de Vales de devolución.
- Identificar los requisitos funcionales a partir de los procesos de negocio modelados y de las actividades automatizables identificadas.
- Realizar la Especificación de Requisitos de los requisitos identificados.
- Realizar la administración de los requisitos.
- Realizar el Modelo conceptual.
- Realizar los prototipos de Interfaz de Usuario.

- Realizar las validaciones de los requisitos.
- Realizar el diagrama de componentes.
- Realizar el diseño de clases.
- Implementar la solución.
- Realizar los Casos de Prueba.
- Aplicar los diseños de casos de prueba a la solución que se implemente.
- Realizar el manual de usuario del proceso de Vales de devolución.

Métodos teóricos:

La Modelación: Para crear diagramas y modelos mediante los cuales dar una idea o imagen clara con el objetivo de explicar el funcionamiento de los procesos de negocio.

Sistémico: Para estudiar la Ingeniería de Requisitos mediante la definición de sus principales actividades.

Analítico - Sintético: Para arribar a conclusiones respecto a la información obtenida de la investigación.

Métodos empíricos:

Entrevista: Para obtener la información sobre el proceso de gestión de los Vales de devolución en las entidades, así como los requisitos deseados por el cliente para el futuro sistema.

Idea a defender:

El desarrollo de un componente para la realización de la gestión de vales de devolución para los procesos de Inventario en Cedrux, posibilitará satisfacer las necesidades del cliente.

Estructura del trabajo

El trabajo está estructurado en tres capítulos como se describe a continuación:

En el Capítulo 1 se definen los principales conceptos que a tratar y se realiza un estudio del estado del arte sobre la gestión de Vales de devolución en los sistemas que requieren de esta actividad para la mejora de sus procesos. Además se realiza una fundamentación de la metodología a utilizar para el desarrollo del componente.

En el Capítulo 2 se definen elementos generales para el entendimiento de la propuesta. Se expondrá la modelación del negocio actual. Además se definen los requisitos obteniéndose sus especificaciones. Se realiza una descripción detallada de cada una de las actividades y artefactos.

En el Capítulo 3 se realiza la validación de los requisitos y del diseño, se realizan las pruebas del sistema partiendo de los resultados obtenidos en el diseño para llegar a una conclusión definitiva sobre la efectividad de la propuesta.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

En este capítulo se expone en qué consiste el proceso de Vales de devolución y se realiza un estudio del estado del arte sobre la gestión de los Vales de devolución en los sistemas nacionales y extranjeros que lo implementan. Se presentan además los aspectos teóricos relacionados con el modelo de desarrollo a utilizar, las herramientas, las tecnologías y técnicas que se utilizan en esta investigación. Para un mejor entendimiento de la presente investigación, a continuación se realiza una conceptualización de los principales conceptos que se manejarán.

Vale: Papel que se puede cambiar por una cantidad de dinero, por un objeto o por un servicio. Nota firmada que se da al entregar una cosa y que sirve para demostrar que se ha hecho la entrega (5).

Vales de devolución: Es el documento emitido por un establecimiento comercial de modo voluntario cuando el consumidor opta por devolver uno o más artículos adquiridos en él, y que le permite canjearlo por otros productos del mismo establecimiento con un límite igual al importe del vale de compra y durante el período de validez que figura en el propio documento (6).

1.2 Proceso Vales de devolución

El proceso se inicia cuando el área de una entidad se dirige al almacén para devolver materiales que le fueron entregados mediante un Vale de entrega, que debe tener presente para realizar la devolución porque el mismo es la prueba de la entrega del material. Una vez recibido y verificado que los materiales estén en buen estado, se elabora el Vale de devolución. Luego de esto, el Supervisor del almacén se encarga de ubicar en las áreas de almacenamiento los materiales devueltos y le entrega al Contador del almacén el Vale de devolución para actualizar el submayor de inventario. Quedando plasmadas, en el Vale de devolución, las firmas del Supervisor del almacén y la entidad y así, se formaliza la devolución de materiales; se entregan copias al Supervisor para su archivo y a la entidad que devuelve los materiales como constancia de la devolución (7). En Cedrux este proceso actualmente no se realiza, lo que trae consigo las problemáticas expuestas en la introducción de esta investigación. No obstante en el mundo existen sistemas que sí lo implementan. A continuación se realiza un análisis de los mismos, con el fin de que sirvan de base a la solución que se desea desarrollar.

1.2.1 Sistemas informáticos que implementan los Vales de devolución

Para la selección de los sistemas a analizar, se tuvieron en cuenta los siguientes indicadores:

- Sistemas informáticos que implementen el proceso Vale de devolución para poder indagar en la implementación de las funcionalidades.
- Sistemas certificados que se estén usando actualmente en entidades cubanas o extranjeras con un grado de confiabilidad probado.
- Sistemas Web e implementados en plataformas libres, para en caso de encontrar alguno afín con las necesidades del cliente de Cedrux, poder reutilizar su código.

Como resultado de la búsqueda realizada, se obtuvieron un total de 8 sistemas que cumplían con alguna de las características de búsqueda, no obstante se decide profundizar solo en 5, pues aunque no todos cumplen con todas las características buscadas sí fueron los más cercanos a los criterios seleccionados.

➤ **Software de gestión de almacenes Easy WMS:**

Es un sistema desarrollado por la empresa Mecalux en el año 2000. Es un software dedicado a la gestión de almacenes y que cuenta con una cobertura internacional en más de 70 países. Además es un software basado en tecnologías privativas desarrollado bajo estándares de hardware y software actuales. Tiene una arquitectura cliente-servidor, en la que el servidor contiene los datos y las reglas de negocio y el cliente solamente contiene funcionalidades gráficas para mejorar la velocidad de respuesta (8).

Ejecución del proceso Vales de devolución: Este sistema lleva a cabo la gestión de los Vales de devolución mediante el proceso de recepción, que posibilita la entrada de mercancía al almacén ya sea por la compra a proveedores, por órdenes de fabricación, producción o por devoluciones.

Dentro de las funcionalidades de devoluciones que implementa se encuentran:

1. Alta de devoluciones manuales.
2. Alta de devoluciones asociadas a una orden de entrada.
3. Tratamiento de los contenedores y ubicación de los mismos según las reglas establecidas para la devolución.

➤ **SGA Software Gestión de Almacén:**

Este sistema fue desarrollado por la empresa Asintec, la cual desde 1997 se dedica al desarrollo de sistemas especialmente en los sectores de distribución, logística y comerciales (9). El sistema SGA fue

desarrollado a partir del año 2000 que permite el control, gestión de los movimientos y ubicaciones de productos en uno o varios almacenes. Es un sistema Web y desarrollado sobre tecnologías propietarias.

Ejecución del proceso de Vales de devolución: El Software Gestión de Almacén maneja en la recepción la entrada de devoluciones.

➤ **OpenERP:**

Es un ERP desarrollado actualmente por la empresa matriz OpenERP S.A. en Bélgica, está presente en más de 60 países y ha sido traducido a 30 idiomas. Este sistema cubre las necesidades de las áreas de contabilidad, ventas, compras, y almacén, entre otras (10). OpenERP se suministra bajo licencia GPL, por lo que no se abonan licencias de adquisición. Es un sistema de código abierto, que cumple las 4 libertades del software libre, basado en estándares abiertos y desarrollado con plataformas libres (11).

Ejecución del proceso Vales de devolución: La versión actual de este sistema que se encuentra en la Universidad de Ciencias Informáticas como material de estudio, no cuenta con el proceso de Vales de devolución, sin embargo en su versión 6.0.1 realiza la venta y devolución de mercancías utilizando tanto el módulo *point of sale* como el módulo base *sale*.

➤ **RODAS XXI**

Es un sistema desarrollado por la empresa cubana CITMATEL con el propósito de automatizar el funcionamiento de cualquier empresa o unidad presupuestada. Es un sistema en constante evolución por lo que actualmente se encuentran desarrollando nuevas versiones del producto (12). Este sistema está certificado y es utilizado actualmente por gran parte de las empresas cubanas. Es un sistema de escritorio basado en tecnologías propietarias.

Ejecución del proceso de Vales de devolución: Permite tener un control detallado de las existencias de su entidad, actualizando en el mismo momento que se registra un movimiento. Permite operaciones de entradas y salidas de los almacenes, generando el documento asociado al movimiento de que se trate de forma automática.

➤ **Assets**

Es un sistema comercializado por la firma panameña D'MARCO S.A. y distribuido en Cuba en el año 1997 por INFOMASTER, entidad informática perteneciente a la Empresa Nacional de Producción y Servicios a

la Educación Superior del Ministerio de Educación Superior. Se encuentra desarrollado sobre tecnologías propietarias y es una aplicación cliente-servidor programada en Visual Basic 6.0 que emplea el gestor de bases de datos Microsoft SQL Server 2000 (13).

Ejecución del proceso Vales de devolución: El sistema cuenta con el módulo Devoluciones de mercancías que permite la emisión de Notas de Débitos y Notas de Créditos por devoluciones en cantidades o valores de las mercancías recepcionadas de sus proveedores, facturadas a sus clientes o sacadas del inventario por salidas al costo. Manteniendo actualizados los Submayores de Cuentas por Cobrar y Cuentas por Pagar y el inventario perpetuo del almacén. Generándose automáticamente todos los asientos de diario que permiten dar tratamiento contable a las notas de débitos y créditos (14).

1.2.2 Valoración de los sistemas estudiados

Luego de haber realizado un estudio a los sistemas informáticos antes mencionados y teniendo en cuenta algunos de los aspectos de la realización de la gestión de Vales de devolución, se pudo apreciar que:

1. Algunos de los sistemas permiten la realización de la gestión de Vales de devolución, pero al ser software privativos, no fue posible profundizar en los parámetros relacionados con el proceso mencionado, así como en aspectos tecnológicos y de diseño que pudieran ser útiles en la informatización de dicho proceso para el sistema Cedrux.
2. Varios de estos sistemas realizan la gestión de los Vales de devolución mediante la recepción, aunque es posible realizarlo de esta forma en el sistema Cedrux, pues se desea realizar esta operación como un proceso independiente y no dentro de la recepción. Lo cual posibilita la facilidad del uso del sistema por aquellos usuarios que deseen tener el producto Cedrux.
3. Por otra parte la mayoría son aplicaciones de escritorio, lo que viola una restricción de los clientes de Cedrux.

En el estudio realizado a los sistemas anteriormente descritos, se pudo detectar que no existe actualmente una solución que pueda ser reutilizada de forma total o parcial en el sistema Cedrux para la gestión de los Vales de devolución, por lo que se realiza una solución específica utilizando el modelo de desarrollo que se describe en el epígrafe siguiente:

1.3 Modelo de Desarrollo

Como se ha venido explicando la gestión de los Vales de devolución forma parte del subsistema de inventario del sistema integral de gestión Cedrux y será desarrollado bajo el modelo de desarrollo que ha sido utilizado para la construcción del resto de la solución. En el modelo propuesto por la Subdirección de Producción del CEIGE se describe la secuencia de actividades de alto nivel para la construcción y desarrollo de soluciones. Además se muestran cada uno de los artefactos que se generan en las fases de su ciclo de vida.

A continuación se presentan elementos que se tendrán en cuenta en cada una de las fases por las que transitará el desarrollo del módulo de gestión de Vales de devolución.

Fase Estudio preliminar: se realiza un estudio inicial de la organización del cliente que permite obtener información fundamental acerca del alcance del proyecto y realizar estimaciones de tiempo, esfuerzo y costo.

➤ **Artefactos a generar:**

- Cronograma de actividades: Secuencia de actividades a realizar durante la construcción de la solución.

Fase Modelado de negocio: Es la fase destinada a comprender los procesos de negocio de la gestión de los Vales de devolución. Se comprende cómo funciona el negocio que se desea automatizar para tener garantías de que el software desarrollado va a cumplir su propósito.

➤ **Técnicas candidatas a utilizar para el descubrimiento del negocio:**

- **Entrevistas:** Actividad en la que se realizaron entrevistas a los funcionales que sirven para entender cómo realmente se llevan a cabo las actividades para la gestión de Vales de devolución. Encuentro cara a cara que permite evaluar y medir las necesidades del cliente, donde son evaluados los datos e impresiones capturadas, se realiza además un informe de evaluación objetivo (15).
- **Storyboard:** Permite a través de esbozos informales realizados, llegar a un consenso con el cliente sobre lo que cada proceso debía realizar. Posibilitando la creatividad, la animación a la lectura, la organización y secuencia de ideas a través de un guión (16).

- **Arqueología de documentos:** Se realizó un estudio de la documentación aportada por el cliente, la cual sirve de apoyo para elaborar las descripciones de los procesos de negocio. La aplicación de esta técnica trata de determinar posibles requisitos sobre la base de inspeccionar la documentación utilizada por la empresa (17).
- **Patrones de control de flujo candidatos a utilizar:**
 - **Patrones de control básico:**
 - ✓ **Patrón *Sequence*:** Ejecuta actividades en forma de secuencia. Este patrón es utilizado para modelar dependencia entre tareas, es decir, una tarea no puede empezar hasta que otra no haya terminado.
 - ✓ **Patrón *Split*:** Ejecuta actividades en paralelo. Una división paralela es un punto del proceso donde un camino es dividido en dos o más ramas paralelas las cuales son ejecutadas al mismo tiempo.
 - ✓ **Patrón *Synchronization*:** Sincroniza dos hilos de ejecución paralelos. La sincronización es un punto en el proceso donde dos o más ramas del proceso se unen en una sola. Se llama sincronización porque se espera a que todas las ramas entrantes se completen antes de continuar con la siguiente actividad.
 - ✓ **Patrón *Exclusive Choise*:** Selecciona un camino entre varios posibles. La decisión exclusiva representa un punto en el proceso donde se debe escoger un solo camino de varios disponibles dependiendo de una decisión o de datos del proceso.
 - ✓ **Patrón *Simple Merge*:** Unifica dos caminos alternativos sin sincronización. La unión simple es un punto del proceso donde dos o más caminos alternativos convergen en uno solo. Es un supuesto de este patrón que los caminos alternativos no son ejecutados en paralelo.
 - **Patrones estructurales:**
 - ✓ **Patrón *Arbitrary Cycles*:** Representa un punto en el proceso donde una o más actividades pueden volver a ejecutarse.
 - ✓ **Patrón *Implicit Termination*:** Representa un punto del proceso donde no quedan más actividades que ejecutar. El patrón es utilizado para determinar cuándo la instancia de un proceso se considera completa. Un determinado proceso (o sub proceso) debe terminar cuando no hay tareas pendientes o cuando no hay posibilidad de crear nuevos trabajos en el futuro (18).

➤ **Lenguaje a utilizar:**

- **UML:** Lenguaje Unificado de Modelado (Unified Modeling Language, UML por sus siglas en inglés) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Este entrega una forma de modelar sucesos conceptuales como lo son procesos de negocio y funciones de sistema, además de escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables. Este lenguaje posibilita la captura, comunicación y nivelación de conocimiento estratégico, táctico y operacional para facilitar el incremento de valor, aumentando la calidad, reduciendo costos y reduciendo el tiempo de presentación al mercado; manejando riesgos y siendo proactivo para el posible aumento de complejidad o cambio (19).

➤ **Notación para el modelado de los procesos de negocio:**

- **Businesses Process Modeling Notation:** (BPMN, del español Notación para Modelado de procesos de negocio) es un estándar cuyo principal objetivo es proporcionar una notación fácilmente comprensible por todos los usuarios del negocio, desde los analistas, los desarrolladores técnicos, hasta aquellos que monitorizarán y gestionarán los procesos.

➤ **Para la validación se propone utilizar:**

- Carta de aceptación del cliente: Documento por el cual el cliente expresa la satisfacción del modelado del negocio y que está de acuerdo con el mismo.

➤ **Artefactos a generar:**

- **Mapa de proceso de negocio:** Mapa el cual muestra la organización en paquetes de todos los procesos existentes en el negocio y como se relacionan entre ellos.
- **Diagrama del proceso de negocio:** Diagrama que ofrece el conjunto de tareas relacionadas lógicamente que se llevan a cabo para lograr como resultado un negocio definido.
- **Modelo conceptual:** Permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del problema.
- **Descripción del proceso de negocio:** Proceso en el que se describen las reglas sobre cómo realizar el proceso para transformar las entradas en salidas. Describiendo todos los flujos asociados a la devolución de materiales.

Fase Requisitos: En esta etapa se realizará la ingeniería de requisitos perteneciente al proceso de negocio anteriormente modelado.

- **Técnicas candidatas a utilizar para la elicitación de requisitos:** Actividad que se realiza con el fin de conocer el dominio del problema e identificar las necesidades reales de clientes y usuarios, para esto son consultadas distintas fuentes de información como clientes, usuarios y expertos en el dominio.
 - **Entrevistas:** Técnica utilizada para obtener información sobre el negocio o sistema a automatizar y a partir de estos es que se definen los requisitos. Para realizar esta actividad se debe estudiar el dominio del sistema, con el objetivo de conocer los conceptos fundamentales para el cliente y que este vea que se entienden sus pensamientos.
 - **Tormenta de ideas:** Esta técnica consiste en una reunión de pocas personas, donde los participantes exponen sus ideas de forma libre y espontánea. Se usa para generar ideas que suelen ofrecer una visión general de las necesidades del sistema.
 - **Arqueología de documentos:** Trata de determinar posibles requisitos sobre la base de inspeccionar la documentación utilizada por la empresa; por ejemplo, manuales de procedimientos, reglamentos, boletas y facturas.
 - **Sistemas existentes:** Consiste en analizar distintos sistemas ya desarrollados que estén relacionados con el sistema que se intenta construir. Las interfaces de usuario pueden ser analizadas con el objetivo de observar el tipo de información que se maneja y cómo se maneja.
 - **Storyboards:** Esta técnica radica en representar sobre papel y de forma muy esquemática, las diferentes interfaces al usuario. Muy usada por los diseñadores gráficos de aplicaciones en el entorno web. Posibilita la creatividad, la animación a la lectura, la organización y secuencia de ideas a través de un guión (16).
- **Herramientas a utilizar:**
 - **OSRMT:** Para el seguimiento de los requisitos. Herramienta de Software Libre pensada para asistir en todo el Ciclo de Vida del Desarrollo del Software. Permite la descripción avanzada de diversos tipos de requisitos y garantiza la trazabilidad entre todos los documentos relacionados con la ingeniería de requisitos (funcionalidades, requisitos, casos de uso, casos de prueba) (20).

- **Visual Paradigm 8.0:** Para el diseño de prototipos de interfaz de usuario. Las denominadas herramientas CASE que utiliza UML como notación para elaborar los modelos son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software (21). Es una herramienta que facilita y permite a las organizaciones visualizar, diseñar, integrar y distribuir sus aplicaciones empresariales (22).
- **Para la validación de los requisitos se pueden utilizar las siguientes técnicas:**
 - **Revisiones:** Técnica en la cual un grupo de personas se ocupan de revisar el documento de requisitos y los modelos. Consta de tres fases: búsqueda de problemas, reunión y acuerdos.
 - **Auditorías:** Consiste en un chequeo de los resultados contra una lista de chequeo predefinida o definida a comienzos del proceso, es decir sólo una muestra es revisada.
 - **Prototipos:** Permite al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario, permitiendo descubrir con rapidez si el usuario se encuentra satisfecho o no, con los requisitos.
- **Patrones de casos de uso posibles a aplicar:** De los patrones de casos de usos que pueden ser aplicados a los requisitos se propuso utilizar:

Nombre	El nombre revela la intención.
Problema	Los nombres genéricos sin sentido pueden no ubicar las expectativas del lector o proveer un punto focal que las personas pueden convenientemente echar para atrás. Utilizar nombres descriptivos para los casos de uso es una buena práctica, porque ellos revelan exactamente la intención de cada caso de uso.
Solución	Nombra los casos de uso utilizando un verbo activo o frase que represente la meta del actor primario.

Tabla 1 Patrón El nombre revela la intención.

Nombre	Completar una Única Meta.
Problema	Objetivos inadecuados pueden perder a los escritores a la hora de determinar dónde un caso de uso termina y otro comienza.

Solución	Escribir cada caso de uso dirigiéndose hacia una completa y bien definida meta.
-----------------	---

Tabla 2 Patrón Completar una única meta.

Nombre	Escenario más Fragmentos.
Problema	El lector debe ser capaz de fácilmente seguir el camino a través del flujo específico o historia en la que están interesados, en caso contrario probablemente se frustren o no se entienda la información importante
Solución	Escribe los eventos del flujo principal como un escenario simple sin considerar posibles fallos. Debajo ubica los fragmentos del flujo que muestran que condición alternativa puede ocurrir.

Tabla 3 Patrón Escenario más fragmentados

CRUD: es el acrónimo de Crear, Recuperar, Actualizar y Eliminar. Este patrón es una excepción típica al nombrar un caso de uso, según su objetivo. Es utilizado con la idea de agrupar varios objetivos en un caso de uso, llamado por convención, Gestionar<X>. Por ejemplo: los objetivos "crear Vale", "eliminar Vale" se satisfacen en el caso de uso Gestionar Vales de devolución.

➤ **Artefactos a generar:**

- Listado de requisitos.
- Clasificación de requisitos.
- Especificación de requisitos
- Prototipos de interfaz de usuario.

Fase de Análisis y diseño: En esta etapa se realizará el diseño de la solución que se propone para que soporte los requisitos funcionales anteriormente elicitados. El diseño de este componente se basará en la misma arquitectura que se utiliza en Cedrux.

➤ **Estilos arquitectónicos candidatos a utilizar:**

- **Modelo-Vista-Controlador (MVC):** patrón de arquitectura de las aplicaciones software. Separa la lógica de negocio de la interfaz de usuario, lo que facilita la evolución por separado de ambos aspectos e incrementa reutilización y flexibilidad. El objetivo de este tipo de modelos es intentar

repetirse lo menos posible y de tener todo organizado y hacer una distinción entre la lógica de toda la aplicación y presentación (23).

- ✓ **Modelo:** Esta capa es la representación específica de la información con la cual el sistema opera. Se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas.
- ✓ **Vista:** Esta capa presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- ✓ **Controlador:** Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y notifica a la vista.



Figura 1 Estilo arquitectónico Modelo-Vista-Controlador

➤ **Patrones de diseño posibles a aplicar:**

- **Patrones GRASP:** Patrones de software para asignación de responsabilidades (General Responsibility Assignment Software Patterns, GRASP acrónimo en inglés), los cuales describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (24).

- ✓ **Creador:** El patrón creador es evidenciado en la mayoría de las clases del marco de trabajo ya que varias de las clases tienen la responsabilidad de crear instancias de otras, varias clases son creadoras de los objetos de otras. Ejemplo de esto son las clases del paquete *Domain*, quienes son las encargadas de crear los objetos de tipo *Doctrine_Query*.
- ✓ **Experto:** Se muestra en la definición de las clases controladoras. Plantea la asignación de responsabilidades al experto en información, es decir, la clase que cuenta con la información necesaria para cumplir las funcionalidades que deben realizarse a partir de la información manejada dentro del componente (25).
- ✓ **Controlador:** Este patrón plantea asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase. Donde un evento del sistema es un evento de alto nivel generado por un actor externo; es decir, un evento de entrada externa (24). La clase controladora definida, la cual se denomina *GestvaledevolucionController* es un ejemplo de la aplicación de este patrón, y se encargará de manejar los eventos dentro del componente en cuestión.
- **Herramientas a utilizar:**
 - **Visual Paradigm 8.0:** Herramienta descrita anteriormente en la fase de requisitos.
 - **Para la validación del diseño se pueden utilizar las siguientes métricas:** Se puede definir validación como la aportación de datos que respalden la existencia o veracidad de que se han cumplido los requisitos para una utilización o aplicación específica prevista. La validación se aplica sobre todo, en el ámbito del proceso de diseño y desarrollo de un producto. Se realiza de acuerdo con lo planificado para asegurarse de que el producto resultante es capaz de satisfacer los requisitos para su aplicación especificada o uso previsto, cuando sea conocido (26).
 - **Tamaño Operacional de Clase:** Está dado por el número de métodos asignados a una clase y evalúa los siguientes atributos de calidad:
 - ✓ **Responsabilidad:** Un aumento del TOC implica un aumento de la responsabilidad asignada a la clase.
 - ✓ **Complejidad de implementación:** Un aumento del TOC implica un aumento de la complejidad de implementación de la clase.
 - ✓ **Reutilización:** Un aumento del TOC implica una disminución del grado de reutilización de la clase.

- **Relaciones entre Clase:** Está dado por el número de relaciones de uso de una clase con otra y evalúa los siguientes atributos de calidad:
 - ✓ **Acoplamiento:** Un aumento del RC implica un aumento del Acoplamiento de la clase.
 - ✓ **Complejidad de mantenimiento:** Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
 - ✓ **Reutilización:** Un aumento del RC implica una disminución en el grado de reutilización de la clase.
 - ✓ **Cantidad de pruebas:** Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.
- **Artefactos a generar:**
 - **Diagramas de secuencias:** Usados para modelar los aspectos dinámicos de un sistema, son una variante del Diagrama de Actividad, muestran un panorama general del flujo de control dentro del sistema o proceso de negocio (27).
 - **Diagramas de clases:** Es uno de los diagramas principales para el análisis y diseño de un software ya que sirve además para visualizar las distintas relaciones estructurales.
 - **Diagrama de la base de datos el Modelo de datos:** Esquema en él se realiza la descripción de una base de datos. Describe las estructuras de los datos, así como su tipo, descripción, restricciones de integridad entre otros.

Fase Implementación: En esta etapa se realizará la implementación del diseño de la solución que se propone para que soporte los requisitos funcionales.

- **Herramientas a utilizar:**
 - **NetBeans IDE 7.3:** Es un entorno de desarrollo de aplicaciones para múltiples plataformas, ofrece un rendimiento y experiencia de codificación considerablemente mejor, con las nuevas capacidades de análisis de código estático en el Editor de Java y más inteligente de proyectos de exploración. La nueva versión de **NetBeans IDE** también incluye características notables, como la integración con el Generador de Escena para la creación de formas visuales JavaFX, soporte de múltiples frameworks PHP y muchas otras mejoras en Java EE, Maven, C / C++ y la plataforma de desarrollo (28).
 - **Subversion en su versión 1.6.6:** Es un sistema de control de versiones: maneja los archivos y las carpetas de un proyecto y sus modificaciones en el transcurso del tiempo. SVN es libre y Open

Source, ya que está distribuido bajo la licencia Apache. Subversion trabaja replicando el modelo cliente/servidor: uno o más clientes se conectan a un servidor central que tiene la última copia del proyecto, como también copias de sus versiones anteriores, dentro de lo que llamaremos un repositorio. Es un sistema centralizado para compartir información que permite realizar modificaciones atómicas y gestionar archivos, directorios y sus cambios a través del tiempo, lo que facilita las tareas administrativas (29).

- **Postgres SQL 9.1:** Es un sistema de base de datos relacional perteneciente al ámbito del software libre que destaca por su robustez, escalabilidad y cumplimiento de los estándares SQL. Cuenta con versiones para una amplia gama de sistemas operativos, entre ellos: Linux, Windows, Mac OS X, Solaris, BSD, Tru64 y otros más (30).
- **Apache 2.0:** Es un servidor web gratuito, potente y que ofrece un servicio estable y sencillo de mantener y configurar. Es indiscutiblemente uno de los mayores logros del Software Libre. Algunas de sus características son que es multiplataforma, sencillo de configurar, amplias librerías de PHP y Perl a disposición de los programadores, posee diversos módulos que permiten incorporarle nuevas funcionalidades, es capaz de utilizar lenguajes como PHP, TCL y Python.
- **Firefox 3.6:** Es un navegador libre y de código abierto. Es usado para visualizar páginas web. Incluye corrector ortográfico, búsqueda progresiva y marcadores dinámicos. Además se pueden añadir funciones a través de complementos desarrollados por terceros. Es multiplataforma, permite la integración con el antivirus, realiza la navegación por pestañas, presenta compatibilidad para múltiples extensiones y utiliza el sistema SSL para proteger la comunicación con los servidores web, utilizando fuerte criptografía cuando se utiliza el protocolo HTTP (31).

➤ **Frameworks a utilizar:**

- **ExtJS en su versión 2.2:** Permite que con pocas líneas de código sea posible realizar interfaces amigables para los usuarios. Es la librería más avanzada para el desarrollo rápido de aplicaciones con una apariencia totalmente novedosa y una arquitectura flexible. ExtJS es una librería Javascript que permite construir aplicaciones complejas en Internet, además permite crear aplicaciones complejas utilizando componentes predefinidos así como un manejador de layouts similar al que provee Java Swing, gracias a esto provee una experiencia consistente sobre cualquier navegador, evitando el tedioso problema de validar que el código escrito funcione bien en cada uno (32). Usada en la capa de presentación para construir interfaces.

- **Zend Framework:** Para el desarrollo de aplicaciones y servicios web con PHP. Se trata de un framework para desarrollo de aplicaciones y servicios web con PHP, brinda soluciones para construir sitios web modernos, robustos y seguros. Está formado por una serie de métodos estáticos y componentes que usarán estos métodos. Trabaja con MVC. El Marco de Zend también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar base de datos, sin tener que escribir ninguna consulta SQL (33).
 - **Doctrine:** Usado para el mapeo de relación de objetos para PHP. Es un potente y completo sistema ORM (object relational mapper) para PHP 5.2 con un DBAL (*database abstraction layer*) incorporado. Brinda la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir convertir clases (convenientemente creadas siguiendo las pautas del ORM) a tablas de una base de datos (34) .
 - **Sauxe 2.0:** Para la gestión de entidades y compuesto por los descritos anteriormente. Cuenta con una arquitectura en capas que a su vez presenta en su capa superior un MVC. Contiene un conjunto de componentes reutilizables que provee la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo (35).
- **Lenguajes a utilizar:**
- **PHP 5.2 (Para la programación del lado del servidor):** Es un lenguaje de código abierto interpretado de alto nivel, especialmente pensado para desarrollo web y el cual puede ser incrustado en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil. Es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales.
 - **JavaScript (Para la programación del lado del cliente):** Es un lenguaje de programación que se puede utilizar para construir sitios Web y para hacerlos más interactivos. Aunque comparte muchas de las características y de las estructuras del lenguaje Java, fue desarrollado independientemente. El lenguaje Javascript puede interactuar con el código HTML, permitiendo a los programadores web utilizar contenido dinámico. Se utiliza en páginas web HTML, para realizar tareas y operaciones en el marco de la aplicación cliente (36).

➤ **Artefactos a generar:**

- Código de la aplicación.
- Prototipos de interfaz funcionales.
- Diagrama de componentes.

Fase de pruebas internas: Durante esta fase se desarrollan las pruebas del grupo de calidad del centro verificando el resultado de la implementación. Permite identificar posibles errores en la documentación y el software, es decir requisitos que el producto debería cumplir y que aún no los cumple.

➤ **Para la verificación y validación funcional se propone:**

- Pruebas de caja negra: Las pruebas de caja negra se llevan a cabo sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa. Se parte de los requisitos funcionales, a muy alto nivel, para diseñar pruebas que se aplican sobre el sistema sin necesidad de conocer como está construido por dentro. Para el desarrollo de estas pruebas se emplean las siguientes técnicas:
 - ✓ **Partición de Equivalencia:** representa una de las más efectivas pues permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que de otro modo requerirían la ejecución de muchos casos antes de detectar el error genérico. Dirige la definición de casos de pruebas que descubran clases de errores, reduciendo el número de clases de prueba que hay que desarrollar (37).
 - ✓ **Análisis de Valores Límites:** El análisis de valores límite (AVL) es una técnica de diseño de casos de prueba que completa a la técnica Partición de equivalencia. En lugar de seleccionar cualquier elemento de una clase de equivalencia, el AVL lleva a la elección de casos de prueba en los extremos de la clase. En lugar de centrarse solamente en las condiciones de entrada, el AVL obtiene casos de prueba también para el campo de salida, según lo indicado por (38).

Para un mejor uso de las técnicas mencionadas anteriormente, las mismas son aplicadas de forma independiente a cada componente de la solución, haciendo uso de los niveles de prueba que a continuación se describen.

Niveles de prueba: A la hora de evaluar dinámicamente un sistema, se debe comenzar por los componentes más simples y pequeños e ir avanzando progresivamente hasta probar todo el software en su conjunto. Las pruebas se aplican en distintos niveles de trabajo, tales como:

- **Pruebas de Integración:** los componentes individuales son combinados con otros componentes para asegurar que la comunicación, enlaces y los datos compartidos ocurran apropiadamente.
- **Pruebas del Sistema:** son usualmente conducidas para asegurar que todos los módulos trabajan como sistema sin error. Es similar a la prueba de integración pero con un alcance mucho más amplio.
- **Pruebas de Aceptación:** Son realizadas principalmente por los usuarios con el apoyo del equipo del proyecto. El propósito es confirmar que el sistema está terminado, que desarrolla puntualmente las necesidades de la organización y que es aceptado por los usuarios finales.
- **Pruebas de Unidad:** conocidas también como unitarias son pruebas individuales a las unidades separadas de un sistema de software.

Los niveles que se proponen utilizar en la realización de las pruebas del componente son los correspondientes a las Pruebas de Aceptación y Pruebas de Unidad, con el objetivo de corregir los errores encontrados en el primero, antes de que el segundo nivel sea aplicado por el Departamento de Calidad del CEIGE.

➤ **Para la validación del código se proponen realizar las siguientes pruebas:**

- **Pruebas de caja blanca:** Conociendo el código y siguiendo su estructura lógica, se pueden diseñar pruebas destinadas a comprobar que el código hace correctamente lo que el diseño de bajo nivel indica y otras que demuestren que no se comporta adecuadamente ante determinadas situaciones.

➤ **Artefactos a generar:**

- Diseño de casos de prueba.
- Documento de no conformidades.
- Acta de liberación.

1.4 Conclusiones

Finalizado el presente capítulo se arribaron a las siguientes conclusiones:

- Los sistemas utilizados actualmente en el país no se adaptan a las necesidades de independencia tecnológica. El estudio de los mismos brindaron ventajas y desventajas que pueden ser utilizadas en el desarrollo e implementación del sistema.
- Las tecnologías y herramientas definidas a utilizar para el desarrollo de los artefactos a entregar descritos por el Modelo de desarrollo, brindan la posibilidad de desarrollar una solución que apoye el cumplimiento de los principios de soberanía e independencia tecnológica y posibilite la realización de la gestión de los Vales de devolución, sentándose de esta manera; el marco teórico de la investigación.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

2.1 Introducción

En este capítulo se presenta la solución realizada para el componente Vale de devolución del subsistema de inventario. Son tratados los elementos inmersos en el desarrollo del componente. Se realiza una valoración de la propuesta del diseño del sistema exponiendo las principales ventajas y deficiencias del mismo. Se presentan los requisitos funcionales y no funcionales con los que debe cumplir el módulo propuesto. Se hace referencia a las técnicas empleadas tanto en la captura como en la validación de los requisitos.

2.2 Modelo de negocio del proceso Vales de devolución

El objetivo del modelo del negocio es describir los procesos existentes u observados, con el propósito de comprenderlos. Se representan el grupo de actividades que las personas implicadas en el proceso efectúan para alcanzar una meta. Además permite visualizar el comportamiento del sistema donde un proceso puede ser evaluado y corregido determinando si es técnicamente factible. Como resultado de este modelado gana una mayor comprensión del negocio en términos de sus objetivos, sus procesos del negocio, su estructura organizacional, así como sus reglas del negocio.

2.2.1 Mapa de procesos del negocio

El mapa de procesos muestra la organización de todos los procesos existentes en el negocio y como se relacionan entre ellos.

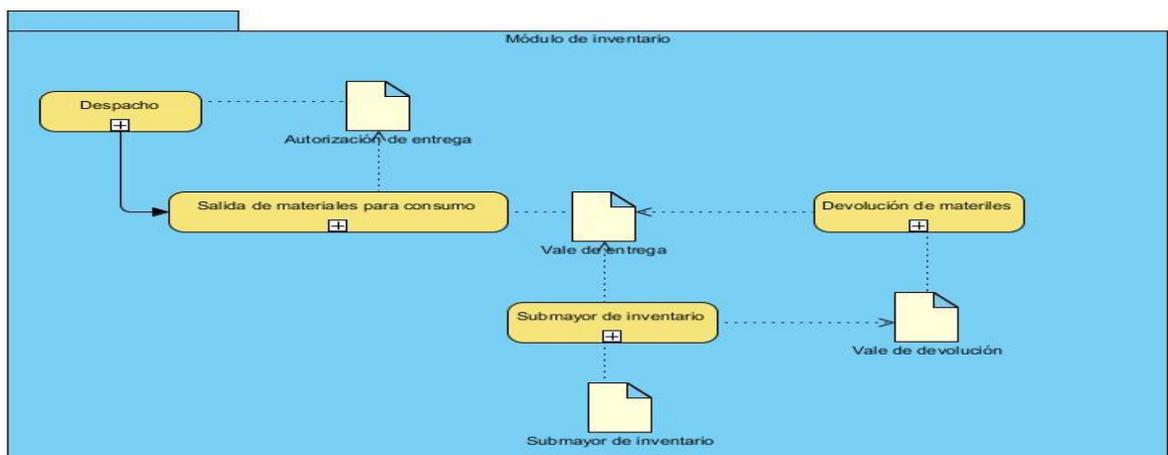


Figura 2 Mapa de proceso de negocio.

2.2.2 Diagrama del proceso de negocio

Un proceso de negocio tiene como objetivo fundamental satisfacer al cliente y no es más que un conjunto de tareas relacionadas lógicamente que se llevan a cabo para lograr como resultado un negocio definido. En el diagrama del proceso de negocio que a continuación se muestra se aprecian las áreas así como las personas implicadas en el negocio. El mismo es iniciado cuando un cliente decide devolver alguna mercancía que le ha sido entregado por el almacén, haciéndole entrega de estos al Supervisor del almacén con constancia del vale de entrega que le fue dado al despacharle la mercancía. Una vez que el Supervisor verifica el estado de los bultos, elabora el Vale de devolución y ubica los materiales, seguido de esto el Contador del almacén actualiza el submayor de inventario y entrega copias del Vale de devolución al Supervisor y al cliente.

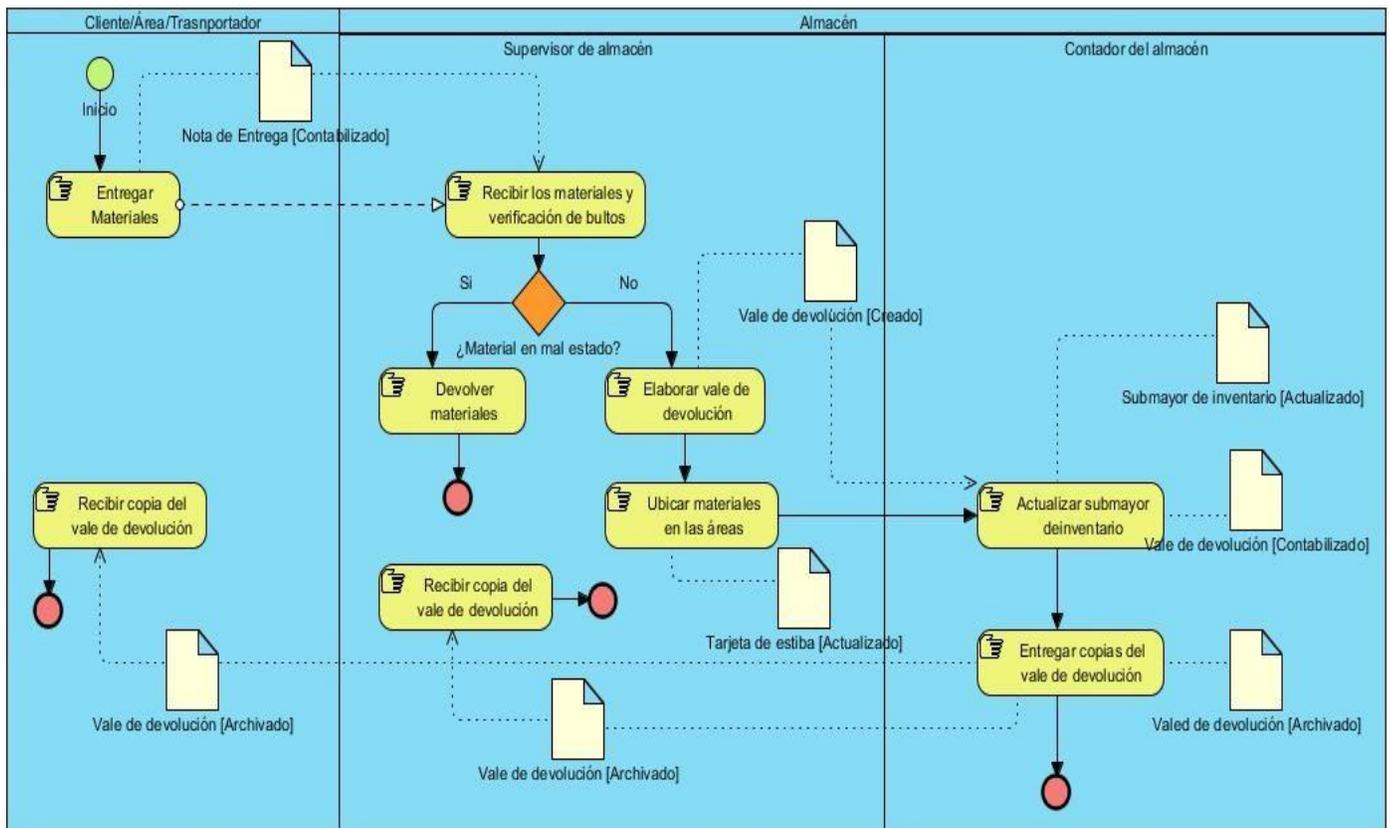


Figura 3 Proceso de negocio: Devolución de materiales.

2.2.3 Descripción del proceso del negocio

A continuación se muestra la descripción del proceso de devolución de materiales al almacén por donde fueron despachados, donde se podrá explicar detalladamente en qué consiste el proceso.

Objetivo	Devolver al almacén distribuidor los materiales que fueron despachados por el mismo.
Evento(s) que lo genera(n)	El cliente, área o transportador devuelve los materiales.
Pre condiciones	Se cuenta con el vale de entrega que ampara la devolución.
Marco legal	
Clientes internos	Supervisor de almacén Contador del almacén (Supervisor administrativo)
Clientes externos	Cliente/Área/Transportador
Entradas	Nota de entrega (XML)
Flujo de eventos	
Flujo básico Devolución de materiales	
1.	Devolver materiales. El proceso se inicia cuando el cliente o conductor se dirige al almacén para devolver materiales que fueron entregados mediante un vale de entrega. Para realizarse este proceso es preciso que conste el vale de entrega que amparó la entrega del material que se desea devolver.
2.	Recibir los materiales y verificación de bultos. El supervisor del almacén recibe los materiales verificando el estado de los bultos y que la cantidad de materiales devueltos sea menor o igual que la cantidad de materiales que refleja el vale de entrega que genera la devolución de materiales. La cantidad de materiales que se desean devolver debe ser menor o igual que la cantidad reflejada el vale de entrega. (Una copia del vale de entrega debe portarla el supervisor del almacén).
3.	Elaborar Vale de devolución. Si los materiales verificados se encuentran en buen estado, el supervisor del almacén procede a elaborar el Vale de devolución.

4. Ubicar materiales en las áreas. El supervisor de encarga de ubicar en las áreas de almacenamiento los materiales devueltos, actualizando la tarjeta de estiba. La Tarjeta de estiba del producto debe ser actualizada cuando se realiza algún movimiento de entrada o salida del producto en el almacén.
5. Actualizar submayor de inventario. El supervisor del almacén le entrega al Contador del almacén (Supervisor administrativo) el Vale de devolución para actualizar el submayor de inventario. El contador del almacén, consulta el Vale de devolución y se encarga de actualizar los saldos en el Submayor de inventario.
6. Entregar copias del Vale de devolución: El Contador del almacén es responsable de plasmar en el Vale de devolución las firmas del Supervisor del almacén y Cliente (o Conductor) formalizando la devolución de materiales; entregado copias del documento al Supervisor del almacén para su archivo y a la entidad que devuelve los materiales, como constancia de la devolución.
7. Concluye el proceso.

Post-condiciones

1. Se ha registrado un nuevo Vale de devolución
2. Se ha ingresado materiales al almacén, actualizado la Tarjeta de estiba
3. Se ha actualizado el Submayor de inventario.

Salidas

1. Vale de devolución (XML)
2. Tarjeta de estiba (formato duro)
3. Submayor de inventario (XML)

Flujos paralelos

N/A

Post-condiciones

N/A

Salidas

N/A

Flujos alternos

3.a Existen bultos en mal estado

1. Devolver materiales. El supervisor se encarga de devolver los materiales que se encuentran en mal estado.
2. Concluye el proceso.

Post-condiciones

N/A

Salidas

N/A

Asuntos pendientes

N/A

Tabla 4 Descripción del Proceso de negocio: Devolución de materiales.

2.2.4 Modelo conceptual

El Modelo Conceptual permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del problema.



Figura 4 Modelo Conceptual: Devolución de materiales

2.3 Ingeniería de requisitos asociado al proceso Vale de devolución

Al definir los requisitos de software se especifican las condiciones o capacidades que el sistema debe cumplir y las restricciones bajo las cuales debe operar, logrando un entendimiento entre el equipo de desarrollo y el cliente y especificando las necesidades reales de forma que satisfaga sus expectativas. Al definir claramente los requisitos funcionales que debe tener el sistema, se realizará una descripción detallada de cada uno de ellos de forma que los clientes y usuarios los entiendan correctamente.

2.3.1 Requisitos funcionales

Los requisitos funcionales deben ser completos, claros y concisos, además pueden definirse a partir de reglas del negocio o la propia interacción de los usuarios (39). La técnica de captura de requisitos utilizada fue la entrevista del grupo de desarrollo con el cliente. Partiendo del proceso de negocio anteriormente analizado se arribó a la conclusión de que existen 8 agrupaciones de requisitos para un total de 21 requisitos funcionales.

- RF-1 Gestionar Vales de entrega contabilizados.
 - RF-1.1 Listar Vales de entrega contabilizados.
 - RF-1.2 Buscar Vales de entrega contabilizados.
- RF-2 Gestionar productos del Vale de entrega contabilizado.
 - RF-2.1 Listar productos del Vale de entrega contabilizado.
 - RF-2.2 Buscar producto del Vale de entrega contabilizado.
- RF-3 Gestionar Vale de devolución.
 - RF-3.1 Adicionar Vale de devolución.
 - RF-3.2 Modificar Vale de devolución.
 - RF-3.3 Eliminar Vale de devolución.
 - RF-3.4 Listar Vales de devolución.
 - RF-3.5 Consultar Vale de devolución.
 - RF-3.6 Buscar Vale de devolución.
 - RF-3.7 Imprimir Vale de devolución.
 - RF-3.8 Imprimir listado de Vales de devolución.
- RF-4 Gestionar producto del Vale de devolución.
 - RF-4.1 Adicionar producto al Vale de devolución.

- RF-4.2 Modificar producto del Vale de devolución.
- RF-4.3 Eliminar producto del Vale de devolución.
- RF-4.4 Listar productos del Vale de devolución.
- RF-4.5 Buscar producto del Vale de devolución.
- RF-5 Confirmar Vale de devolución.
- RF-6 Cancelar estado del Vale de devolución.
- RF-7 Contabilizar el Vale de devolución.
- RF-8 Cancelar el Vale de devolución.

Precondiciones	El usuario ha sido validado. Debe existir al menos un vale de entrega registrado en el sistema con estado de “Contabilizado”.
Flujo de eventos	
Flujo básico Adicionar Vale de devolución	
1.	Se selecciona el vale de entrega a devolver.
2.	Se cargan los siguientes datos del documento: nombre de la entidad nombre del almacén número del documento fecha de emisión del documento estado del documento
3.	El sistema muestra los datos del documento origen de la devolución: tipo de documento número de documento fecha del documento nombre de la entidad cliente
4.	Se introducen los siguientes datos del Vale de devolución: observaciones
5.	El sistema valida (ver validación 1) los datos introducidos.
6.	Si los datos son correctos el sistema los registra.
7.	El sistema confirma el registro de los datos.
8.	Concluye el requisito.
Post-condiciones	
4.	Se registró en el sistema un nuevo Vale de devolución.
Flujos alternativos	

Flujo alternativo 7.aInformación errónea		
1		El sistema señala los datos erróneos y permite corregirlos.
2		El usuario corrige los datos.
3		Volver al paso 6 del flujo básico.
Pos-condiciones		
1		N/A
Flujo alternativo 7.bInformación incompleta		
1		El sistema señala los datos vacíos y permite corregirlos.
2		El usuario corrige los datos.
3		Volver al paso 6 del flujo básico.
Pos-condiciones		
1		N/A
Flujo alternativo *.a El usuario cancela la acción		
1		Concluye el requisito.
Post-condiciones		
1		No se registran los datos.
Validaciones		
1		Se validan los datos según lo establecido en el Modelo conceptual <<Referencia al modelo conceptual en cuestión>>.
Relaciones	Requisitos Incluidos	Se selecciona el vale de entrega: Listar notas de contabilizadas; en la agrupación Listas notas de entrega contabilizadas.
	Extensiones	N/A
Conceptos	Vale de devolución	Visibles en la interfaz: nombre del almacén nombre de la entidad número del documento estado del documento fecha de emisión observaciones Utilizados internamente: código entidad código almacén nombre del documento

Documento	Visibles en la interfaz: número de documento fecha de emisión nombre de entidad nombre de almacén observaciones estado del documento Utilizados internamente: código entidad código almacén nombre del documento creado por año aprobado por importe total
Vale de entrega	Visibles en la interfaz: número del documento estado del documento fecha de emisión Utilizados internamente: código entidad código almacén nombre del documento
Cliente	Visibles en la interfaz: nombre del cliente Utilizados internamente: código cliente
Requisitos especiales	Son los requisitos no funcionales específicos para el requisito.
Asuntos pendientes	Posibles mejoras al requisito.

Tabla 5 Descripción del Requisito Adicionar Vale de Devolución.

2.3.2 Prototipo de interfaz de usuario principal

A continuación se muestra el prototipo de interfaz de usuario funcional del requisito gestionar Vale de devolución. El resto de los prototipos de interfaz de usuario se pueden encontrar en los Anexos del documento.

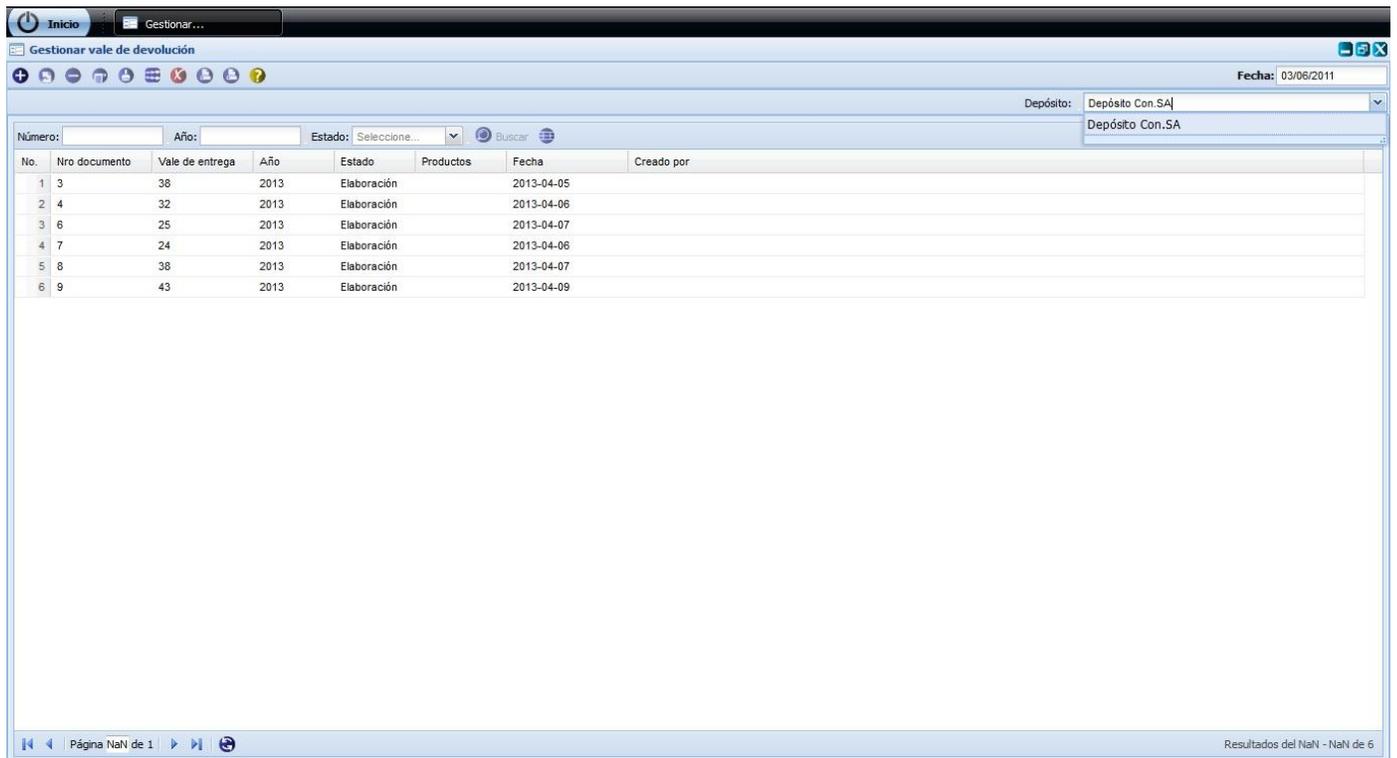


Figura 5 Prototipo de interfaz de usuario.

2.4 Diseño de la solución

El diseño del software se encuentra en el núcleo técnico de la ingeniería de software. Una vez que se analizan y especifican los requisitos del producto, generación de código y pruebas que se requieren para construir y verificar el software. El diseño tiene como objetivo determinar una solución a los requisitos del sistema definidos anteriormente. Entre los requisitos identificados el requisito funcional gestionar Vales de devolución es el más importante por su complejidad, por tanto será el requisito ejemplo para mostrar los artefactos obtenidos durante todo el ciclo de vida del software.

2.4.1 Diagrama de clase del diseño

El Diagrama de Clases de Diseño describe gráficamente las especificaciones de las clases de software y las interfaces en una aplicación. Contiene las definiciones de las entidades de software, en vez de conceptos del mundo real. Este uno de los diagramas principales para el análisis y diseño de un software ya que sirve además para visualizar las distintas relaciones estructurales. El diagrama que a continuación se presenta, muestra la estructura de las clases que servirán de apoyo a la hora de la implementación. En

él se puede apreciar como el archivo `gestvaledevolucion.phtml` contiene los archivos correspondientes a la librería *ExtJS* la cual se encarga de construir las interfaces del componente. Además presenta un formulario que es el responsable de enviar los datos entrados por el usuario hacia el servidor, donde al instanciar el controlador frontal de *Zend*, dichos datos serán enviados a la controladora correspondiente a la gestión de los Vales de devolución. Se aprecian también las clases necesarias para acceder a la información almacenada a nivel de datos.

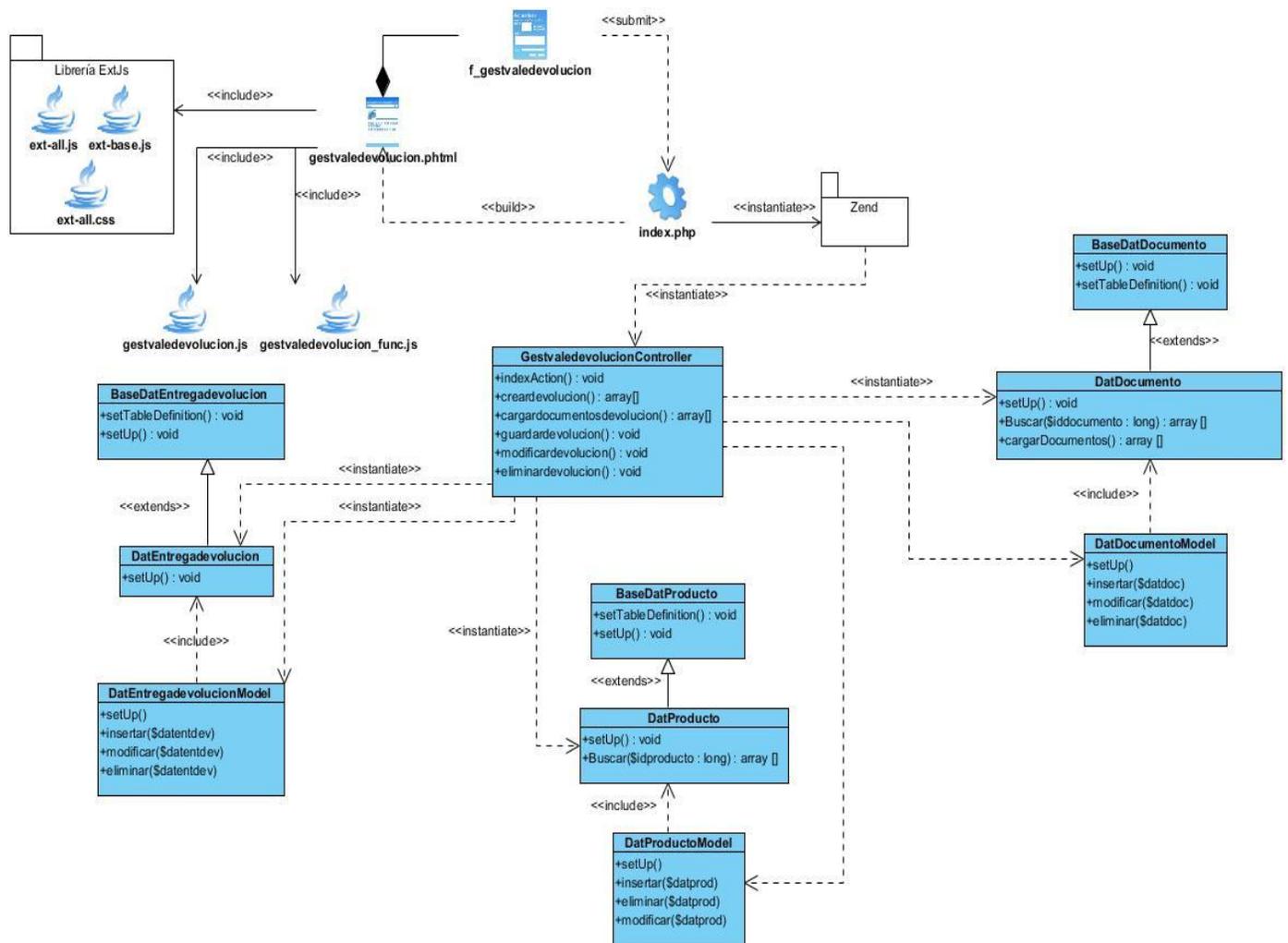


Figura 6 Diagrama de clases del diseño

Patrones de diseños utilizados

El diseño propuesto fue creado siguiendo patrones que de manera general constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos, lo cual permitirá llevar una mejor implementación de la gestión de Vales de Devolución.

Experto: Dicho patrón es evidenciado en la definición de las clases controladoras. Plantea la asignación de responsabilidades al experto en información, es decir, la clase que cuenta con la información necesaria para cumplir las funcionalidades que deben realizarse a partir de la información manejada dentro del componente. Ejemplo de esto es la clase GestvaledevolucionController encargada de realizar la gestión de los Vales de devolución.



Figura 7 Evidencia del Patrón Experto

Creador: El patrón creador es evidenciado en la mayoría de las clases del marco de trabajo ya que varias de las clases tienen la responsabilidad de crear instancias de otras, varias clases son creadoras de los objetos de otras. Ejemplo de esto son las clases del paquete Domain, quienes son las encargadas de crear los objetos de tipo Doctrine_Query.

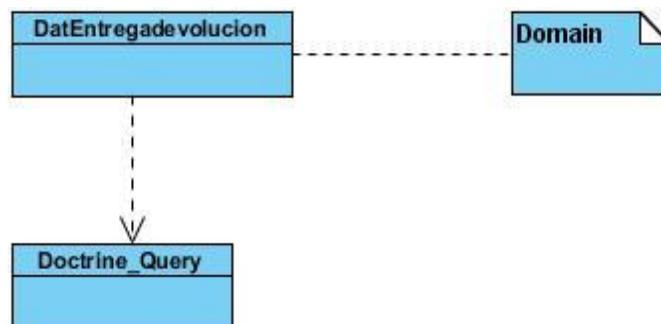


Figura 8 Evidencia del Patrón Creador

Controlador: Este patrón plantea asignar la responsabilidad del manejo de un mensaje de los eventos de un sistema a una clase.

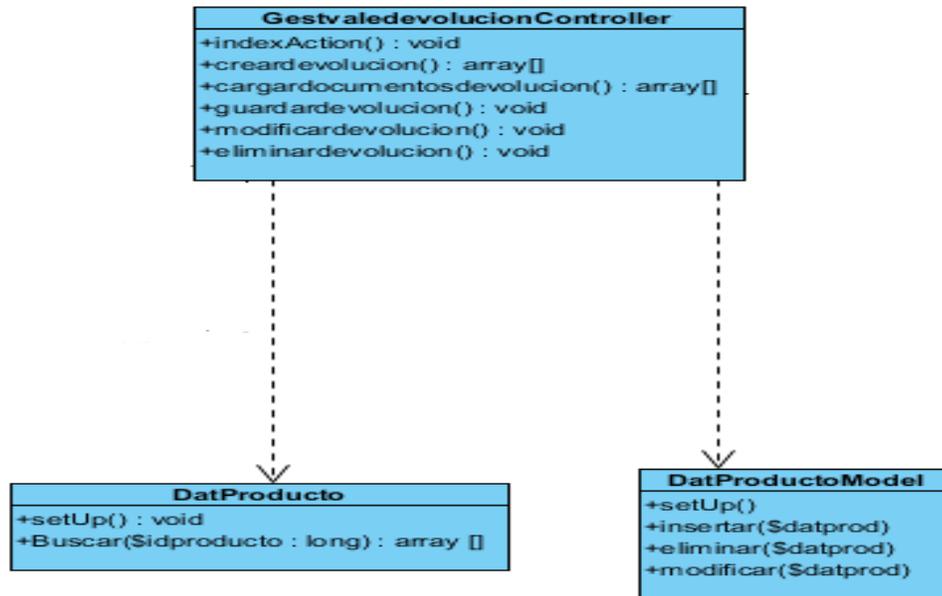


Figura 9 Evidencia del Patrón Controlador

2.4.2 Diagrama de interacción

Luego de lo anterior, se realizaron diagramas de secuencia, que son un ejemplo de diagramas de iteración que muestran la secuencia de la lógica y en el orden en que se suceden los mensajes. A continuación se presenta el diagrama de secuencia asociado al escenario gestionar Vales de devolución.

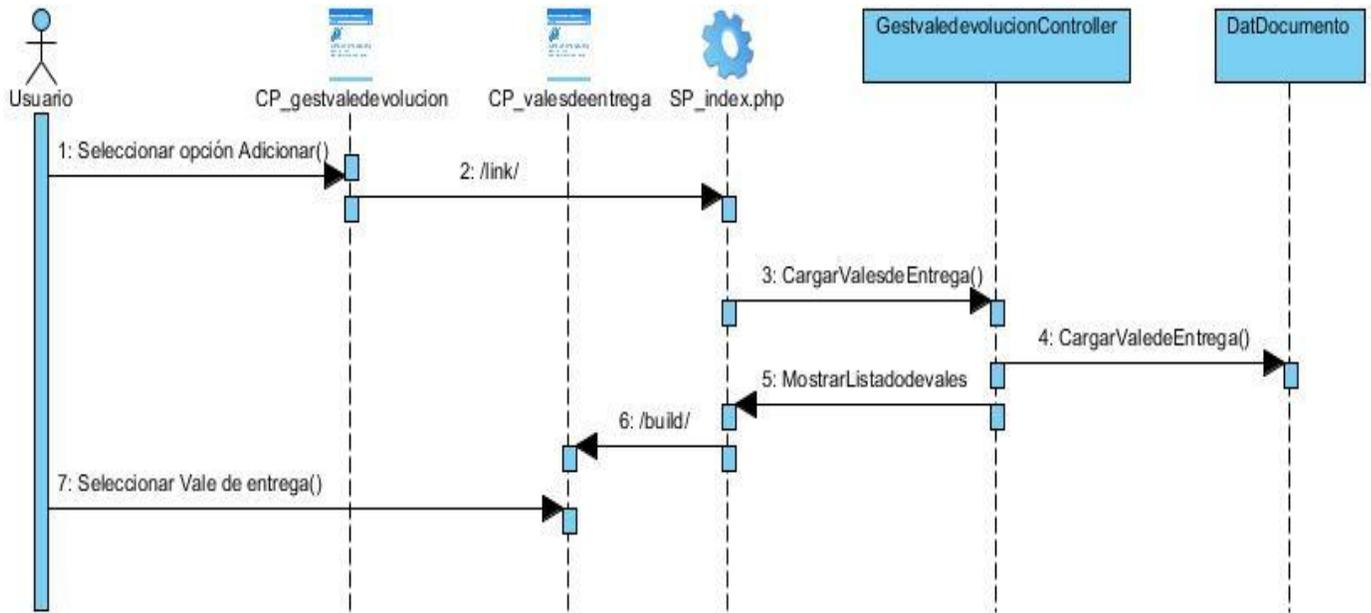


Figura 10 Diagrama de clases del diseño.

2.4.3 Modelo de datos

La base de datos que se muestra a continuación es un componente de CedruX y se le agregó una nueva tabla: mod_inventario.dat_movimiento con el objetivo de almacenar los datos correspondientes a la relación entre los vales de entrega y los Vales de devolución.

2.5 Implementación de la solución

Describe cómo los elementos del modelo del diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. En ella se toma como punto de partida lo arrojado como resultado en el diseño y se implementa el sistema en términos de componentes como ficheros de código binario, código fuente, scripts, ejecutables, entre otros.

2.5.1 Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones, contienen todos los componentes definidos en el subsistema, así como las distintas dependencias existentes entre ellos. Se realizan con el objetivo de obtener una vista general del sistema a partir de las dependencias e integraciones de los componentes.

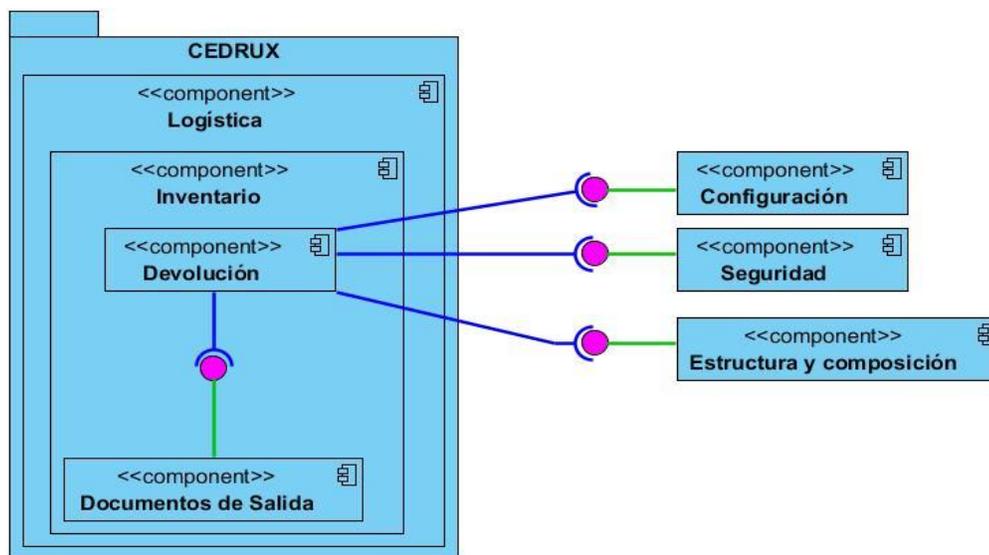


Figura 12 Diagrama de componente.

Componente Devolución: Es el encargado de realizar la gestión de los Vales de devolución. Recibe la estructura común del componente Estructura y composición, además consume los servicios de autenticación del componente Seguridad. Por otra parte recibe la fecha, el período y el ejercicio contable del componente de Configuración General.

2.6 Conclusiones parciales

En el desarrollo del capítulo se obtuvo los siguientes resultados:

- Se definieron todos los requisitos del sistema, mediante los cuales se identificaron todas las funcionalidades requeridas por el proceso de Gestión de Vales de devolución.
- Quedó definida la propuesta de solución, además se obtuvo el mapa de procesos del negocio y diagrama de procesos, que posibilitan comprender como funcionan los procesos de las entidades que serán necesarios informatizar.
- Se diseñó el diagrama de clases, lo cual permitió conocer la estructura y las relaciones de las clases que se manejan en el componente, a partir del cual se realizaron los prototipos de interfaz de usuario funcional.
- Se definieron de los prototipos de interfaz de usuario los cuales permitieron validar los requisitos establecidos para el componente que se será implementado.
- Se logró conocer la ubicación física del sistema en términos de componentes y nodos. Además de obtener la estructura que presenta el componente Devolución.
- Se logró desarrollar el componente Devolución a partir de la arquitectura definida.

Capítulo 3: VALIDACIÓN PROPUESTA DE SOLUCIÓN

3.1 Introducción

En el presente capítulo se efectúan las validaciones de los requisitos y del diseño, seguido de las pruebas del sistema partiendo de los resultados obtenidos en el diseño. Se aplicarán pruebas de caja blanca y caja negra para la validación de la implementación, las cuales permitirán probar las funcionalidades del componente para detectar posibles errores.

3.2 Validación del modelo de negocio e ingeniería de requisitos

Validación del negocio

Una vez que el proceso ha sido descrito, se realiza la validación del mismo, comprobando que la identificación de este se desarrolló correctamente y que se entiende con claridad lo que se desea informatizar. En nuevas reuniones con el cliente se verifica que la modelación del proceso se corresponde al desarrollo de dicha actividad en las entidades cubanas. El cliente al aprobar las descripciones de los modelos, las mismas son revisadas por especialistas de calidad, donde se verifica el cumplimiento de los estándares establecidos para la descripción, logrando que esta sea entendible para los implicados.

Validación de los requisitos

La validación de los requisitos se realiza mediante las técnicas mencionadas en el Capítulo 1. La técnica Prototipo de interfaz de usuario fue muy utilizada, debido a que en los prototipos se engloban las funcionalidades del sistema y el cliente pudo apreciar una aproximación al producto. La Revisión técnica formal a los documentos y artefactos por parte del cliente y el equipo de desarrollo fue una de las técnicas utilizadas en la validación, con lo que se pudo comprobar que la información brindada por el cliente permitió realizar una correcta especificación de los requisitos que el componente debe presentar.

3.3 Validación del diseño propuesto

El objetivo de usar un conjunto de métricas de software orientadas a determinar, entre otros aspectos, qué características del modelo de diseño se pueden estimar para comprobar que el sistema será fácil de implementar en cuanto a organización. Se seleccionaron las métricas **Tamaño Operacional de Clase y Relaciones entre Clases** para validar el diseño del componente Devolución.

Tamaño Operacional de Clase

Esta métrica se determina por el número total de operaciones que están encapsuladas dentro de la clase. Grandes valores de esta medida muestran que la clase puede tener demasiada responsabilidad, lo cual reducirá la reusabilidad de la misma y complicará su implementación. Por otro lado, en cuanto menor sea el valor medio para el tamaño más probable es que las clases tengan menos responsabilidad y complejidad y más nivel de reutilización.

Los valores de umbrales tomados en cuenta para evaluar el diseño propuesto son los siguientes:

- Responsabilidad de las clases.
 - Baja (\leq Promedio) = 8
 - Media (Entre Promedio y $2 \times$ Promedio) = 3
 - Alta ($> 2 \times$ Promedio) = 1
- Complejidad de implementación de las clases.
 - Baja (\leq Promedio) = 8
 - Media (Entre Promedio y $2 \times$ Promedio) = 3
 - Alta ($> 2 \times$ Promedio) = 1
- Reutilización de las clases.
 - Baja ($> 2 \times$ Promedio) = 1
 - Media (Entre Promedio y $2 \times$ Promedio) = 3
 - Alta (\leq Promedio) = 8

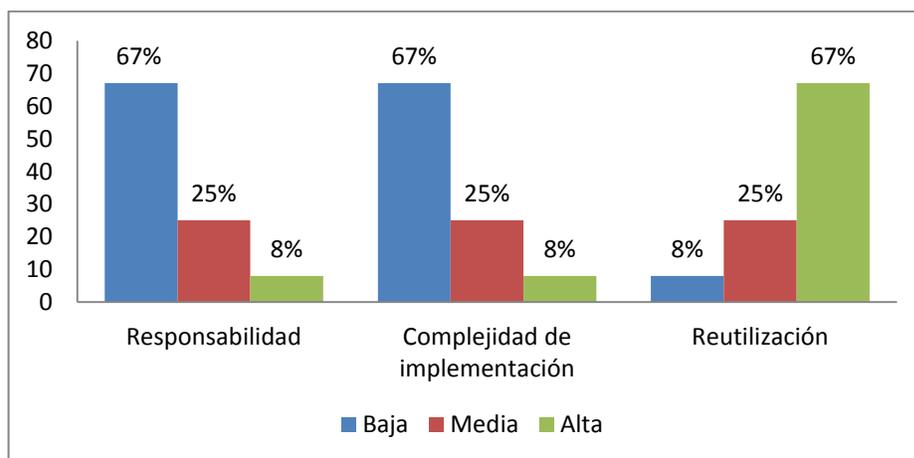


Figura 13 Gráfica tamaño operacional de clase.

La aplicación de esta métrica demuestra que las clases no se encuentran muy sobrecargadas en responsabilidad, presentan una baja complejidad de implementación y presentan un alto nivel de reutilización.

Relaciones entre Clases

Esta métrica se determina por la cantidad de relaciones existentes entre las clases contenidas en el diseño. El número de dependencias es directamente proporcional al nivel de acoplamiento, a la complejidad del mantenimiento y a la cantidad de pruebas a realizar sobre las clases y es inversamente proporcional al grado de reutilización de las mismas.

Los valores de umbrales para dicha métrica son los siguientes:

- Acoplamiento
 - Ninguno $0 = 5$
 - Baja $1 = 2$
 - Media $2 = 5$
 - Alta $> 2 = 0$

- Complejidad mantenimiento
 - Baja (\leq Promedio) $= 7$
 - Media (Entre Promedio y $2 * \text{Promedio}$) $= 5$
 - Alta ($> 2 * \text{Promedio}$) $= 0$

- Reutilización
 - Baja ($> 2 * \text{Promedio}$) $= 0$
 - Media (Entre Promedio y $2 * \text{Promedio}$) $= 5$
 - Alta (\leq Promedio) $= 7$

- Cantidad de Pruebas
 - Baja (\leq Promedio) $= 7$
 - Media (Entre Promedio y $2 * \text{Promedio}$) $= 5$
 - Alta ($> 2 * \text{Promedio}$) $= 0$

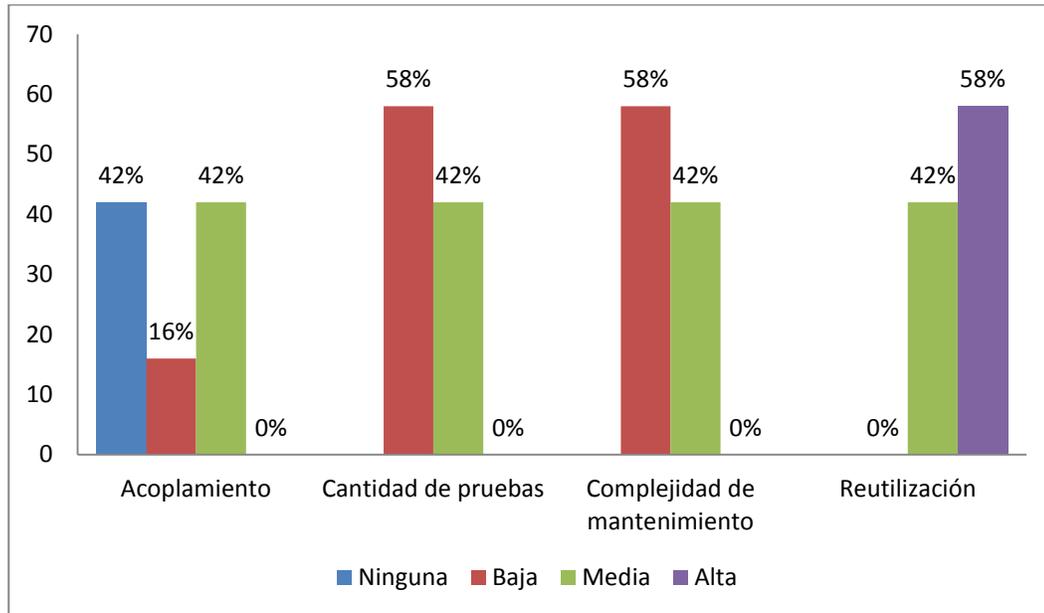


Figura 14 Gráfica relación entre clase.

Con los resultados obtenidos acerca de la evaluación de la métrica RC se puede deducir que existe además bajo acoplamiento entre las clases, lo cual implica que será menor la influencia de los cambios que se realicen en otras clases, además presentan un alto nivel de reutilización. Indican además que el diseño no es complejo, pues la complejidad de mantenimiento es baja, así como la complejidad en las pruebas.

Matriz de inferencia de indicadores de calidad

La matriz de inferencia permite evaluar positiva o negativamente los resultados obtenidos de las relaciones atributo/métrica en una escala numérica, donde el valor 1 representa un resultado positivo y el valor 0 negativo. Si la métrica no influye en el atributo de calidad la relación es considerada como nula y es representada con un guión simple (-). Al promediar por cada atributo las relaciones no nulas, se obtiene un resultado general que al ubicarse en el rango de evaluación (valor ≤ 0.4 : “Mala”, (valor > 0.4 y valor < 0.7 : “Regular”, valor ≥ 0.7 : “Buena”) permite arribar a conclusiones sobre la calidad del diseño propuesto. Teniendo en cuenta que los resultados arrojados con la aplicación de las métricas fueron positivos para todos los atributos, la matriz de la inferencia queda elaborada de la siguiente manera:

Atributos/Métricas	TOC	RC	Promedio
--------------------	-----	----	----------

Responsabilidad	1	(-)	1
Complejidad de implementación	1	(-)	1
Reutilización	1	1	1
Acoplamiento	(-)	1	1
Complejidad de Mantenimiento	(-)	1	1
Cantidad de pruebas	(-)	1	1

Tabla 7 Resultados de la evaluación de la relación atributo/relación.

El promedio general es 1 y teniendo en cuenta el rango de evaluación se concluye que el componente Devoluciones presenta un diseño con buena calidad.

3.4 Validación de la aplicación

El desarrollo de software ha de ir acompañado de una actividad que garantice la calidad. La prueba del software es un elemento crítico para la garantía de calidad del software y representa una revisión final de las especificaciones, del diseño y de la codificación.

3.4.1 Prueba de Caja Blanca

Las pruebas de software no garantizan que un sistema esté libre de errores, sino que se detecten la mayor cantidad de defectos posibles para su debida corrección. Para realizar la prueba del Camino básico es preciso calcular la complejidad ciclomática del algoritmo o fragmento de código a analizar. A continuación se muestra el código del método creardevolucionAction encargado de crear un Vale de devolución a partir de un vale de entrega.

```
public function creardevolucionAction() {
    $iddocumento_ent = $this->_request->getPost('iddocumento_ent'); //1
    $iddocumento_dev = $this->_request->getPost('iddocumento_dev'); //1
    $accion = $this->_request->getPost('accion'); //1
    $iddoc = '9000082'; //1
    $idestructuracomun = $this->_request->getPost('idlugar'); //1

    if ($accion == 'mod'){ //2
        $datos = $this->pIntegrator->documentos->CargarDatosVE($idestructuracomun,$iddocumento_ent , $iddocumento_dev, $iddoc, $accion); //3
        $dev = new DatEntregadevolucion(); //3
        $result = $dev->dameDocumentoXid($iddocumento_ent); //3
        $datosclientes = $this->comun->dameNombreClienteOCentroCosto($result[0]['idcliente'], '9000008'); //3
        $datos->cliente = $datosclientes['denomcliente']; //3
    }
    else
    {
        $dat = $this->pIntegrator->documentos->CargarDatosEncabezado($idestructuracomun, $iddoc); //4
        $datos['entidad'] = $dat->entidad; //4
        $datos['area'] = $dat->area; //4
        $datos['fecha'] = $dat->fecha; //4
        $datos['estado'] = $dat->estado; //4
        $datos['nrodoc'] = $dat->nrodoc; //4
        $datos['almacen'] = $dat->almacen; //4
        $datos['especialidad'] = $dat->especialidad; //4
        $datos['idtipooperacion'] = '90000000004'; //4
    }
    //5
    echo(json_encode($datos)); //6
} //7
```

Figura 15 Método creardevolucionAction.

Para el cálculo de la complejidad ciclomática es necesario representar el grafo de flujo asociado al código antes presentado a través de nodos, aristas y regiones, quedando como se muestra a continuación.

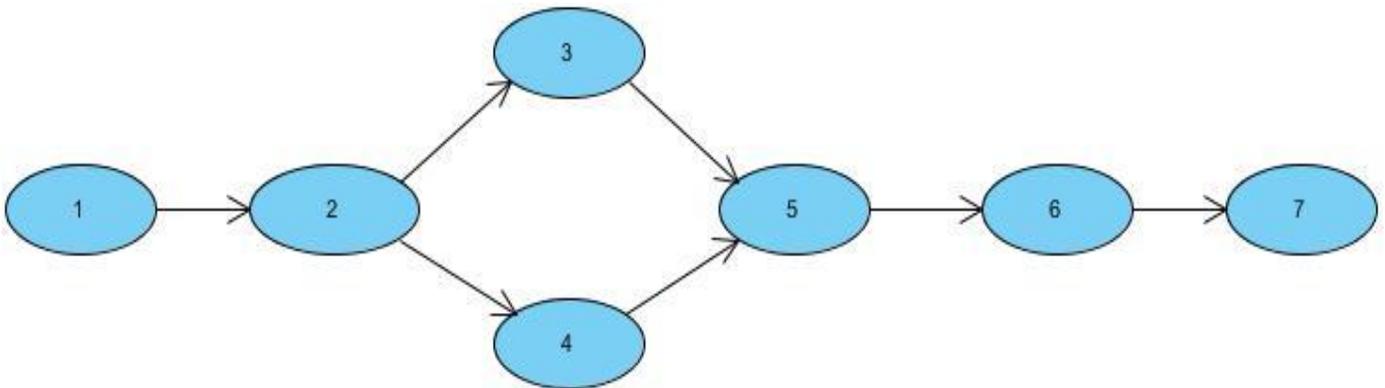


Figura 16 Grafo de flujo asociado al método

Una vez construido el grafo de flujo asociado al procedimiento anterior se determina la complejidad ciclomática, el cálculo es necesario efectuarlo mediante tres vías o fórmulas, se debe utilizar el mismo grafo en cada caso:

Fórmula 1 . $V(G) = (A - N) + 2$

- Siendo "A" la cantidad total de aristas y "N" la cantidad total de nodos.

Resultado.

$$V(G) = (7 - 7) + 2$$

$$V(G) = 2$$

Fórmula 2 . $V(G) = P + 1$

- Siendo "P" la cantidad total de nodos predicados (son los nodos de los cuales parten dos o más aristas).

Resultado.

$$V(G) = 1 + 1$$

$$V(G) = 2$$

Fórmula 3 . $V(G) = R$

- Siendo "R" la cantidad total de regiones, se incluye el área exterior del grafo, contando como una región más.

Resultado.

$$V(G) = 2$$

Una vez realizado los cálculos se puede apreciar que arrojan el mismo resultado por lo que se puede decir que la complejidad ciclomática del código es de 2, significando esto que existen dos posibles caminos por donde puede circular el flujo, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado.

Seguidamente es necesario especificar los caminos básicos que puede tomar el algoritmo durante su ejecución. En estas representaciones se subrayan los elementos de cada camino que los hacen independientes a los demás.

Camino básico # 1: 1, 2, 3, 5, 6, 7.

Camino básico # 2: 1, 2, 4, 5, 6, 7.

Después de haber extraído los caminos básicos del flujo, se procede a ejecutar los casos de pruebas para este procedimiento y se debe realizar al menos un caso de prueba por cada camino básico.

Camino básico # 1: 1, 2, 3, 5, 6, 7.	
Descripción	Se crea un Vale de devolución a partir de la selección de un vale de entrega.
Condición de ejecución	Debe existir al menos un vale de entrega en estado contabilizado.
Entrada	<code>\$iddocumento_ent = \$this->_request->getPost('iddocumento_ent');</code> <code>\$iddocumento_dev = \$this->_request->getPost('iddocumento_dev');</code> <code>\$accion = \$this->_request->getPost('accion');</code> <code>\$iddoc = '9000082';</code> <code>\$idestructuracomun = \$this->_request->getPost('idlugar');</code>
Resultados esperados	Se debe de crear un Vale de devolución.
Resultados	Se crea un Vale de devolución.
Salida	Renderizar texto.

Tabla 8 Caso de prueba para el camino básico 1.

Camino básico # 1: 1, 2, 4, 5, 6, 7.	
Descripción	Se modifica el Vale de devolución.
Condición de ejecución	Seleccionar un vale de devolución.
Entrada	<code>\$iddocumento_ent = \$this->_request->getPost('iddocumento_ent');</code> <code>\$iddocumento_dev = \$this->_request->getPost('iddocumento_dev');</code> <code>\$accion = \$this->_request->getPost('accion');</code> <code>\$iddoc = '9000082';</code> <code>\$idestructuracomun = \$this->_request->getPost('idlugar');</code>
Resultados esperados	Se debe de modificar un Vale de devolución.
Resultados	Se modifica un Vale de devolución.
Salida	Rende-rizar texto.

Tabla 9 Caso de prueba para el camino básico 2.

Con la realización de las pruebas de caja blanca a diferentes métodos significativos de la aplicación, fue posible apreciar como estos respondían satisfactoriamente a los resultados que se esperaban por cada caso de prueba. Por lo que se puede observar que los códigos implementados realizan correctamente las funciones que le han sido asignadas, con el objetivo de asegurar el correcto funcionamiento de las interfaces, o flujo de datos entre componentes. Además se logra como resultado que disminuya en un gran porcentaje el número de errores existentes en el sistema y por ende una mayor calidad.

3.5 Verificación y validación funcional

Se aplicará el método de caja negra para probar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene.

3.5.1 Pruebas de caja negra

Las pruebas se aplican sobre el sistema empleando un determinado conjunto de datos de entrada y observando las salidas que se producen para determinar si la función se está desempeñando correctamente por el sistema bajo prueba. Realizar pruebas de forma que se compruebe que cada función es operativa. En la prueba de la caja negra, los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta.

Para el correcto desarrollo de las pruebas se hizo uso de las técnicas: Partición de equivalencia y Análisis de valores límites, descritas en el capítulo 1.3 en la fase de pruebas internas.

3.5.1.1 Aplicación de las pruebas de caja negra en el sistema

Para la realización de las pruebas de caja negra se empleó la técnica Partición de Equivalencia que representa una de las más efectivas, permite examinar los valores válidos e inválidos de las entradas existentes en el software, descubre de forma inmediata una clase de errores que de otro modo requerirían la ejecución de muchos casos antes de detectar el error genérico. Dirige la definición de casos de pruebas que descubran clases de errores, reduciendo el número de clases de prueba que hay que desarrollar.

3.5.2 Diseño de los casos de pruebas

Para verificar que la aplicación se comporta según los requerimientos establecidos por el cliente, se diseñan casos de pruebas usando el método de Caja Negra. A continuación se representa el diseño de caso de prueba para el requisito funcional Adicionar Vale de devolución.

3.5.2.1 Diseño de caso de prueba Requisito Funcional. Adicionar vale de devolución

Durante las pruebas que se realizaron de calidad se encontraron en la primera iteración 5 No Conformidades (NC) corrigiéndose en su totalidad en la segunda iteración.

Condiciones de ejecución

- Se debe identificar y autenticar ante el sistema y además debe tener los permisos para ejecutar esta acción.
- Se debe seleccionar el subsistema de Logística.
- Se debe seleccionar la opción **Inventario/Devolución/Gestionar vale de devolución.**

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
1: Adicionar vale de devolución.	El sistema debe permitir generar un vale de devolución a partir de un vale de entrega.	EP 1.1: Adicionar vale de devolución a partir de un vale de entrega.	<ul style="list-style-type: none">– Se selecciona un vale de entrega que genere vale de devolución.– Se presiona el botón Aceptar.– Se muestra una interfaz con los datos referentes al vale de entrega para confirmarlos.– Se presiona el botón Aceptar.– Se muestra un mensaje de confirmación, del documento creado

Tabla 10 Requisitos a probar.

No	Nombre de campo	Tipo	Válido	Inválido	Inválido	Inválido	Inválido
N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Tabla 11 Descripción de variable.

Id del escenario	Escenario	Respuesta del sistema	Resultado de la prueba
EP 1.1	Adicionar vale de devolución a partir de un vale de entrega.	El sistema genera el documento y muestra el mensaje de confirmación: "Se ha generado el vale de devolución."	

Tabla 12 Juegos de datos a probar.

3.5.3 Análisis de la satisfacción del cliente

A partir de las pruebas anteriormente realizadas en cada una de las etapas del desarrollo del componente Gestión de Vales de devolución se puede afirmar que el mismo está perfectamente dispuesto para llevar a cabo la misión que tiene encomendada, de esta manera queda satisfecho el cliente con la solución. Con la puesta en práctica de la versión del subsistema de inventario que contiene la solución que aquí se plantea, las empresas que lo utilicen podrán contar con una mayor organización de la información generada por la entrada de mercancías al almacén a partir de una devolución.

3.6 Conclusiones

Al finalizar el presente capítulo se puede apreciar que:

- La buena calidad del diseño propuesto, comprobada finalmente a través de la aplicación de dos de las métricas de validación de diseño, sentó de manera eficiente las bases para la implementación de las funcionalidades.

- La utilización de las técnicas definidas para la validación de requisitos, demostró que estos presentan las condiciones requeridas, y permitió continuar con el diseño e implementación de los mismos.
- Al aplicar las pruebas de caja negra se obtuvieron importantes resultados sobre el comportamiento del sistema. Dichas pruebas permitieron comprobar los requisitos funcionales definidos para el componente a partir de los diseños de casos de pruebas, evaluándose la calidad del componente desarrollado, lo cual constituyó un elemento clave del capítulo que recién concluye.

CONCLUSIONES

Una vez finalizado el presente trabajo se puede concluir que se desarrollaron todas las tareas a fin de cumplir los objetivos propuestos, resaltando que:

- Con el estudio de sistemas que realizan la gestión de Vales de devolución se obtuvo información referente a dicho proceso, la cual sirvió de ayuda para la implementación del componente como propuesta de solución.
- La generación de los artefactos durante el desarrollo de los flujos de análisis, diseño e implementación proporcionó la obtención del componente Devolución para la correcta integración del mismo al sistema CedruX.
- La validación del diseño mediante la aplicación de las métricas, las pruebas realizadas, demostró que la solución cumple con los requerimientos, la eficiencia y la estabilidad necesaria para ser desplegado en las entidades cubanas.
- Las pruebas de caja negra permitieron detectar y corregir los errores no detectados durante la implementación posibilitando cumplir con las especificaciones requeridas y la validación del componente implementado.

De manera general se solucionó el problema existente a través de la realización de todas las tareas definidas, obteniéndose como resultado el componente Devolución, que permitirá, a través de sus funcionalidades mejorar la gestión de los Vales de devolución en el subsistema Inventario-CedruX.

RECOMENDACIONES

Atendiendo a las conclusiones a las que se arriban con el desarrollo del trabajo, se recomienda:

- Aprovechar los estudios realizados en esta investigación para mejorar la solución presentada en futuras versiones que se realicen del sistema.
- Ampliar las funcionalidades del módulo con los nuevos requerimientos que surjan por necesidades del cliente.
- Corregir las no conformidades detectadas en las pruebas pilotos que se le realicen en el despliegue del producto, con el objetivo de refinar la solución.

BIBLIOGRAFÍA

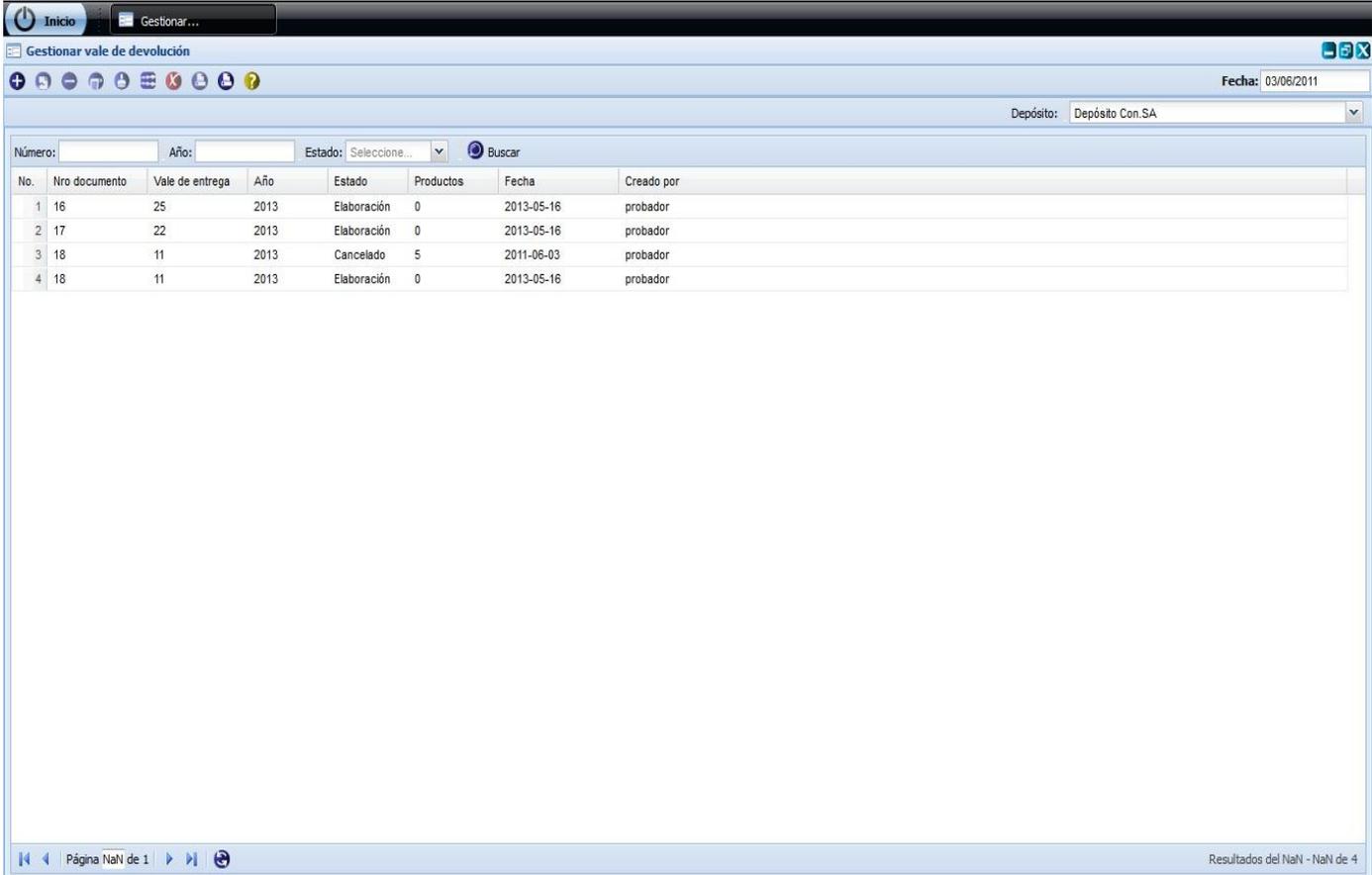
1. Grupo de Soporte CEZ s.a. [En línea] 2007. <http://www.cez.com.pe/Sistemas/ERP.html>.
2. Aner Sistemas Informáticos. [En línea] <http://www.aner.com/software-de-gestion-empresarial/que-es-un-erp.html>.
3. **Valda, Juan Carlos.** Grandes Pymes. [En línea] 15 de 12 de 2011. <http://jcvalda.wordpress.com/2011/12/15/erp-planificacion-de-recursos-empresariales/>.
4. **Entidades, Centro de Informatización de Gestión de.** *MODELO DE DESARROLLO DE SOFTWARE.* 2012.
5. THEFREEDICTIONARY. [En línea] 2012. <http://es.thefreedictionary.com/vale>.
6. Ayuntamiento de Madrid. [En línea] 2011.
<http://www.madrid.es/portales/munimadrid/es/Inicio/Ayuntamiento/Consumo-y-Comercio/Consumo/Diccionario-de-Consumo?vgnextfmt=default&vgnextoid=ab28d0f730fc8210VgnVCM1000000b205a0aRCRD&vgnextchannel=560b9ad016e07010VgnVCM100000dc0ca8c0RCRD&idioma=es&idiomaP>.
7. **Martínez, Ing. Saily Oliva.** *0128_Descripción de procesos de negocio_Devolución de materiales.* 2012.
8. **MECALUX.** *easy wms.* 2010.
9. ASINTEC. [En línea] <http://www.asintec.es/empresa/quienes-somos/quienes-somos>.
10. **domarix.com.** OpenERP SPAIN. [En línea] <http://www.openerpspain.com/areas-de-trabajo>.
11. OpenERPSpain. [En línea] <http://openerpspain.com/openerp/propiedades-de-openerp/>.
12. RODAS XXI Sistema Integral Económico Administrativo. [En línea] 2002-2013.
<http://www.rodasxxi.cu/rodasxxi.php>.
13. **Riveaux, Miguel Díaz.** *Desarrollo de un componente para la gestión de conceptos contables asociados a grupos y contenidos económicos en el subsistema de contabilidad general.* 2012.
14. Assets Sistema de gestión Integral. [En línea] 2004. <http://www.assets.co.cu/assets.asp>.
15. uni>ersia. [En línea] 2011. <http://desarrollo-profesional.universia.es/mercado-laboral/proceso-seleccion/entrevistas/objetivo-entrevista/>.
16. redTIC. [En línea] 2004. <http://redtic.educacontic.es/content/view/4124/356.html>.

17. **Carrillo, Adriana Milena Rangel.** *Análisis comparativo de técnicas de obtención de requerimientos para el módulo de facturación del aplicativo gestasoft hospitalario para imsalud.* 2007.
18. *Patrones de modelado de procesos.*
19. **Attribution Non-commercial.** CURSO 1: Desarrollo de Aplicaciones I. [En línea] 4 de 11 de 2011. [Citado el: 16 de 1 de 2013.] <http://es.scribd.com/doc/52731260/6/Definicion-de-UML>.
20. IPCORP. [En línea] 11 de 12 de 2008. <http://www.ipcorp.com.ar/blog/2008/12/11/osrmt-open-source-requirements-management-tool/>.
21. **monografias.com.** Herramientas CASE para BD. [En línea] [Citado el: 16 de 1 de 2013.] <http://www.monografias.com/trabajos24/herramientas-case/herramientas-case.shtml>.
22. **Martínez Rivera Daniela.** Herramienta Case Visual Paradigm. [En línea] 17 de 4 de 2012. [Citado el: 16 de 1 de 2013.] <http://dianbeel.blogspot.com/2012/06/segundo-trabajo-herramienta-case-visual.html>.
23. **Mestras, Juan Pavón.** *Estructura de las Aplicaciones Orientadas a Objetos.* 2008-2009.
24. **Larman, Craig.** *UML y Patrones Introducción al análisis y diseño orientado a objetos.* México : Prentice Hall, 1999. ISBN 0-13-748880-7.
25. —. *UML y Patrones Introducción al análisis y diseño orientado a objetos.* México : Prentice Hall, 1999. ISBN 0-13-748880-7.
26. Asociación española para la calidad. [En línea] 1961. <http://www.aec.es/web/guest/centro-conocimiento/validacion-de-diseno>.
27. **Emilio Aviles Avila.** Diagramas de iteración (Secuencia y Comunicación/Colaboración). [En línea] 07 de 06 de 2009. <http://www.slideshare.net/techmi/curso-uml-23-diagramas-de-interaccin>.
28. **Descargarte.net.** software java. [En línea] 25 de 7 de 2012. [Citado el: 16 de 1 de 2013.] <http://www.descargarte.net/tag/software-java>.
29. **RODRIGO GALÍNDEZ.** Control de versiones usando subversion. [En línea] 16 de 5 de 2008. [Citado el: 11 de 12 de 2012.] <http://www.rodriegalindez.com/files/14.pdf>.
30. UptoDown. [En línea] <http://postgresl.uptodown.com/>.
31. **Mozilla Firefox.** GetFirefox. [En línea] 2009. [Citado el: 22 de 11 de 2012.] <http://www.getfirefox.es/firefox-features..>

32. **Frederick, S., Ramsay, Colin y Blades, Steve 'Cutter'**. *Learning Ext JS*. 2008.
33. **Esser, S.** *Secure Programming with the Zend-Framework*. 2009.
34. **Aquino, A.y.L., Yasser.** *Implementación del módulo de Contabilidad General del Sistema Integral de Gestión Cedrux*. 2009.
35. **Ing. Oiner Gómez Baryolo, Ing. Yoandry Morejón Borbón.** *ARQUITECTURA TECNOLÓGICA PARA EL DESARROLLO DE SOFTWARE*.
36. **masadelante.com.** ¿Qué es JavaScript? - Definición de Javascript. [En línea] 1999. [Citado el: 16 de 1 de 2013.] <http://www.masadelante.com/faqs/javascript>.
37. **Pressman, Roger S.** *Ingeniería de software. Un enfoque práctico*. Madrid : s.n., 1997.
38. **Fernández González, Mairelys y Zorrilla Rivera, Osley.** Diseño e implementación del componente Ajuste al Costo del Subsistema Costos y Procesos del Sistema Integral de Gestión de Entidades CEDRUX. *Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas*. Ciudad de la Habana : UCI, 2010.

ANEXOS

Anexo 1 Prototipos de interfaz de usuario funcional.



No.	Nro documento	Vale de entrega	Año	Estado	Productos	Fecha	Creado por
1	16	25	2013	Elaboración	0	2013-05-16	probador
2	17	22	2013	Elaboración	0	2013-05-16	probador
3	18	11	2013	Cancelado	5	2011-06-03	probador
4	18	11	2013	Elaboración	0	2013-05-16	probador

Figura 17 Gestionar vale de devolución

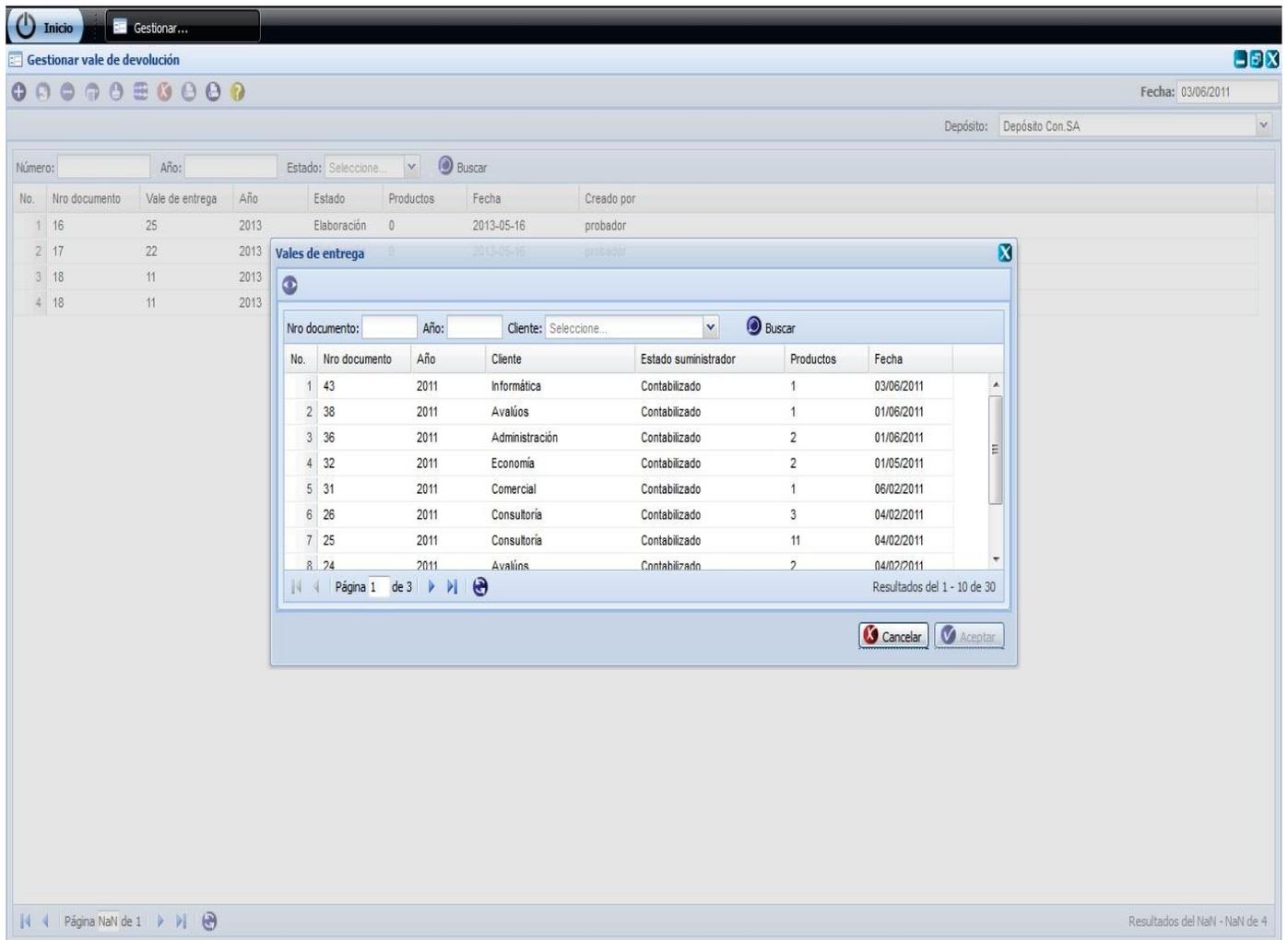


Figura 18 Listar vales de entrega

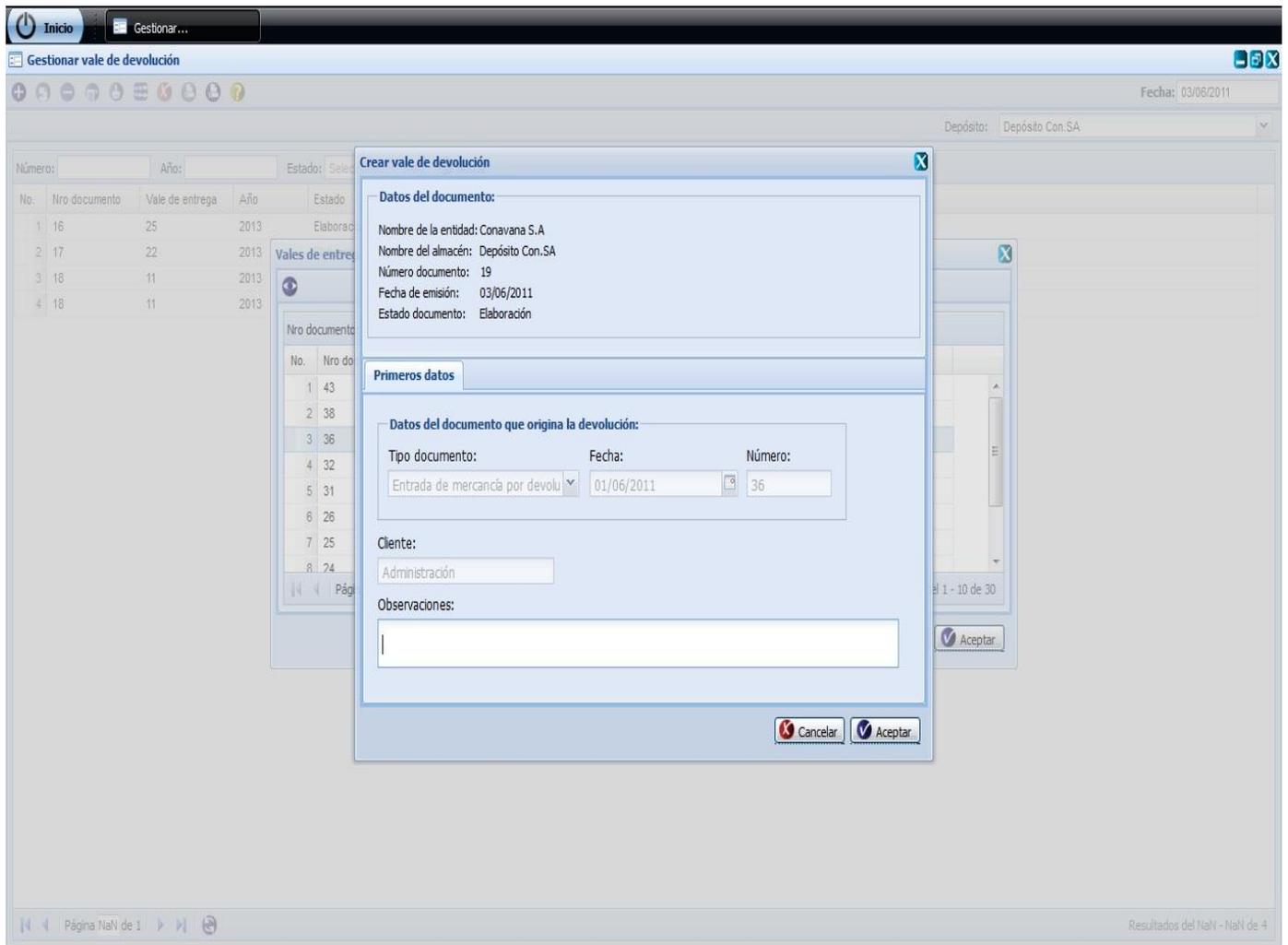


Figura 19 Crear vale de devolución

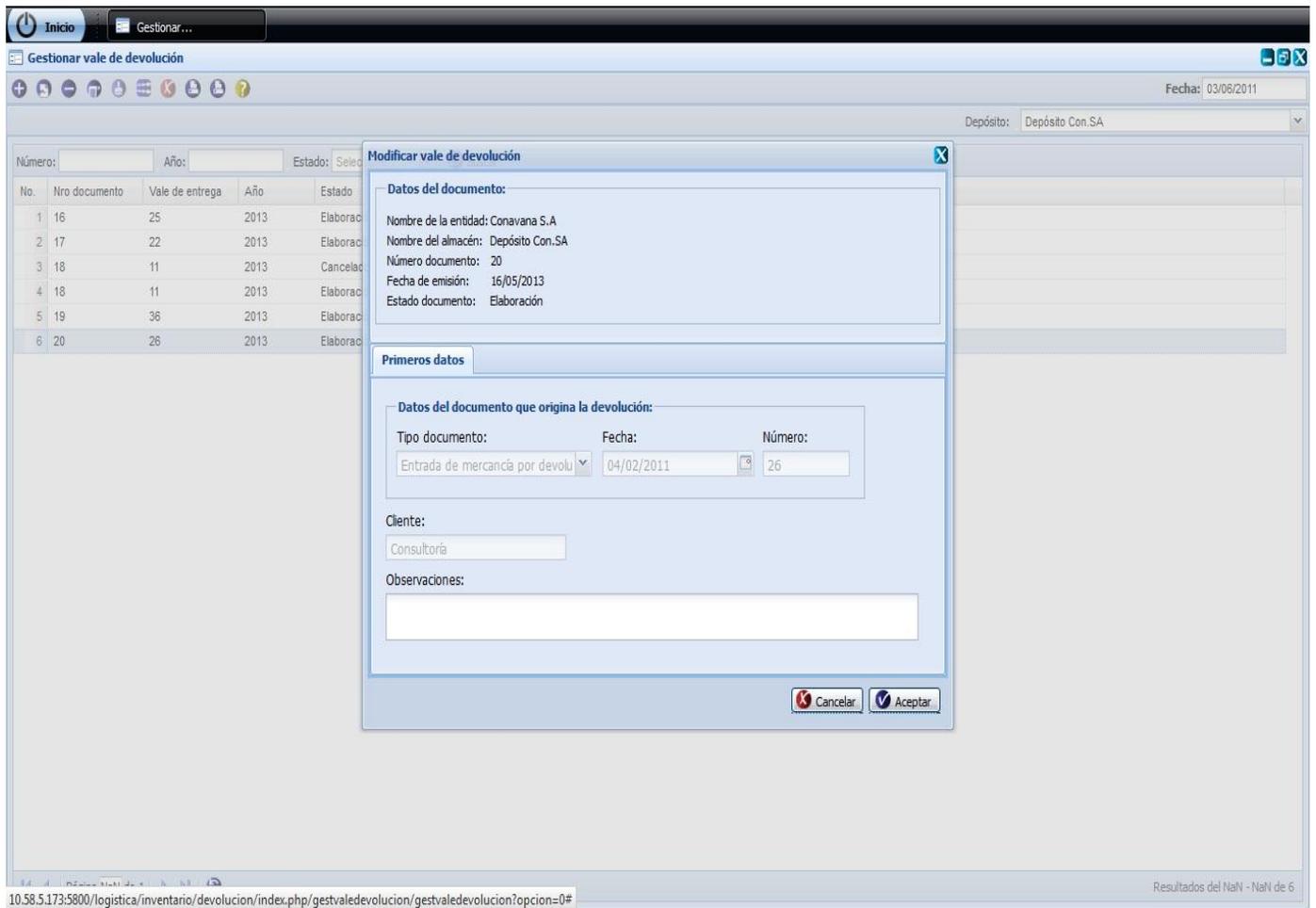


Figura 20 Modificar vale de devolución

Inicio Gestionar...

Gestionar vale de devolución

Productos del documento

Datos del documento:

Nombre del almacén: Depósito Con.SA
Número documento: 20
Fecha de emisión: 16/05/2013
Estado documento: Elaboración

Código: Nro. pieza: Descripción: Buscar

No.	Código	Categoria	Nro. pieza	Descripción	UM	Precio	Cantidad	Importe	Disponibil...	Existen...	Pais
-----	--------	-----------	------------	-------------	----	--------	----------	---------	---------------	------------	------

Página 1 de 1

No hay resultados para mostrar.

10.58.5.173:5800/logistica/inventario/devolucion/index.php/gestvaledevolucion/gestvaledevolucion?opcion=0#

Resultados del X Aceptar

Figura 21 Gestionar productos del vale de devolución

Productos del vale de entrega

No.	Código	Descripción	UM	Precio	A devolver	Cantidad	Importe	Disponibil...	Existen...	Pais	Estado Suminist...
1	100023	NEUMATICO P/ MOTO	Uno	18.900000	0	2		0	9	CUB	0
2	300606	SCOTH TAPE	Uno	0.210000	0	1		0	0	CUB	0
3	300609	FILE DE CARTULINA...	Uno	0.055000	0	10		270	320	CUB	0

Resultados del 1 - 3 de 3

Cerrar

Nombre del almacén: Depósito Con.SA
Número documento: 20
Fecha de emisión: 16/05/2013
Estado documento: Elaboración

Código: Nro. pieza:
No. Código Categoría

Página 1 de 1

No hay resultados para mostrar.

10.58.5.173:5800/logistica/inventario/devolucion/index.php/gestvaledevolucion/gestvaledevolucion?opcion=0#

Aceptar

Figura 22 Listar productos del vale de entrega

Inicio Gestionar...

Gestionar vale de devolución

Productos del documento

Datos del documento:

Nombre del almacén: Depósito Con.SA
Número documento: 20
Fecha de emisión: 16/05/2013
Estado documento: Elaboración

Código: Nro. pieza: Descripción: Buscar

No.	Código	Categoría	Nro. pieza	Descripción	UM	Precio	Cantidad	Importe	Disponibil...	Existen...	País
1	100023	Categoría III		NEUMATICO PI MOTO	Uno	18.900000	2	37.80	0	9	CUB
2	300809	Ninguna		FILE DE CARTULINA CON PESTAÑA	Uno	0.055000	5	0.55	270	320	CUB

Página 1 de 1 Resultados de 1 - 2 de 2

Página NoN de 1 Resultados del Aceptar

Figura 23 Productos del vale de devolución

Informatización del proceso Vales de devolución del subsistema Inventario- CedruX.

UM:		VALE DE ENTREGA O DEVOLUCIÓN VD			Entrega :			Fecha:		
Almacén:		Entregado a:			Emisión:			Devolución:		
Depósito:		Devuelto por:			Vencimiento:					
Orden de Entrega:		Orden de Trabajo:		Centro de Costo:		Cuenta Control:			Análisis:	
Código	Categoría	Nro. pieza	Descripción	UM	Precio	Cantidad	Importe	Disponibilidad	Existencia	País
100023	Categoría III		NEUMÁTICO P/ MOTO	Uno	18.900000	2	37.80		9	CUB
300609	Ninguna		FILE DE CARTULINA CON PESTAÑA	Uno	0.055000	5	0.55	270	320	CUB
Observaciones:										
Entregado por			Recibido por			Anotado por:				
Nombres y apellidos:			Nombres y apellidos:			Tarjeta de Estiba:		Sub. May. Inv.		
Cargo:			Cargo:			Contabilizado por:				
Firma:			Firma:		Fecha:					

Figura 24 Imprimir vale de devolución

Listado de documentos

Nro documento	Vale de entrega	Año	Estado	Productos	Fecha	Creado por
16	25	2013	Elaboración		2013-05-16	probador
17	22	2013	Elaboración		2013-05-16	probador
18	11	2013	Cancelado	5	2011-06-03	probador
18	11	2013	Elaboración		2013-05-16	probador
19	36	2013	Elaboración		2013-05-16	probador
20	26	2013	Contabilizado	7	2011-06-03	probador

Figura 25 Imprimir listado de vales de devolución