

Universidad de las Ciencias Informáticas

Centro de Soluciones de Gestión



Implementación de los Módulos de Ajuste y Baja del Subsistema de Gestión de Inventario del Sistema Integral de Gestión Cedrux.

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

Autor: Modesto Cervantes Alpizar

Tutores: Lic. Arismayda Dorado Risco
Ing. Meylin Martínez Chong

Ciudad de La Habana, Cuba

Junio, 2009

“Año del 50 Aniversario del Triunfo de la Revolución”

AGRADECIMIENTOS

A mis Padres y Abuela por todo el apoyo que me han dado y por el sacrificio que solo yo se que han pasado para que yo llegue hasta aquí.

A mis tías Marle, la China, y la Nena por estar siempre pendientes de mí y ayudarme en todo lo que puedan.

A mi Hermana y Primas por estar siempre ahí, apoyándome en todo momento.

A toda mi familia por ser la mejor del mundo y cuidar de mi mamá mientras yo no estaba.

A mi segunda familia Nenita, Margarita, Armando y toda la familia de Indira por apoyarme en estos cinco años.

A mis Tutoras por el apoyo que me brindaron para la realización de la tesis.

A todas las personas que de una forma u otra han contribuido en mi desarrollo profesional, que cada día han estado ahí para enseñarme cosas nuevas de la vida y llegar a la meta que me he propuesto, esas personas que han estado en los buenos y malos momentos y que nunca me han dado de lado.

A mis Amigos por darme su mano cuando la he necesitado.

A Fidel, a Raúl y a la Revolución por darme la posibilidad de estudiar en la “Universidad del Futuro”.

A Todos Muchas Gracias.

DEDICATORIA

A Fidel y a mi familia y amigos...

PENSAMIENTO



Che

*La revolución se lleva en el corazón,
no en la boca para vivir de ella...*

RESUMEN

Actualmente la situación general de los procesos de gestión de las entidades presupuestadas y empresariales a escala nacional está afectada por la existencia de sistemas informáticos que no cumplen con las expectativas de las nuevas tecnologías y la misión del país en el desarrollo de sus empresas. Proteger los recursos de las entidades así como mejorar la exactitud y confiabilidad de la información económica y contable, ha llevado a la necesidad de mejorar sus procesos en lo referente al Subsistema Inventario utilizando plataformas confiables y eficientes. Hay que destacar que el control de inventario en los almacenes constituye una tarea importante en cualquier empresa. Un adecuado control de los mismos permite minimizar las pérdidas y podemos además saber con exactitud la cantidad real de productos en almacenamiento. El presente trabajo abarca la implementación de los módulos Ajuste y Baja del Subsistema de Inventario, para mejorar la eficiencia en la gestión de los procesos inventaríales en las entidades del país. Para ello se desarrollará un sistema informático utilizando las herramientas propuestas por el equipo de arquitectura del proyecto.

Palabras claves: ajuste, baja, control de inventarios.

ÍNDICE DE CONTENIDO

INTRODUCCIÓN 1

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA 5

1.1 Introducción..... 5

1.2 Flujo actual de los procesos de Ajuste y Baja de Inventario 5

1.3 La gestión de inventarios en la planeación de recursos empresariales 8

1.4 Sistemas Informatizados existentes vinculados al campo de acción..... 9

1.5 Modelo de desarrollo basado en componentes..... 14

1.6 Herramientas, Tecnologías y Lenguajes utilizados 15

 1.6.1 Arquitectura Cliente / Servidor 16

 1.6.2 Las aplicaciones Web..... 16

 1.6.3 Lenguajes del lado del cliente 17

 1.6.4 Lenguaje del lado del Servidor 19

 1.6.5 Sistema Gestor de Base de Datos..... 20

 1.6.6 Servidor Web 20

 1.6.7 Marcos de trabajo e Inversión de control 20

 1.6.8 El Lenguaje de Modelamiento Unificado (UML 2.0)..... 22

 1.6.9 Entorno de Desarrollo Integrado..... 23

 1.6.10 Navegador Mozilla Firefox 3.0..... 24

 1.6.11 Control de Versiones..... 25

1.7 Conclusiones parciales 26

CAPÍTULO II. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA...... 27

2.1 Introducción..... 27

2.2 Flujo del proceso..... 27

 2.2.1 Valoración crítica de los artefactos propuestos por los analistas 27

 2.2.2 Objetos de automatización 30

 2.2.3 Información que se maneja 31

 2.2.4 Propuesta del sistema 31

2.3 Descripción de propuestas del equipo de arquitectura del proyecto..... 32

 2.3.1 Diagrama de despliegue 32

 2.3.2 Integración entre componentes 34

 2.3.3 Estrategia para la captura de errores 35

 2.3.4 Estándares de codificación..... 36

2.4 Descripción de la implementación 37

 2.4.1 Análisis de reutilización de componentes, código o módulos 37

 2.4.2 Diagrama de integración de componentes 39

 2.4.3 Descripción de las principales clases y operaciones a utilizar 42

2.5 Conclusiones parciales 44

CAPÍTULO III. VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.	45
3.1 Introducción	45
3.2 Pruebas de Software	45
3.5.3 Objetivos	46
3.5.4 Alcance.....	46
3.3 Pruebas de Unidad	47
3.3.1 Prueba de Caja Blanca o Estructurales.....	47
3.3.2 Prueba de Caja Negra o Funcionales	48
3.4 Aplicación de pruebas de caja blanca	50
3.5 Evaluación de la calidad utilizando métricas	55
3.5.1 Resultados del instrumento de evaluación de la métrica Tamaño operacional de clase (TOC) ..	59
3.5.2 Resultados del instrumento de evaluación de la métrica Relaciones entre Clases (RC)	61
3.5.3 Resultados del instrumento de evaluación de la métrica Profundidad de Herencia (PH).....	63
3.5.4 Resultados del instrumento de evaluación de la métrica Número de Descendientes (ND)	65
3.5.5 Resultados del instrumento de evaluación de la métrica Número de Operaciones Redefinidas (NOR).....	67
3.6 Matriz de inferencia de indicadores de calidad	69
3.7 Conclusiones parciales	71
 CONCLUSIONES GENERALES	 72
RECOMENDACIONES	73
REFERENCIA BIBLIOGRÁFICA	74
BIBLIOGRAFÍA	76
ANEXOS	78
GLOSARIO DE TÉRMINOS.....	87

ÍNDICE DE FIGURAS

Figura 1 Estructura de las Líneas de Desarrollo	14
Figura 2 Arquitectura Cliente /Servidor.....	16
Figura 3 Arquitectura de las Aplicaciones Web	17
Figura 4 Diagrama de Despliegue.....	33
Figura 5 Arquitectura en tres capas.....	35
Figura 6 Diagrama de Integración	39
Figura 7 Diagrama del Componente Ajuste.....	40
Figura 8 Diagrama del Componente Baja.....	41

Figura 9 Pruebas de Caja blanca y Caja negra.....	47
Figura 10 Componentes de los grafos de flujo	50
Figura 11 Representación del algoritmo <i>Contabilizar()</i>	51
Figura 12 Grafo de flujo asociado al algoritmo <i>Contabilizar()</i>	52
Figura 13 Tamaño operacional de clase (TOC)	57
Figura 14 Relaciones entre clases (RC)	57
Figura 15 Profundidad de Herencia (PH)	58
Figura 16 Número de Descendientes (ND).....	58
Figura 17 Número de Operaciones Redefinidas para una clase hija (NOR)	59
Figura 18 Representación de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.....	59
Figura 19 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.....	60
Figura 20 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Cohesión.....	60
Figura 21 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de Implementación.....	60
Figura 22 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización.....	61
Figura 23 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.....	61
Figura 24 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.....	62
Figura 25 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de Mantenimiento.....	62
Figura 26 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de Pruebas.....	62
Figura 27 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización	63
Figura 28 Representación en % de los resultados obtenidos en el instrumento agrupados por nivel.....	63
Figura 29 Representación de la incidencia de los resultados de la evaluación de la métrica PH en el atributo Complejidad de Mantenimiento.....	64
Figura 30 Representación de la incidencia de los resultados de la evaluación de la métrica PH en el atributo Complejidad de Implementación	64
Figura 31 Representación de la incidencia de los resultados de la evaluación de la métrica PH en el atributo Reutilización	64
Figura 32 Representación de la incidencia de los resultados de la evaluación de la métrica ND en el atributo Reutilización	65
Figura 33 Representación de la incidencia de los resultados de la evaluación de la métrica ND en el atributo Abstracción de la clase base.....	65
Figura 34 Representación de la incidencia de los resultados de la evaluación de la métrica ND en el atributo Cohesión de la Jerarquía de clases.....	66
Figura 35 Representación de la incidencia de los resultados de la evaluación de la métrica ND en el atributo Cantidad de Pruebas.....	66

Figura 36 Representación en % de los resultados obtenidos en el instrumento agrupados en los valores existentes	67
Figura 37 Representación de la incidencia de los resultados de la evaluación de la métrica NOR en el atributo Complejidad del Mantenimiento	67
Figura 38 Representación de la incidencia de los resultados de la evaluación de la métrica NOR en el atributo Cantidad de Pruebas	68
Figura 39 Representación de la incidencia de los resultados de la evaluación de la métrica NOR en el atributo Violación de la Abstracción representada por la superclase	68
Figura 40 Representación de la incidencia de los resultados de la evaluación de la métrica NOR en el atributo Calidad del Diseño.....	68
Figura 41 Gráfico resumen	70
Figura 42 Porcentaje de Cumplimiento de los atributos de Calidad	70
Figura 43 Modelo de Ajuste de Inventario	78
Figura 44 Modelo de Baja de Medios Materiales	79
Figura 45 Resultados de la evaluación de la métrica RC y su influencia en los atributos de calidad	81
Figura 46 Gráfica de los resultados de la evaluación de la métrica RC agrupados por la tendencia de los valores	81
Figura 47 Gráfica de los resultados de la evaluación de la métrica PH.....	82
Figura 48 Resultados de la evaluación de la métrica ND y su influencia en los atributos de calidad	83
Figura 49 Representación de los resultados obtenidos al aplicar los instrumentos que evalúan la métrica ND	83
Figura 50 Resultados de la evaluación de la métrica NOR y su influencia en los atributos de calidad	84
Figura 51 Representación de los resultados obtenidos al aplicar los instrumentos que evalúan la métrica NOR	84
Figura 52 Interfaz para el acceso a los módulo de Inventario	84
Figura 53 Interfaz principal para la gestión de los documentos de Ajuste	85
Figura 54 Interfaz adicionar documento de Ajuste	85
Figura 55 Interfaz para la gestión de los productos	86
Figura 56 Interfaz para la especificación de los números de series	86

ÍNDICE DE TABLAS

Tabla 1 Requisitos de Ajuste	29
Tabla 2 Requisitos de Baja.....	30
Tabla 3 Clases de Ajuste.....	43
Tabla 4 Clases de Baja.....	44
Tabla 5 Matriz de inferencia de indicadores de calidad.....	69
Tabla 6 Rango de valores de para la evaluación técnica de los atributos de calidad.....	80
Tabla 7 Resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad.....	80
Tabla 8 Rango de valores de para la evaluación técnica de los atributos de calidad relacionados con la métrica RC	81
Tabla 9 Resultados de la evaluación de la métrica PH	82
Tabla 10 Rango de valores de para la evaluación técnica de los atributos de calidad relacionados con la métrica ND.....	83
Tabla 11 Rango de valores de para la evaluación técnica de los atributos de calidad relacionados con la métrica NOR	83

INTRODUCCIÓN

Con el desarrollo de la informática en el mundo, se han generado grandes demandas de software para resolver diferentes problemas teóricos y prácticos existentes en la sociedad. En Cuba se está llevando a cabo el proceso de informatización empresarial, el cual es uno de los mecanismos indispensables para el desarrollo económico de una empresa, en búsqueda del aumento en los niveles de eficiencia, organización de las actividades y optimización de los procesos.

Hoy más que nunca las empresas requieren de herramientas que permitan el control y centralización de su información. Los ERP¹ son una solución robusta para aquellas entidades que buscan dichos avances en pos de un mayor desarrollo económico con un menor consumo de recursos. En víspera de los adelantos tecnológicos y el desarrollo de la informática, desde hace algunos años se inició en Cuba el proceso de informatización de la actividad económica nacional. La mayoría de los software de gestión que se comercializan incluyendo los nacionales no cumplen con las necesidades ni las características de la economía cubana, pues generalmente estos se centran en sectores específicos o fueron desarrollados para economías no muy semejantes a la nuestra. Por otro lado esta la existencia de dos monedas que coexisten y comparten legalmente las mismas funciones, lo que se denomina dualidad monetaria, unido a que se fueron actualizando las normas de información financiera por el MFP², aspectos necesarios a tener en cuenta para un mejor desarrollo de los procesos contables.

La economía cubana tiene características específicas, que la diferencia de las restantes en el mundo, se hace necesario por parte del MFP, en especial, por la Dirección de Política Contable de conjunto con los demás Ministerios y Organismos del Estado, la realización de un ERP. El mismo debe ajustarse a las necesidades de las empresas nacionales, de forma que unifique y estandarice toda la información que se genera, que sirva para mejorar la gestión económica de los recursos, y que además ayude a los directivos a la toma de decisiones.

¹ *Planificación de Recursos Empresariales del inglés Enterprise Resource Planning.*

² *Ministerio de Finanzas y Precios.*

El Sistema Integral de Gestión está compuesto por una serie de procesos integrados, la Gestión de Inventario es uno de ellos, el cual tiene entre sus módulos: Ajuste y Baja, logrando un equilibrio entre las existencias físicas y los controles de inventario correspondientes, posibilitando un mejor control de los medios de las entidades, además permite conocer en tiempo real la disponibilidad de recursos para la toma de decisiones, así como mejor y mayor seguridad en el almacenamiento y procesamiento de la información.

Luego de un estudio detallado de la situación anteriormente reflejada se logró identificar el siguiente **problema científico**: ¿Cómo obtener un producto funcional a partir de los requerimientos identificados para la gestión de los procesos de Ajuste y Baja del Subsistema Inventario en las entidades empresariales y unidades presupuestadas del país?

En consecuencia el **objeto de estudio** de la presente investigación son los procesos para el control de inventario, centrándose **el campo de acción** en: los procesos de Ajuste y Baja en las entidades empresariales y unidades presupuestadas del país.

Basado en lo anteriormente expuesto se puede definir el **objetivo general** del presente trabajo como: Implementar los procesos de Ajuste y Baja de los productos en almacenamiento para las entidades empresariales y unidades presupuestadas del país.

De aquí que los **Objetivos Específicos** sean:

- Analizar los procesos de Ajuste y Baja, los sistemas que existen actualmente para su gestión, así como las herramientas que se utilizarán para el desarrollo de la solución.
- Implementar los módulos Ajuste y Baja para su integración al subsistema de Gestión de Inventario.
- Validar la solución propuesta.

Para lograr dichos objetivos se plantearon las siguientes **tareas**:

- Análisis de los sistemas nacionales e internacionales para la gestión de los ajustes y bajas.
- Análisis de las tecnologías, lenguajes y herramientas propuestas para el desarrollo de la aplicación.

- Implementación de las interfaces a partir del prototipo entregado por los analistas.
- Implementación de las funcionalidades del subsistema.
- Realización de las pruebas de unidad.
- Aplicación de métricas para validar los resultados del producto obtenido.

Como **Idea a defender** se tiene: Si se realiza la implementación de los procesos de Ajuste y Baja de los productos en almacenamiento para las entidades empresariales y unidades presupuestadas del país, se logrará obtener un producto funcional para la gestión de dichos procesos.

La culminación e implantación de esta aplicación Web proporcionará a las entidades los siguientes **aportes prácticos**:

- ✓ Las entidades dispondrán de una aplicación Web totalmente funcional, cumpliendo con todos los requerimientos establecidos. Permitirá elaborar el modelo de ajuste y baja de inventario por los conceptos establecidos, que sean aprobados los documentos por el jefe responsable y que se actualicen las existencias de los medios materiales.
- ✓ Permitirá registrar los números de series a los productos que por su condición lo requieran, para ser dados de baja de la entidad.
- ✓ El módulo de ajuste permite además hacerle un ajuste al importe del producto ya sea por exceso o por defecto. Brindará además un servicio para crear automáticamente un documento de ajuste cuando en la aprobación de un inventario físico exista un faltante o un sobrante en alguno de sus productos.

El presente documento está estructurado en 3 capítulos fundamentales que se describen brevemente a continuación:

- **En el Capítulo 1. Fundamentación Teórica:** Se desarrolla un estudio del estado del arte de algunos de los sistemas ERP, así como los distintos sistemas informáticos existentes en el mundo para el control de inventario. Además se exponen los conceptos fundamentales relacionados con el tema de investigación y se describen los lenguajes, las herramientas y metodologías a utilizar para el desarrollo de la aplicación.

- **En el Capítulo 2. Descripción y análisis de la solución propuesta:** Se desarrolla una valoración crítica de los artefactos propuestos por los analistas, se realiza un análisis de posibles implementaciones, componentes o módulos ya existentes y que puedan ser reutilizados, estrategias de integración, descripción de la solución propuesta, información que se maneja, descripción de las clases que se utilizan para representar computacionalmente dicha estructura y descripción de las nuevas clases u operaciones necesarias.
- **En el Capítulo 3. Validación de la solución propuesta:** Se valida la solución propuesta, se realiza la descripción de las pruebas de unidad teniendo en cuenta (Objetivo, Alcance, Tipos y detalles de las mismas), así como el análisis de los valores utilizados para las pruebas y evaluación de la ejecución de estas y de los resultados obtenidos. También se utilizan algunas métricas de calidad para validar la solución.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

La tecnología en la actualidad se encuentra en constante evolución, por lo que se hace necesario tener un conocimiento avanzado y actualizado de esta a la hora de desarrollar aplicaciones informáticas. Su incidencia ha propiciado innumerables ventajas como el almacenamiento y centralización de grandes volúmenes de información, rapidez en la obtención de resultados, facilidades para encontrar información adecuada y/o actualizada, ahorro de tiempo y dinero, posibilidades para procesar datos; todo esto hace que los sistemas informáticos constituyan un mecanismo de apoyo para la toma de decisiones. Teniendo en cuenta lo antes planteado se ha decidido en el presente capítulo abordar sobre las herramientas, lenguajes, tecnologías y modelo de desarrollo basado en componentes propuesto por el equipo de arquitectura del proyecto, se realiza una síntesis acerca de algunos sistemas, que permiten el control de inventario y se exponen los principales conceptos y aspectos relacionados con el tema de investigación.

1.2 Flujo actual de los procesos de Ajuste y Baja de Inventario

Ajuste de inventario

Durante el proceso para la realización de los ajustes de inventario en el caso de los medios materiales controlados por el almacén, se confrontan los saldos de los modelos “Tarjeta de Identificación y Control de Estiba” contra los “Submayores de Inventario Continuo” según corresponda y, si se observan diferencias entre ellos, se revisan las operaciones aritméticas y se cotejan las anotaciones efectuadas en ambos modelos, con el fin de detectar operaciones dejadas de anotar o anotadas indebidamente, así como las anotaciones de cantidades en exceso o en defecto.

Luego se revisan los pases a ambos registros de las operaciones realizadas desde la fecha del último conteo físico hasta la fecha en que se detecta el sobrante o faltante, partiendo de los documentos primarios.

Los resultados de las comprobaciones descritas anteriormente se plasman en el “Movimiento documental”, el cual se elabora de forma individual por cada uno de los medios materiales que presentan diferencias. A este documento se le anexa la relación de las entradas y salidas dejadas de anotar o anotadas incorrectamente, así como los errores aritméticos detectados. Si al concluir la elaboración del “Movimiento Documental” se determina que el faltante o sobrante es originado por errores aritméticos o de anotación, se procede al ajuste del inventario en correspondencia con el origen de las diferencias detectadas, de la manera siguiente:

Ajustes por faltantes o sobrantes

La comisión designada para realizar los conteos físicos generales o parciales, o el funcionario que detecta el faltante procede a levantar un acta. El jefe de la entidad, en caso de ser necesario, designa una comisión a fin de realizar las investigaciones y comprobaciones que correspondan y, en un plazo que no exceda los sesenta días naturales, deberá completar un expediente integrado por el original del acta, los modelos “Movimiento Documental” y un modelo de “Ajuste de Inventario” por cada uno de los productos afectados.

El expediente se remite al jefe de la entidad para su aprobación, quien debe decidir además la aplicación de la responsabilidad material, disciplinaria o penal cuando corresponda. En los casos en que se requiera la aprobación del nivel superior, el jefe de la entidad correspondiente envía el expediente al nivel inmediato superior para su trámite o aprobación.

Ajustes por mermas, averías o vencimiento

Cuando las mermas o averías no excedan de las normas aprobadas o las diferencias se produzcan por vencimiento natural de los medios, se confecciona únicamente el modelo “Ajuste de Inventario”. Cuando se trate de mermas o averías no normadas o que excedan las normas establecidas, se elabora un dictamen pericial por un especialista que designa el jefe de la entidad, el cual se anexa al expediente que se confecciona por el faltante.

Ajustes por faltantes o averías ocurridos como consecuencia de hechos extraordinarios.

La comisión designada para investigar los hechos, elabora un acta. A continuación se cumplen los demás pasos de los procedimientos establecidos en el punto correspondiente a “Ajustes por mermas, averías o vencimiento” y se procede de acuerdo con lo establecido en las Normas y Procedimientos.

Baja técnica

Para llevar a cabo el proceso de baja técnica a algún medio material el jefe que la propone está obligado a nombrar una comisión técnica, compuesta por no menos de tres miembros en correspondencia con el tipo de medio, la cual procede a realizar la defectación del medio, determinar las causas de la baja, significando si el medio es inservible o debe desarmarse para aprovechar sus partes, piezas y componentes, en correspondencia con lo establecido en las “tareas técnicas” de las entidades.

Luego dicha comisión elabora un dictamen con los resultados de su trabajo. En este se hace constar las causas que originan la proposición de baja. Este dictamen se confecciona en tres ejemplares. A continuación se elabora el modelo “Baja de Medios Materiales”, al cual se le anexa el dictamen técnico emitido.

Se remite el original y duplicado de ambos modelos al jefe del nivel superior, quien aprobará o tramitará la aprobación de la baja, según sea el caso, con el jefe facultado para ello. Se habilita un expediente donde se archiva toda la documentación relacionada con la solicitud.

Una vez aprobada la baja, el nivel que la aprueba actualiza los registros que tiene habilitados para el control de las existencias de dichos medios.

En el caso de que se le den baja a medios especiales, se sigue el mismo procedimiento y se tramitan por las entidades que controlan dichos medios.

Si la necesidad de dar baja a un medio se origina por un hecho extraordinario que lleve implícito un daño material, además de cumplirse lo establecido en la presente norma, el jefe de la entidad, a la cual pertenece el medio, procede, en correspondencia con lo establecido para la exigencia de la responsabilidad material, disciplinaria o penal, según corresponda.

Al aprobarse la baja técnica de un medio la comisión designada que propone la baja realiza la defectación técnica de las partes, piezas y componentes del medio y determina las que pueden ser utilizadas como repuestos, las que pueden comercializarse como chatarra. Una vez realizado el desarme de los medios materiales se procede a cerrar el expediente con el modelo “Acta Final de Medios Materiales”. En el caso de la destrucción se utiliza el modelo “Acta de Utilización o Destrucción de Medios Materiales”.

Si en la baja se declara el medio inservible o se le asigna un nuevo destino, la comisión verifica que se le dé el destino previsto según la decisión del jefe que aprobó la baja técnica, siguiendo los procedimientos establecidos.

1.3 La gestión de inventarios en la planeación de recursos empresariales

La Planificación de Recursos Empresariales (Enterprise Resource Planning, ERP) es una forma de utilizar la información en áreas claves como fabricación, compras, administración de inventario y cadena de suministros, control financiero, administración de recursos humanos, logística y distribución, ventas, mercadeo y administración de relaciones con clientes. Se trata de unir estos elementos, y proporcionar a los usuarios del sistema una manera universal de acceder, ver, y utilizar la información que se guarda en diferentes sistemas de gestión empresarial a través de una sola aplicación.

Uno de los módulos gerenciales que existen comúnmente en las empresas y que es integrable a un ERP lo es el referente a los procesos del Control de Inventarios. Este módulo revierte gran importancia en el manejo de los productos en almacenes controla los medios que posee una organización. Se encarga de inspeccionar las existencias de productos, dándole entrada y salida del almacén a los mismos, regulando el flujo de mercancía en almacenamiento, con el fin de hacer más rentable su posesión y garantizar en cierto grado el éxito de la organización.

El Control de Inventario involucra entre otros, los procesos de Recepción, Apertura, Despacho, Ajuste y Baja.

Los objetivos principales de los sistemas ERP son:

1. Optimización de los procesos empresariales.
2. Acceso a información confiable, precisa y oportuna.
3. La posibilidad de compartir información entre todos los componentes de la organización.

4. Eliminación de datos y operaciones innecesarias.
5. Reducción de tiempos y de los costes de los procesos.

1.4 Sistemas Informatizados existentes vinculados al campo de acción

Existen hoy diversos sistemas informáticos que gestionan los procesos de inventario en las empresas; estos integran y automatizan muchas de las prácticas del negocio asociadas con los aspectos operativos o productivos de una empresa, entre los que se incluyen el ajuste y baja de los productos en almacenamiento. El uso de estos sistemas informáticos continuará expandiéndose ampliamente en el futuro, debido a su impacto sobre los negocios y la industria. En la actualidad no existe en Cuba un sistema informático integral de gestión que cumpla con la totalidad de los requerimientos de funcionalidad, interoperabilidad y seguridad que espera el gobierno cubano de una solución de este tipo, de manera que pueda ser utilizada como herramienta para potenciar el cumplimiento de las funciones de las entidades a todos los niveles con un máximo de racionalidad y control de los recursos financieros, materiales y humanos.

Sistemas Internacionales existentes en nuestro país:

- ✓ **SAP:** Desde 1972, SAP es una empresa de software empresarial, líder a nivel mundial, especializada en software para gestión empresarial. A través de sus soluciones de software de gestión, SAP ofrece alternativas para más de 25 sectores industriales. Con subsidiarias en más de 50 países. Para conseguir procesos óptimos de gestión de inventarios necesita una funcionalidad sólida para gestionar su logística. El soporte para la gestión de inventarios le ayuda a registrar y a realizar un seguimiento de los materiales, según la cantidad y el valor. La funcionalidad de gestión de inventarios de almacén cubre los movimientos de almacén internos y el estocaje. El software de SAP para la gestión de inventarios también permite realizar, en tiempo real, la planificación de la carga de trabajo, picking por fases y consolidación de pedidos, el escaneado de radiofrecuencia y de códigos de barras, la gestión de las unidades de manipulación y el cross-docking. (1)

El subsistema Inventario de SAP permite reducir los costes de almacenamiento, transporte, cumplimiento de pedidos y manipulación de materiales y, a la vez, mejorar el

servicio al cliente. Puede mejorar significativamente la rotación de inventario, optimizar el flujo de mercancías y acortar las rutas en su almacén o centro de distribución. Entre los beneficios adicionales de la gestión de inventarios se encuentran la mejora del flujo de caja, la visibilidad y la toma de decisiones.

Entre sus características principales se encuentran:

- Implementado en .NET y WebSphere.
 - SAP también ofrece una nueva plataforma tecnológica denominada SAP NetWeaver, esta plataforma tecnológica convierte a SAP en un programa Web-enabled, lo que significa que estaría totalmente preparado para trabajar con él mediante la web.
 - Trabaja sobre el sistema operativo Windows.
 - Soporte para bases de datos Oracle .
- ✓ **Openbravo:** Es un reconocido desarrollador de soluciones profesionales en software libre para empresas, que ofrece la primera alternativa real al software propietario. Su sistema en entorno web de gestión integral de empresas (ERP) y de gestión de punto de venta (PoS), los más populares en su área, han sido descargados más de un millón de veces y se utilizan en alrededor de 50 países. Openbravo ERP ha sido específicamente diseñado para ayudar a las empresas a mejorar su rendimiento. La cobertura funcional del producto incluye todas las áreas típicas de un sistema de gestión integrado. Está desarrollado sobre Java y Javascript, dispone de soporte para bases de datos PostgreSQL y Oracle. (2)

Los procesos referentes al Control de Inventario que incorpora Openbravo ERP permiten que las existencias en su organización estén siempre al día y correctamente valoradas. La posibilidad de definir la estructura de almacenes de su organización hasta el mínimo nivel (ubicación) facilita que los stocks estén siempre perfectamente localizados. Adicionalmente, las capacidades para gestionar los lotes de mercancías y la posibilidad de utilizar números de serie aseguran el cumplimiento de los requisitos de trazabilidad impuestos en la mayoría de las industrias.

- ✓ **OpenERP:** Es un completo sistema de gestión empresarial (ERP) que cubre las necesidades de las áreas de contabilidad, ventas, compras, y almacén e inventario, entre otras. ESTUDIO42 colabora activamente en la localización y desarrollo de OpenERP para la Argentina, y brinda soluciones completas para diversos sectores de actividad: fabriles, de servicios, de venta minorista o distribución mayorista, y comercio electrónico. OpenERP soporta múltiples monedas, múltiples compañías y múltiples contabilidades; además incorpora funcionalidades de gestión de documentos para agilizar la colaboración entre departamentos y equipos en la empresa; y permite trabajar remotamente mediante una interfaz web desde una computadora conectada a Internet. La gestión de inventario se lleva a cabo con doble-entrada, tal como en la contabilidad. Los lotes no se crean en ubicaciones de inventario, sino que son movidos de una ubicación a otra. (3)

Características:

- Desarrollada en Python y PyGTK.
- Base de Datos PostgreSQL.
- Tiene desarrollado clientes en la librería qt.
- Frontend web basado en TurboGears.

Sistemas Nacionales:

- ✓ **Versat Sarasola:** Sistema cubano de contabilidad confiable, permite enviar información eficaz, de forma inmediata, desde lugares apartados, a la vez que ofrece mayor organización, control y disciplina en cada gestión. Fue éste el primer sistema de contabilidad cubano certificado, en cuya evaluación participaron el Ministerio de Finanzas y Precios, consultorías internacionales y el organismo encargado de la seguridad informática. Es un sistema económico integrado. Constituido por 12 módulos entre los que se encuentra el de **Control de inventarios**. Actualmente lo utilizan alrededor de 200 entidades de varias provincias. Registra todos los documentos primarios que tradicionalmente generan las entradas y salidas en los almacenes; además tiene incorporada una gran cantidad de opciones que permiten ejecutar los conteos físicos y el tratamiento de los diferentes tipos de inventarios, que para el subsistema se llaman categorías. Permite nombrar productos.

En el módulo de **Control de Inventarios** se definen formatos del clasificador de productos para lograr una uniformidad en el registro y la agregación de información en los reportes de salida, se conceptualizan los movimientos para lograr una información amplia sobre los orígenes y destinos de los recursos. Permite el control de las existencias y movimientos en diferentes monedas. Muestra el cuadro diario de cada uno de los almacenes por las diferentes cuentas. Ofrece la posibilidad de duplicar documentos para agilizar los pases de los mismos y lograr que un mismo documento se convierta en otro con solo adicionar un mínimo de información, realiza un control de las existencias y movimientos por custodios y se emiten diferentes reportes e información de utilidad para la correcta administración de los recursos materiales.

Características:

- Es una aplicación de escritorio.
 - Implementado en Delphi.
 - Trabaja sobre el sistema operativo Windows.
 - Soporte para base de datos SQL Server 2000.
- ✓ **Rodas XXI Versión 3.0:** Sistema multiempresa y multiusuario creado por CITMATEL para la automatización de la gestión empresarial. Contiene diferentes módulos que pueden usarse integrados o independientes uno de ellos es **Inventarios**.

El módulo de Inventario de Rodas XXI le permite tener un control detallado de los inventarios de su entidad, realizando en el mismo momento que se registra un movimiento, su contabilización. Se pueden realizar todo tipo de operaciones de entradas y salidas de los almacenes con facilidad en el momento que se desee, generando el documento asociado al movimiento de que se trate de forma automática previa configuración del sistema para ello. Es posible trabajar con varios almacenes trabajando cada uno de forma independiente. Este módulo le permitirá además visualizar información correspondiente a períodos anteriores, tan sólo con cambiar de período contable a periodos anteriores ya cerrados, aunque en dichos periodos no podrá realizar ninguna operación. Esta característica es compartida por todos los módulos de RODAS XXI.

- ✓ **Siscont5:** El sistema se aviene a las definiciones y conceptos del Ministerio de la Industria Básica aunque por las acciones contables financieras que permite puede ser utilizado en otras entidades nacionales. Puede ser explotado en régimen monousuario y multiusuario. Se define para monoentidad y multientidad, para esta última existe el control de su acceso para las entidades en un mismo equipo de cómputo como servidor.

El módulo de Inventario maneja toda la información referida al Sub mayor de Inventarios de la entidad, garantizando el cuadro permanente con las respectivas cuentas de la Contabilidad General. El sistema esta preparado para controlar el saldo de cada material en dos monedas, a partir de los procedimientos vigentes en el país en cuanto a política monetaria. Permite el tratamiento de las contabilizaciones de forma transaccional, por resúmenes diarios o de forma personalizada según se defina por parámetros. Incorpora tratamiento de lotes a los productos.

- ✓ **Siscomip:** Es un Sistema Contable Integral diseñado para las empresas del Ministerio de la Industria Pesquera. Su arquitectura no es modular. Está concebido para llevar a cabo el registro contable de la actividad económica, a nivel de documento primario, en una entidad cualquiera compuesta por unidades, de forma compartimentada o no, y está en explotación actualmente en entidades pertenecientes al Ministerio de la Industria Pesquera. Permite intercambio de información entre los subsistemas de forma automática y por opciones. Puede ser implementado en entornos de redes o en máquinas aisladas.

(4)

Valoración crítica

Luego de analizar los sistemas implantados en nuestro país, se puede concluir que no es factible la utilización de los mismos en las entidades cubanas. Los sistemas nacionales fueron desarrollados sobre plataformas de software propietario, lo que implica incrementos de gastos en licencias de uso y mantenimiento del software. La mayoría son aplicaciones de escritorio lo que conlleva que el usuario deba instalar la aplicación en cada estación de trabajo. Son productos que se caracterizan por abordar solamente partes del problema de la gestión de la empresa, no soportan mecanismos estándares de integración con otras aplicaciones, la mayoría fueron desarrollados para un ambiente multiusuario, casi ninguno bajo conceptos de informática multicapa y distribuida en la red. De los sistemas internacionales que se utilizan en nuestro país

se puede decir que los mismos tienen un alto nivel de configuración permitiendo cubrir la mayoría de las necesidades de los módulos de Ajuste y Baja, pero tienen la desventaja que alguno de ellos están implementados con tecnología que no es posible su acceso desde nuestro país debido a las restricciones impuestas por el gobierno de los Estados Unidos, ejemplo de esto es el OpenBravo el cual está basado en la plataforma J2EE³ cuya máquina virtual es propiedad de la compañía americana SUN⁴, además del elevado consumo de memoria en comparación con PHP/Apache. Otra de las desventajas que estos sistemas presentan es que no cumplen con las necesidades ni las características de la economía cubana, pues generalmente se centran en sectores específicos o fueron desarrollados para economías no muy semejantes a la nuestra como lo es la capitalista que tiene un modelo de gestión y de procesos muy diferente a las empresas cubanas. El resto de los software son propietarios lo cual no resulta una buena elección pues significan una suma elevada de dinero para nuestro país en concepto de licencias y mantenimiento.

1.5 Modelo de desarrollo basado en componentes

El modelo de desarrollo utilizado fue definido por la subdirección de Arquitectura del Centro de Soluciones de Gestión. Cuenta con los siguientes **Roles**:

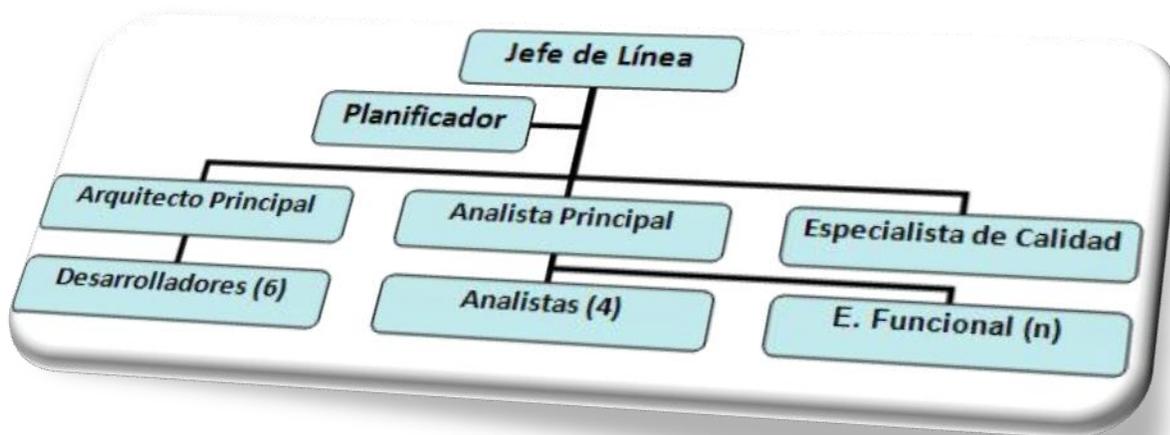


Figura 1 Estructura de las Líneas de Desarrollo

Flujos de Trabajo/Artefactos:

³ Java Platform, Enterprise Edition o Java EE es una plataforma de programación para desarrollar y ejecutar aplicaciones en lenguaje de programación Java.

⁴ Stanford University Network, Red de la Universidad de Stanford.

Inicio

- Plan de Iteración (J. de la Línea y Planificador)
- Plan de Trabajo Individual (Lo elaboran Todos) (Lo controla el planificador y lo evalúa el J. de la Línea)

Análisis

- Arquitectura de Negocio
 - Mapa de Proceso (Analista Principal)
 - Descripción de Procesos de Negocio (Analista)
- Especificación de Requisitos (Analista)
- Casos de Prueba (E. Calidad, Analistas y E. Funcionales)
- Arquitectura del Sistema (Arquitecto de Sistema)

Diseño e Implementación

- Modelo Conceptual (Arquitecto de Sistema) (Desarrolladores, Analistas)
- Modelo de Datos (Desarrollador)
- Diseño de Clases (Desarrollador)
- Diseño de Interfaz de Usuario (Desarrollador)
- Release (Arquitecto de Sistema)

Prueba

- Casos de Prueba (E. de Calidad los refina) (Equipo de Pruebas Central)
- Registro de no Conformidades (E. de Calidad responsable) (Equipo de Pruebas Central registran las NC)

Estabilización e Integración

- Historia de la Iteración (J. de la Línea)
- Informe de Integración (Arquitecto de Sistema)

Mantenimiento

- Peticiones de Actualizaciones (En teoría es el Dpto. de Implantación)
- Registro de Bugs (En teoría es el Dpto. de Implantación)

(5)

1.6 Herramientas, Tecnologías y Lenguajes utilizados

Las herramientas, lenguajes y tecnologías que se describen a continuación en el capítulo, son las utilizadas en la implementación de los componentes. Fue una decisión determinada por el equipo de arquitectura del Centro de Soluciones de Gestión.

1.6.1 Arquitectura Cliente / Servidor

Cuando se habla de aplicaciones Web se debe pensar en la mejor arquitectura para el control e intercambio de información a través de la red. La Arquitectura Cliente / Servidor es una de las más importantes y usadas en este ámbito de enviar y recibir información, como su nombre lo indica está compuesto por un cliente y un servidor, donde el cliente realiza una petición de recursos, información o servicios al servidor, y este último se encarga de proporcionar al cliente las peticiones realizadas. Una de las ventajas de esta arquitectura es que el acceso a la información se realiza de forma más ágil y al estar almacenada en el servidor existe un mejor control de la seguridad. (6)

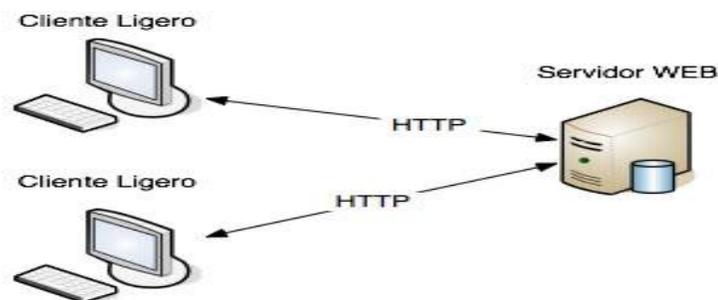


Figura 2 Arquitectura Cliente /Servidor

1.6.2 Las aplicaciones Web.

Las aplicaciones Web se desarrollan como una extensión de los Sistemas Web para agregar funcionalidad de negocio al proceso. En otros términos, es un sistema informático que los usuarios utilizan accediendo a un servidor Web a través de Internet o de una Intranet. Su arquitectura general es la de un sistema cliente/servidor. Normalmente instalar una aplicación Web consiste en configurar los componentes del lado del servidor en la red y no es necesaria una instalación o configuración en el lado cliente. Una aplicación Web está comúnmente estructurada como una aplicación de tres-capas. En su forma más común, el navegador Web es la primera capa, un motor usando alguna tecnología Web dinámica es la capa intermedia, y una base de datos como última capa.

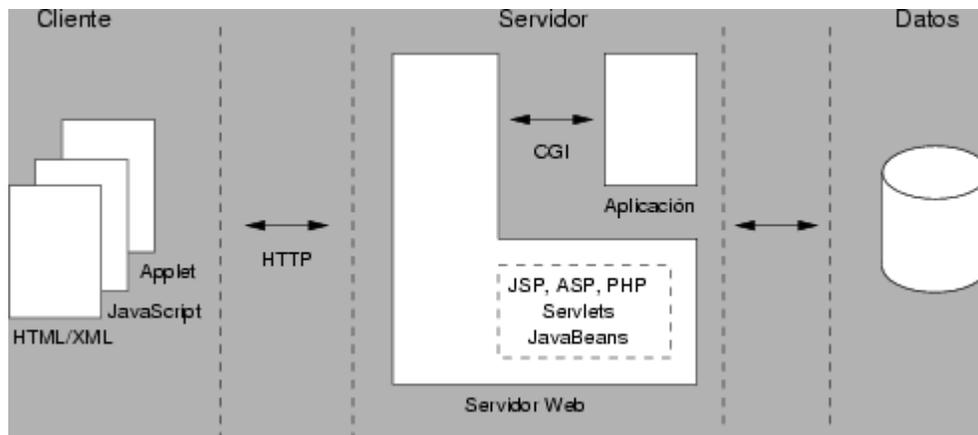


Figura 3 Arquitectura de las Aplicaciones Web

1.6.3 Lenguajes del lado del cliente

JavaScript

Javascript es un lenguaje con muchas posibilidades, utilizado para crear pequeños programas que luego son insertados en una página web y en programas más grandes, orientados a objetos mucho más complejos. Con Javascript podemos crear diferentes efectos e interactuar con nuestros usuarios. Este lenguaje posee varias características, entre ellas podemos mencionar que es un lenguaje basado en acciones que posee menos restricciones. Además, es un lenguaje que utiliza Windows y sistemas X-Windows, gran parte de la programación en este lenguaje está centrada en describir objetos, escribir funciones que respondan a movimientos del mouse, aperturas, utilización de teclas, cargas de páginas entre otros. Es necesario resaltar que hay dos tipos de JavaScript: por un lado está el que se ejecuta en el cliente, este es el Javascript propiamente dicho, aunque técnicamente se denomina Navigator JavaScript. Pero también existe un Javascript que se ejecuta en el servidor, es más reciente y se denomina LiveWire Javascript.

(7)

HTML (HyperText Markup Language)

Un científico llamado Tim Berners-Lee inventó HTML allá por 1990. El objetivo era facilitar a científicos de diferentes universidades el acceso a los documentos de investigación de cada uno de ellos. El proyecto obtuvo un éxito mucho mayor del que Tim Berners-Lee nunca hubiese llegado a imaginar. Al inventar HTML, este científico sentó las bases de la web tal y como la conocemos hoy día. HTML es un lenguaje que hace posible presentar información (por ejemplo, investigaciones científicas) en Internet. Lo que ves al visualizar una página en Internet es la

interpretación que hace el navegador del código HTML. Es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas Web. Gracias a Internet y a navegadores como Explorer o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos. (8)

XML (Extensible Markup Language)

XML, con todas las tecnologías relacionadas, representa una manera distinta de hacer las cosas, más avanzada, cuya principal novedad consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes. Así pues, el XML juega un papel importantísimo en este mundo actual, que tiende a la globalización y la compatibilidad entre los sistemas, ya que es la tecnología que permitirá compartir la información de una manera segura, fiable, fácil. Además, XML permite al programador y los soportes dedicar sus esfuerzos a las tareas importantes cuando trabaja con los datos, ya que algunas tareas tediosas como la validación de estos o el recorrido de las estructuras corre a cargo del lenguaje y está especificado por el estándar, de modo que el programador no tiene que preocuparse por ello.

XML no está sólo, sino que hay un mundo de tecnologías alrededor de él, de posibilidades, maneras más fáciles e interesantes de trabajar con los datos y, en definitiva, un avance a la hora de tratar la información, que es en realidad el objetivo de la informática en general. XML, o mejor dicho, el mundo XML no es un lenguaje, sino varios lenguajes, no es una sintaxis, sino varias y no es una manera totalmente nueva de trabajar, sino una manera más refinada que permitirá que todas las anteriores se puedan comunicar entre si sin problemas, ya que los datos cobran sentido. (9)

AJAX (Asynchronous JavaScript And XML)

AJAX, acrónimo de Asynchronous JavaScript And XML («JavaScript y XML asíncronos»). Técnica de desarrollo web para crear aplicaciones interactivas mediante la combinación de tres tecnologías ya existentes:

- HTML (o XHTML) y Hojas de Estilo en Cascada (CSS) para presentar la información.

- Document Object Model (DOM) y JavaScript, para interactuar dinámicamente con los datos.
- XML y XSLT, para intercambiar y manipular datos de manera desincronizada con un servidor web (aunque las aplicaciones AJAX pueden usar otro tipo de tecnologías, incluyendo texto llano, para realizar esta labor).

Como el DHTML o LAMP, AJAX no constituye una tecnología en sí, pero es un término que engloba a un grupo de éstas que trabajan conjuntamente.

Las aplicaciones AJAX usan navegadores web que soportan las tecnologías mencionadas más arriba. Entre estos se incluyen Mozilla, Firefox, Internet Explorer, Opera, Konqueror y Safari. (10)

1.6.4 Lenguaje del lado del Servidor

PHP 5.2.3 (Hypertext Preprocessor)

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor, gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación. Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como PHP está desarrollado en política de código abierto, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores. Actualmente PHP se encuentra en su versión 5, que utiliza el motor Zend, desarrollado con mayor meditación para cubrir las necesidades de las aplicaciones web actuales. Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por poner dos ejemplos. Algunas de las más importantes capacidades de PHP son: compatibilidad con las bases de datos más comunes, como MySQL, mSQL, Oracle, Informix, y ODBC, por ejemplo. Incluye funciones para el envío de

correo electrónico, upload de archivos, crear dinámicamente en el servidor imágenes en formato GIF, incluso animadas y una lista interminable de utilidades adicionales. (12)

1.6.5 Sistema Gestor de Base de Datos

PostgreSQL 8.2

Es un servidor de base de datos relacional, libre. Tiene soporte total para transacciones, disparadores, vistas, procedimientos almacenados, almacenamiento de objetos de gran tamaño. Se destaca en ejecutar consultas complejas, consultas sobre vistas, subconsultas y joins de gran tamaño. Permite la definición de tipos de datos personalizados e incluye un modelo de seguridad completo. Como toda herramienta de software libre PostgreSQL tiene entre otras ventajas las de contar con una gran comunidad de desarrollo en Internet, su código fuente está disponible sin costo alguno y algo muy importante es que dicha herramienta es multiplataforma.

1.6.6 Servidor Web

Apache 2.0

Apache 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.

Alguna de sus características:

- Es multiplataforma, aunque idealmente está preparado para funcionar bajo Linux.
- Muy sencillo de configurar.
- Es Open-source.
- Muy útil para proveedores de Servicios de Internet que requieran miles de sitios pequeños con páginas estáticas.
- Amplias librerías de PHP y Perl a disposición de los programadores.
- Posee diversos módulos que permiten incorporarle nuevas funcionalidades, estos son muy simples de cargar.
- Es capaz de utilizar lenguajes como PHP, TCL, Python, entre otros. (13)

1.6.7 Marcos de trabajo e Inversión de control

Zend Framework

Es un framework ⁵open source, que está diseñado para php 5 y buenas capacidades de ampliación.

Dentro de sus principales características están:

- Trabaja en 3 capas, o sea con el Modelo Vista Controlador.
- Cuenta con módulos para manejar archivos en formato de documento portátil, canales de sindicación de noticias y servicios web.
- Incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar la base de datos.
- Completa documentación y pruebas de alta calidad.
- Robustas clases para autenticación y filtrado de entrada.
- Clientes para servicios web.

Zend_Ext Framework

Es un framework open Source, que está diseñado para php 5 y buenas capacidades de ampliación. Es elaborado a partir de Zend Framework cumpliendo con todas sus características. Este trae de novedoso un controlador vertical para el control de las acciones realizadas por las vistas hacia el controlador, un motor de reglas para las validaciones en el servidor, se le incluyó el IoC⁶ para la comunicación entre los módulos o componentes. Se le incorporó la integración con el ORM Doctrine Framework para trabajo en la capa de abstracción a base de datos y el ExtJs Framework para el desarrollo de las vistas.

Doctrine Framework

Framework Doctrine es un potente y completo sistema de mapas de relaciones de objetos (Object Relational Mapper, ORM por sus siglas en ingles) para PHP 5.2+ con una base de datos con capas de abstracción incorporada. Además es multiplataforma.

UCID Framework

Es el Framework encargado del trabajo con las vistas. Abarca la integración de ExtJs Framework con el sistema incluyendo el integrador de interfaz, el generador de interfaz dinámica y la impresión de documentos. Integra la iconografía, los diferentes temas de escritorio de la aplicación, el multilinguaje.

⁵ Marco de trabajo

⁶ Inversión de Control del inglés *Inversion of Control*

ExtJS Framework 2.2

Su potencia radica en la rica colección de componentes para el diseño de interfaces haciendo uso extensivo de Ajax. Entre los componentes que este framework ofrece encontramos cuadros de diálogo, menús, tablas editables, layouts, paneles, pestañas y todo lo necesario para construir atractivos desarrollos al estilo de Web 2.0, además de crear componentes propios haciendo uso de las clases que trae. Permite una comunicación asincrónica, agilidad, estética y mejor trabajo con la capa de presentación. Consta con un estilo amigable, muy similar a las aplicaciones de escritorio y es muy aceptado por los usuarios. (11)

Inversión de Control

IoC es un concepto junto a unas técnicas de programación en las que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales, en los que la interacción se expresa de forma imperativa haciendo llamadas a procedimientos (procedure calls) o funciones. Tradicionalmente el programador especifica la secuencia de decisiones y procedimientos que pueden darse durante el ciclo de vida de un programa mediante llamadas a funciones. En su lugar, en la inversión de control se especifican respuestas deseadas a sucesos o solicitudes de datos concretas, dejando que algún tipo de entidad o arquitectura externa lleve a cabo las acciones de control que se requieran en el orden necesario y para el conjunto de sucesos que tengan que ocurrir. El flujo habitual se da cuando es el código del usuario quien invoca a un procedimiento de una librería. La inversión de control sucede cuando es la librería la que invoca el código del usuario. Típicamente sucede cuando la librería es la que implementa las estructuras de alto nivel y es el código del usuario el que implementa las tareas de bajo nivel.

1.6.8 El Lenguaje de Modelamiento Unificado (UML⁷ 2.0)

El Lenguaje de Modelamiento Unificado es un lenguaje gráfico para visualizar, especificar y documentar cada una de las partes que comprende el desarrollo de software. Permite especificar todas las decisiones de análisis, diseño e implementación, construyéndose así modelos precisos, no ambiguos y completos. Estimula el crecimiento de herramientas orientadas a objetos. Integra las mejores prácticas para la modelación y puede soportar todos los lenguajes de programación. (14)

⁷ Unified Modeling Language

1.6.9 Entorno de Desarrollo Integrado

ZendStudio for Eclipse 6.0

Programa de la casa Zend, uno de los mayores impulsores de PHP, orientada a desarrollar aplicaciones web. Zend Studio es un editor de texto para páginas PHP que proporciona un buen número de ayudas desde la creación y gestión de proyectos hasta la depuración del código. A diferencia de las versiones anteriores a la v5.5.1.281, ya no se trata de un IDE ⁸desarrollado en Java (excesiva lentitud y consumo de memoria en algunos casos), ahora esta basado en Eclipse, Eclipse es una plataforma de software de código abierto independiente de una plataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido". Soporta varias librerías como: ExtJS, Yahoo UI y JQuery para la presentación así como la inclusión de FrameWork para el desarrollo de la web pudiendo combinarlas fácilmente en una aplicación. ZendStudio está disponible como una aplicación independiente, se puede encontrar para dos plataformas fundamentales: Windows y GNU/Linux.

Entre sus principales Características se encuentran:

- Soporta PHP4 y PHP5.
- Asistente de código.
- Plantillas (PHP, PHPDoc, Nuevo archivo).
- Código plegable (clases, funciones y PHPDoc).
- Tiempo real la detección de errores.
- Barra de Herramientas de Apoyo
- Inserción automática (entre paréntesis, llaves, PHPDoc).
- PHP (Project) Vista Exploratoria.
- Elemento de búsqueda de PHP. Archivo / Proyecto / PHP Inspectores (esquemas).
- Buscar y reemplazar en archivos.
- Plataforma de Integración.
- Plataforma API.
- Framework Project.
- Galerías de código.

⁸ Entorno de Desarrollo Integrado en inglés, *Integrated Development Environment*

- Zend Guard integration.
- Zend Studio Migration.
- Zend Studio Keymaps.
- PHPDocumentor.
- Fácil de depuración. (15)

1.6.10 Navegador Mozilla Firefox 3.0

Mozilla Firefox es un navegador de Internet, con interfaz gráfica de usuario desarrollado por la Corporación Mozilla y un gran número de voluntarios externos. Firefox, oficialmente abreviado como Fx o fx, y comúnmente como FF, comenzó como un derivado del Mozilla Application Suite, que terminó por reemplazarlo como el producto bandera del proyecto Mozilla, bajo la dirección de la Fundación Mozilla.

Es un navegador Web multiplataforma, que está disponible en versiones para Microsoft Windows y Linux. Posee diversas características:

- Navegación con pestañas: Firefox abre por defecto las nuevas páginas web en "pestañas", cada una de esas pestañas tiene su propio botón de cerrado. Cuando hay demasiadas pestañas abiertas se ajustan a una sola ventana, aparecen a cada lado unas flechas de desplazamiento, y hay siempre un botón en la parte derecha para que ver la lista de todas las pestañas que tienes abiertas.
- Corrector ortográfico: Firefox cuenta con un corrector ortográfico con miles de palabras en casi cualquier idioma.
- Títulos dinámicos: los títulos dinámicos son resúmenes que se actualizan a menudo con información importante sobre una página web. Cuando la información de la página se actualiza, también lo hace el título dinámico.
- Búsqueda integrada: La barra de búsqueda viene con los motores de búsqueda de Google, Yahoo!, eBay, Diccionario RAE, Wikipedia y Creative Commons cargados por defecto. Estés donde estés navegando, se puede introducir un término en la barra de

búsqueda y recibir respuesta inmediata del motor de búsqueda que hayas elegido. Se puedes seleccionar un nuevo motor desde la barra de búsqueda en cualquier momento, y añadir motores de búsqueda de cientos de webs favoritas. (16)

1.6.11 Control de Versiones

Una versión, revisión o edición de un producto, es el estado en el que se encuentra en un momento dado en su desarrollo o modificación. Se llama control de versiones a la gestión de los diversos cambios que se realizan sobre los elementos de algún producto o una configuración del mismo. Los sistemas de control de versiones facilitan la administración de las distintas versiones de cada producto desarrollado, así como las posibles especializaciones realizadas. Un sistema de control de versiones debe proporcionar un mecanismo de almacenaje de los elementos que deba gestionar y un registro histórico de las acciones realizadas con cada elemento o conjunto de elementos (normalmente brindando la posibilidad de volver o extraer un estado anterior del producto) entre otros aspectos. Todos los sistemas de control de versiones se basan en disponer de un repositorio, que es el conjunto de información gestionada por el sistema. Este repositorio contiene el historial de versiones de todos los elementos gestionados. Cada uno de los usuarios puede crearse una copia local duplicando el contenido del repositorio para permitir su uso. Es posible duplicar la última versión o cualquier versión almacenada en el historial.

Subversion

Subversion es un software de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS, el cual posee varias deficiencias. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como svn, por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de Subversion es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente. En cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en cierto punto del tiempo.

TortoiseSVN 1.6.2

TortoiseSVN es un cliente gratuito de código abierto para el sistema de control de versiones *Subversion*. Maneja ficheros y directorios a lo largo del tiempo. Los ficheros se

almacenan en un *repositorio* central. El repositorio es prácticamente lo mismo que un servidor de ficheros ordinario, salvo que recuerda todos los cambios que se hayan hecho a sus ficheros y directorios. Esto permite que pueda recuperar versiones antiguas de sus ficheros y examinar la historia de cuándo y cómo cambiaron sus datos, y quién hizo el cambio. Esta es la razón por la que mucha gente piensa que Subversion, y los sistemas de control de versiones en general, son una especie de “máquinas del tiempo”.

Algunos sistemas de control de versiones también son sistemas de manejo de configuración del software (SCM). Estos sistemas están diseñados específicamente para manejar árboles de código fuente, y tienen muchas características que son específicas para el desarrollo de software tales como el entendimiento nativo de los lenguajes de programación, o proporcionan herramientas para compilar software. Subversion, sin embargo, no es uno de estos sistemas; es un sistema general que puede ser utilizado para manejar cualquier colección de ficheros, incluyendo código fuente. (17)

1.7 Conclusiones parciales

Con la realización de este capítulo se puede afirmar que el mundo está en un proceso de constante cambio donde las tecnologías de la información están tomando un mayor auge y con ellas las empresas aumentan la producción y mejoran la gestión de sus procesos. En gran medida estos avances están muy relacionados con el surgimiento de los Software de Planificación Empresariales los cuales son muy vendidos y utilizados en el mundo por las entidades para mejorar algunos de los procesos que se llevan a cabo en las mismas como es el control de capital humano, las finanzas y **el control de inventario** que tan necesario es para las mismas. Por tanto se decidió realizar un software que cumpla con las necesidades de los clientes y que además se desarrolle utilizando los métodos y herramientas explicadas en este capítulo como los son el PHP como lenguaje del lado del servidor, Extjs para la parte visual, el apache como servidor web, como gestor de base de datos se utilizará PosgreSQL aprovechando todas las potencialidades que este brinda. Siempre realimentándose por software ya existentes y con una vasta experiencia en esta rama como lo es el **Versat Sarasola**, primer sistema de contabilidad cubano certificado.

CAPÍTULO II. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA.

2.1 Introducción

El siguiente capítulo se abordan los aspectos principales a tener en cuenta a la hora del desarrollo de los módulos, para esto, primeramente se realizó una valoración crítica de los artefactos propuestos por los analistas, un análisis de posibles implementaciones, componentes o módulos ya existentes que puedan ser reutilizados, estrategias de integración, descripción de las clases que se utilizan para representar computacionalmente dicha estructura y descripción de las mismas. También se muestran los principales diagramas resultantes de todo el proceso de desarrollo así como los estándares de codificación utilizados.

2.2 Flujo del proceso

2.2.1 Valoración crítica de los artefactos propuestos por los analistas

En un proceso de desarrollo de software es de gran importancia la descripción de los requisitos funcionales, los cuales son brindados por el analista de sistema y facilitan un mejor entendimiento de los procesos a desarrollar, permitiendo comprender en profundidad el problema en cuestión y facilitando una mejor identificación de las clases y funcionalidades que serán implementadas.

La especificación de requerimientos es la base que permite verificar si se alcanzaron o no los objetivos establecidos en el proyecto ya que estos son un reflejo detallado de las necesidades de los clientes o usuarios del sistema y es contra lo que un requerimiento según la IEEE (Standard Glossary of Software Engineering Terminology) se puede definir como una:

1. Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.
2. Condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato, estándar, u otro documento impuesto formalmente.
3. Una representación documentada de una condición o capacidad como en 1 o 2.

Los requisitos se pueden clasificar en funcionales y no funcionales.

Requerimientos funcionales: Son capacidades o condiciones que el sistema debe cumplir.

Los requerimientos no funcionales: Son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable.

Existen múltiples categorías para clasificar a los requerimientos no funcionales, siendo las siguientes representativas de un conjunto de aspectos que se deben tener en cuenta, aunque no limitan a la definición de otros.

- Requerimientos de Software
- Requerimientos de Hardware
- Restricciones en el diseño y la implementación
- Requerimientos de apariencia o interfaz externa
- Requerimientos de Usabilidad
- Requerimientos de Soporte
- Requerimientos Legales
- Requerimientos de confiabilidad
- Requerimientos de interfaz Interna
- Requerimientos de Seguridad

La seguridad puede ser tratada en tres aspectos diferentes:

- Confidencialidad
- Integridad
- Disponibilidad

Los requisitos funcionales que fueron identificados por los analistas y que posibilitaron la identificación de las clases y las funcionalidades a implementar son:

Requisitos de ajuste

Ajuste	Adicionar documento	Gestionar documento
	Modificar documento	
	Eliminar documento	
	Buscar documento	Buscar documento
	Búsqueda avanzada de documento	Búsqueda avanzada de documento
	Confirmar	Confirmar
	Aprobar	Aprobar
	No aprobar	
	Cancelar estado	Cancelar estado
	Contabilizar	Contabilizar
	Adicionar producto	Gestionar producto
	Modificar producto	
	Eliminar producto	
	Buscar producto	Buscar producto
	Imprimir modelo	Imprimir modelo
	Imprimir listado	Imprimir listado
Ayuda	Ayuda	

Tabla 1 Requisitos de Ajuste

Requisitos de baja

Baja	Adicionar documento	Gestionar documento
	Modificar documento	
	Eliminar documento	
	Buscar documento	Buscar documento

	Búsqueda avanzada de documento	Búsqueda avanzada de documento
	Confirmar	Confirmar
	Aprobar	Aprobar
	No aprobar	
	Cancelar estado	Cancelar estado
	Contabilizar	Contabilizar
	Adicionar producto	Gestionar producto
	Modificar producto	
	Eliminar producto	
	Buscar producto	Buscar producto
	Adicionar equipo	Gestionar equipo
	Modificar equipo	
	Eliminar equipo	
	Buscar equipo	Buscar equipo
	Gestionar nro. Serie	Gestionar nro. Serie
	Imprimir modelo	Imprimir modelo
	Imprimir listado	Imprimir listado
	Ayuda	Ayuda

Tabla 2 Requisitos de Baja

2.2.2 Objetos de automatización

En el proceso de **ajuste de inventario** cuando se detectan diferencias entre las existencias físicas de los medios materiales y los controles en los registros, se genera toda una documentación que avala la realización de los ajustes.

Se desea automatizar:

- La elaboración de los modelos “Ajuste de Inventario”, los cuales se generan en dependencia del origen de las diferencias detectadas y de los productos afectados.
- La transmisión de información de un nivel a otro de forma tal que cada usuario conozca el estado de los documentos al que tiene acceso y pueda desempeñar su trabajo como corresponde.
- La aprobación de los modelos “Ajuste de Inventario” por la persona encargada. Una vez aprobado el ajuste los registros habilitados para el control de las existencias serán actualizados.

En el proceso de **baja técnica** a los medios materiales también se genera una serie de documentos que justifican este proceso.

De esta documentación se desea automatizar:

- La elaboración del modelo “Baja de Medios Materiales”, el cual debe ser tramitado para su aprobación.
- La aprobación de los modelos “Baja de Medios Materiales” por la persona encargada. Una vez aprobada la baja los registros habilitados para el control de las existencias serán actualizados.

Una vez aprobados estos modelos se procede a la actualización de las existencias de los productos afectados.

2.2.3 Información que se maneja

Modelos

➤ ***Ajuste de Inventario:***

Permite amparar desde el punto de vista documental el proceso de ajuste que surja producto de las diferencias entre las existencias físicas y los controles de inventarios correspondientes. Ver en Anexo 1

➤ ***Baja de Medios Materiales:***

Permite amparar desde el punto de vista documental hasta su destino final las solicitudes de baja de los medios materiales. Ver en Anexo #2

2.2.4 Propuesta del sistema

Para llevar a cabo los procesos de ajuste y baja a cualquier producto o medio material, el sistema inicialmente debe permitir crear un documento de ajuste o de baja que avale la búsqueda de los documentos por diferentes criterios y el registro de productos. Los criterios de búsqueda de documento son los siguientes: por año, motivo, estado, número del documento, creado por, aprobado por y por rango de fecha. Para la realización de alguna acción sobre el documento el sistema verifica primeramente que ese documento no esté siendo usado por otro usuario, de ser así el sistema te dará la opción de ver el documento en forma de reporte, de lo contrario según

los privilegios del usuario registrado este podrá realizar cinco acciones diferentes según el estado del documento: una es que se muestra el documento en forma de reporte sin poderse realizar ningún cambio, otra permite actualizar los productos, ya sea insertar, modificar valores(cantidad a ajustar o dar de baja) o eliminar dichos productos del documento, en el caso particular de la baja, si el producto seleccionado es un equipo se le deben registrar los números de series correspondientes al tipo de control que presente el mismo y además podrá visualizar los números de series, el código y el año de fabricación de los mismos. Los números de series que fueron entrados por una recepción o una apertura para ese producto pueden ser reutilizados en la baja, siempre teniendo en cuenta que no supere la existencia total de números de series en el sistema para ese producto. Cuando el motivo del ajuste es por error de importe el sistema solo te debe permitir introducir el importe. La otra acción permite confirmar el documento y una vez confirmado puedes cancelar el estado del mismo, luego se procede a la aprobación o no del documento y finalmente es contabilizado el documento afectando con esta la existencia del producto en el almacén. Las tres últimas acciones cambian el estado del documento (elaboración, preparado, aprobado, contabilizado). Si no existe ningún documento el sistema permite crear nuevos documentos donde una vez elaborados se puede modificar, eliminar y mostrar el documento en formato de reporte para su visualización y futura impresión en caso de que el usuario lo desee. Cuando se registra un ajuste se deben introducir los datos referentes al motivo del ajuste y las observaciones del documento, si lo que estamos registrando es una baja pues necesita el destino final, los motivos, el informe técnico, a que será destinado el mismo y las observaciones. Recordar siempre que según el rol será, la visibilidad o no de las acciones.

2.3 Descripción de propuestas del equipo de arquitectura del proyecto

2.3.1 Diagrama de despliegue

El modelo de despliegue muestra la configuración de los nodos de procesamiento en tiempo de ejecución, los links de comunicación entre ellos, y las instancias de los componentes y objetos que residen en ellos. El propósito del modelo de despliegue es capturar la configuración de los elementos de procesamiento, y las conexiones entre estos elementos en el sistema. El modelo consiste en uno o más nodos (elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos), dispositivos (nodos estereotipados con una capacidad de procesamiento en el nivel modelado de abstracción), y conectores, entre nodos, y

CAPÍTULO II. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

entre nodos y dispositivos. El modelo de despliegue también mapea procesos dentro de estos elementos de procesamiento, permitiendo la distribución del comportamiento a través de los nodos que son representados. El Modelo de Despliegue se utiliza para capturar los elementos de configuración del procesamiento y las conexiones entre esos elementos. También se utiliza para visualizar la distribución de los componentes de software en los nodos físicos.

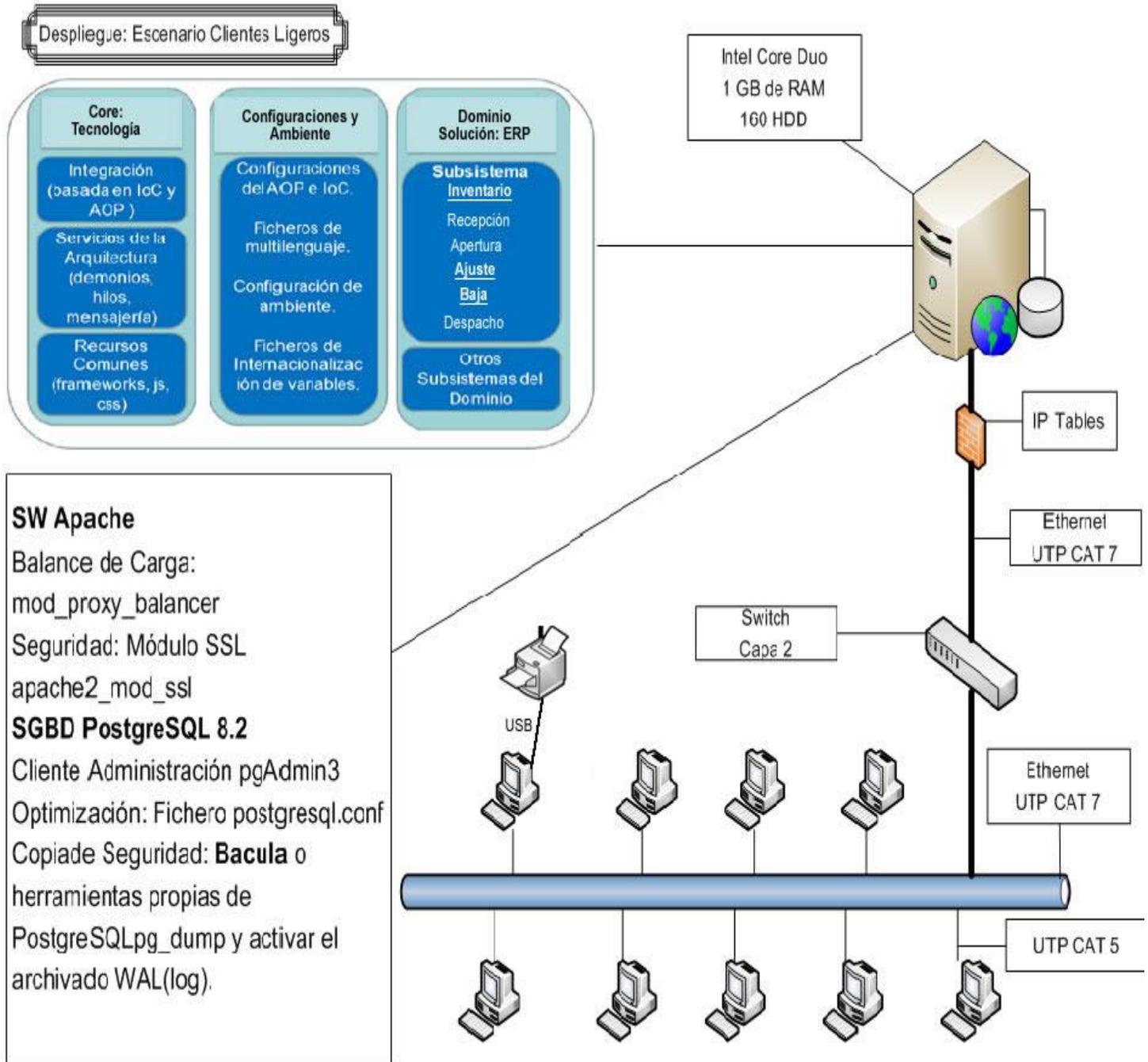


Figura 4 Diagrama de Despliegue

2.3.2 Integración entre componentes

MVC⁹ es un patrón de diseño de arquitectura de software que se usa principalmente en aplicación que maneja gran cantidad de datos y transacciones complejas donde se requiere una mejor separación de los conceptos para que el desarrollo este estructurado de una mejor manera, facilitando la programación en diferentes capas de manera paralela e independiente. Dada las ventajas que presenta dicho patrón se decidió utilizarlo en la implementación de la aplicación. Esta arquitectura posibilita un trabajo más seguro, rápido y eficiente pues antes se programaba de una manera engorrosa todo detrás del botón, lo que hace una aplicación más lenta e insegura. La arquitectura en tres capas maneja el flujo de datos desde la capa superior que es la vista hasta la inferior que es la de Acceso a Datos y viceversa pasando por cada uno de los elementos que la componen. Esta consta de cuatros nodos de integración, el que encontramos entre la vista y el controlador, el que está entre la el controlador y el modelo, el que vincula el modelo con el framework doctrine y el que se encuentra entre el doctrine y la base de datos. La comunicación entre diferentes módulos y componentes se realiza mediante llamadas a la inversión de control (*IoC*). El *IoC* especifica respuestas deseadas a sucesos o solicitudes de datos concretas, dejando que otro módulo o componente lleve a cabo las acciones de control que se requieran en el orden necesario y para el conjunto de sucesos que tengan que ocurrir. En un fichero XML llamado *IoC* son registrados todos los servicios o funcionalidades que brinda cada una de las clases de control de los componentes del sistema, este fichero será mapeado por el framework para su funcionamiento. Existen dos clases de control, uno es el *IoC* interno que es el que integra los componentes de cada módulo o línea de desarrollo y está el *IoC* externo que brinda todos los servicios para que los módulos puedan integrarse de ser necesario logrando un bajo acoplamiento entre los componentes. En los componentes de documento, movimiento y producto se utiliza el patrón de diseño proxy sirviendo de intermediario entre el objeto cliente y objeto deseado obligando a que las llamadas a métodos de un objeto ocurran indirectamente a través de un objeto Proxy. El sistema cuenta también con los *conceptos globales*, los cuales brindan toda la información necesaria sobre el sistema en cualquier momento, ya sea el usuario, la fecha contable, la entidad, el subsistema entre otras que son de interés para el usuario. La base de datos es accedida de forma directa

⁹ Modelo-Vista-Controlador del inglés Model-View-Controller

mediante controladoras y los componentes rehusados son integrados mediante interfaces sencillas, garantizando así una total integración de las capas en el sistema.



Figura 5 Arquitectura en tres capas

2.3.3 Estrategia para la captura de errores

Como todo sistema, para garantizar su correcto funcionamiento se debe tener en cuenta un tratamiento de errores, hacer captura de las excepciones que son lanzadas por el sistema para advertir al usuario de que la cosa va mal y debe ser cambiada, el sistema captura todas aquellas excepciones que son lanzadas y se le da un tratamiento para que el sistema no colapse.

Se sabe de antemano que a la hora de registrar un nuevo usuario por poner un ejemplo, al introducir algún dato que este incorrecto nuestro sistema debe ser capaz de advertir al usuario en caso contrario, esta excepción es lanzada por el sistema. De esta manera se da solución a las problemáticas de los lanzamientos de excepciones que se dan en el sistema según las peticiones del usuario, capturando los tipos de errores lanzados y mostrando de manera visible al usuario mensajes de confirmación, el cual le dará una idea de que anda mal y como solucionarlo.

Mediante la interfaz Web se impedirá que el usuario asuma un papel activo en la introducción de la información, para esto se contará con cuadros de opción, menú de selección lo cual facilitará la entrada de datos. La información que requiera ser adicionada por el usuario se validará mediante funciones o expresiones regulares que garanticen que sea válida y que el cuadro de texto no esté vacío si es obligatorio llenarlo. Si hay un error en la información le saldrá al usuario un mensaje en pantalla indicándole el error, al oprimir Aceptar el mensaje desaparece y el usuario podrá seguir introduciendo los datos en el formulario. También se validarán las opciones correspondientes a la extracción o modificación de datos del servidor de base datos. Si se desea eliminar algún elemento de la BD se preguntará al usuario si está seguro de realizar dicha acción,

al igual que cuando desee modificar alguna información, antes de actualizarla se le preguntará si desea realizarla o no. En el sistema existen funciones que necesitan validaciones previas para que se ejecuten sin ningún problema, para estas validaciones se usan varios ficheros XML entre ellos el *validation* el cual contiene las llamadas a las precondiciones que deben cumplirse si como las poscondiciones, igualmente esta el fichero *exception* el cual contiene los mensajes que serán mostrados según la excepción lanzada, otro de los ficheros XML es el *ManagerException* el cual captura excepciones previas que definimos ya sean para excepciones del Doctrine o del Framework. Así se logra que se realicen las operaciones que se desean y que se rectifique al cometer un error.

2.3.4 Estándares de codificación

Las convenciones o estándares de codificación son pautas de programación que no están enfocadas a la lógica del programa, sino a su estructura y apariencia física para facilitar la lectura, comprensión y mantenimiento del código. La utilización de los estándares permite reducir el tiempo de desarrollo, el coste y el esfuerzo. Es por ello que al comenzar este proyecto se estableció un estándar de codificación para asegurar de que todos los programadores del proyecto trabajaran de forma coordinada.

Nomenclatura según el tipo de clases

1. Clases controladoras

Las clases controladoras después del nombre llevan la palabra: "Controller".

Ejemplo: GestionarUsuarioController

2. Clases de los modelos

- Business (Negocio)

Las clases que se encuentran dentro de Business después del nombre llevan la palabra: "Model".

Ejemplo: MonedaContModel

3. Domain (Dominio)

Las clases que se encuentran dentro de Domain el nombre que reciben es el de la tabla en la Base de Datos.

Ejemplo: Dat_Documento

4. Generated (Dominio Bases)

Las clases que se encuentran dentro de Generated el nombre comienza con la palabra: “Base” y seguido el nombre de la tabla en la Base de Datos.

Ejemplo: BaseDat_Documento

5. Nomenclatura de las funciones

El nombre a emplear para las funciones se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se empleará notación *CamelCasing**, y con sólo leerlo se reconoce el propósito de la misma.

Ejemplo: insertarMoneda

- En caso de ser una acción de la clase controladora se le pone el nombre y seguida la palabra:”Action”

Ejemplo: insertarMonedaAction

Notación CamelCasing

La letra inicial del identificador no debe estar en mayúscula. Esta notación se utilizó para el nombre de las funciones y el nombre de los atributos. A continuación ejemplos:

insertarMoneda: Este nombre de método esta compuesto por 2 palabras, la primera todo en minúsculas y la segunda iniciando con letra mayúscula.

dineroM: Este nombre del atributo esta compuesto por 2 palabras, la primera todo en minúsculas y la segunda iniciando con letra mayúscula.

2.4 Descripción de la implementación

2.4.1 Análisis de reutilización de componentes, código o módulos

A continuación se abordan los componentes que fueron utilizados para la implementación de la aplicación facilitando estos la gestión con los documentos, movimientos, productos y números de series.

CAPÍTULO II. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

Componente Documento

Dentro de los componentes que se utilizan en la implementación de los módulos de Ajuste y Baja se encuentra el componente de documento, el cual es encargado de la gestión de los documentos, teniendo este una relación de herencia con los documentos de ajustes y baja que son hijos de documento ya que son igualmente documentos lo que con algunas características particulares pero siempre con datos comunes que son almacenados en el padre.

Componente Movimiento

El componente de movimiento es el encargado de la unión entre el documento y los productos. Dicho componente facilita todas las operaciones que son necesarias para operar con los movimientos de ese documento. Un movimiento es la entrada o salida de un producto en un documento. Dentro del componente de movimiento se encuentra también el movimiento de los equipos, para en el caso de la baja, la recepción y la apertura incluirle los números de series al producto que sea equipo.

Componente Producto

Es el componente encargado de la gestión de los productos dentro del almacén, ya sea los que están nombrados como los nuevos que se incluyen. Este componente brinda todos los servicios necesarios para operar sobre los productos, que a través de los movimientos se relacionan con los documentos. Los productos pueden ser equipos o no equipos, también tienen categorías según la calidad del mismo, además se registra la nacionalidad, el tipo de control, el número de pieza y otros atributos que son importantes para los almacenes.

Componente Número de Series

El componente de número de serie es el encargado de gestionar los números de series a los productos que sean equipos y que por su tipo de control (*cuantitativo, cualitativo y cualitativo obligatorio*) lo requiera. Los atributos que se registran en este componente son: nacionalidad, año de fabricación y el número de series que es único en todo el almacén. Siempre se tiene en cuenta que la cantidad de números de series no sea mayor que la cantidad en existencia del producto.

2.4.2 Diagrama de integración de componentes

Un *diagrama de integración de Componentes* ilustra los fragmentos de software, controladores embebidos, etc. que conformarán un sistema. Un diagrama de integración tiene un nivel de abstracción más elevado que un diagrama de clase, usualmente un componente se implementa por una o más clases (u objetos) en tiempo de ejecución. Estos son bloques de construcción, como así eventualmente un componente puede comprender una gran porción de un sistema.

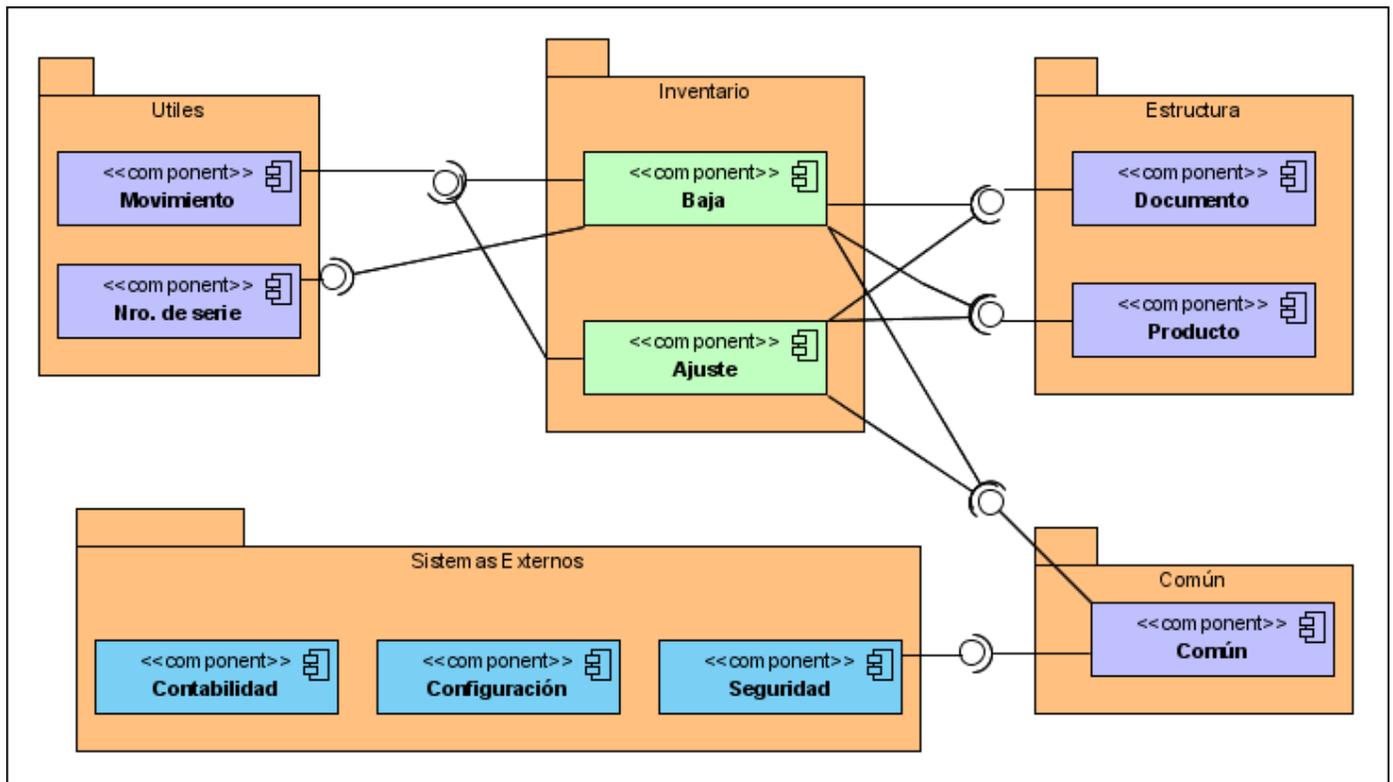


Figura 6 Diagrama de Integración

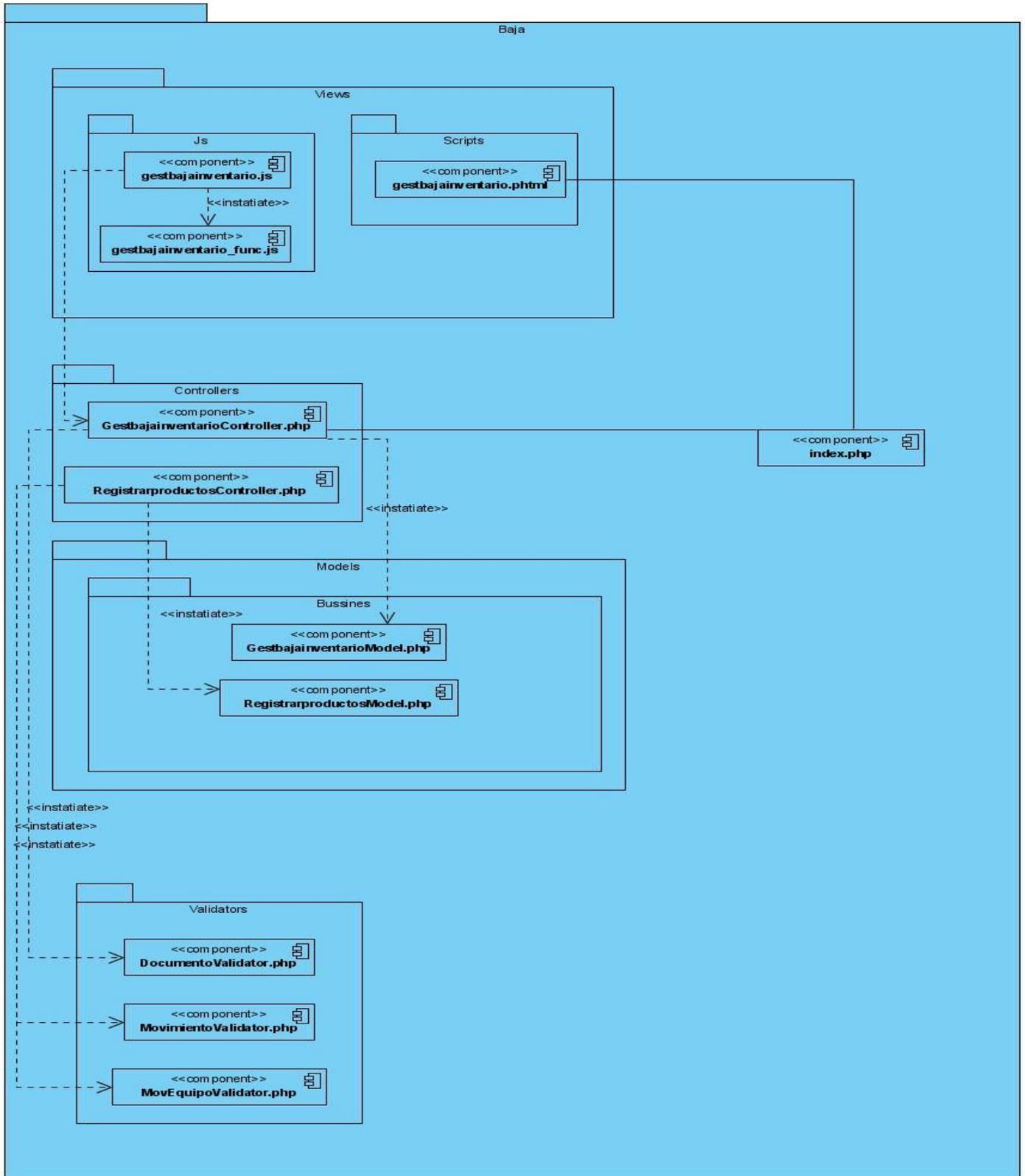


Figura 8 Diagrama del Componente Baja

2.4.3 Descripción de las principales clases y operaciones a utilizar

Clases de Ajuste

Nombre: GestajusteinventarioController	
Tipo de clase: Controladora	
Atributo	Tipo
\$model	GestajusteinventarioModel
\$comun	comunModel
\$modelB	GestBloqueoModel
Para cada responsabilidad:	
Nombre:	Descripción:
init ()	Constructor de la clase.
gestAjusteInventarioAction()	Responsabilidad encargada de validar todos los datos necesarios para gestionar los Ajuste y renderizar la plantilla.
verificarBloqueoAction()	Responsabilidad encargada de verificar si el documento se encuentra bloqueado.
desbloquearDocAction()	Responsabilidad encargada de desbloquear el documento.
cargarComboDepositoAction()	Responsabilidad encargada de cargar el combo de los depósitos.
productosAjusteAction()	Responsabilidad encargada de cargar los productos del Ajuste
cargarComboMotivoAction()	Responsabilidad encargada de cargar el combo de los motivos de ajuste
adicionarProdAjusteAction()	Responsabilidad encargada de insertar los productos al documento
eliminarProductoAction()	Responsabilidad encargada de eliminar los productos del documento
crearProductoAction()	Responsabilidad encargada de adicionar la cantidad que se va a ajustar
adicionarImporteAction()	Responsabilidad encargada de adicionar el importe en el caso que el motivo de ajuste sea error de importe (+ -)
adicionarPrecioAction()	Responsabilidad encargada de adicionar el precio al ajuste
cargarEncabezadoAdicionarAction()	Responsabilidad encargada de cargar el encabezado del documento que se va a adicionar
cargarencabezadomodificarAction()	Responsabilidad encargada de cargar el encabezado del documento que se va a modificar
cargarAjustesAction()	Responsabilidad encargada de cargar los documento de ajuste.
crearAjusteAction()	Responsabilidad encargada de adicionar un nuevo ajuste
modificarAjusteAction()	Responsabilidad encargada de modificar el ajuste seleccionado
eliminarAjusteAction()	Responsabilidad encargada de eliminar el ajuste
contabilizarAction()	Responsabilidad encargada de contabilizar el documento
confirmarAction()	Responsabilidad encargada de confirmar el documento
cancelarEstadoAjusteAction()	Responsabilidad encargada de cancelar el estado del documento

CAPÍTULO II. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

aprobarAction()	Responsabilidad encargada de aprobar el documento
noaprobarAction()	Responsabilidad encargada de no aprobar el documento
cargarGridDocsAction()	Responsabilidad encargada de cargar el reporte del ajuste
cargarComboAprobadoPorAction()	Responsabilidad encargada de cargar el combo de los usuarios que han aprobado documentos de ajuste
cargarCreadosAction()	Responsabilidad encargada de cargar el combo de los usuarios que han creado documentos de ajuste

Tabla 3 Clases de Ajuste

Clases de Baja

Nombre: GestbajainventarioController	
Tipo de clase: Controladora	
Atributo	Tipo
\$model	GestbajainventarioModel
\$comun	comunModel
\$modelB	GestBloqueoModel
Para cada responsabilidad:	
Nombre:	Descripción:
init ()	Constructor de la clase.
gestBajaInventarioAction()	Responsabilidad encargada de validar todos los datos necesarios para gestionar la Baja y renderizar la plantilla.
cargarGridDocsAction()	Responsabilidad encargada de cargar el reporte de la Baja
adicionarProdBajaAction()	Responsabilidad encargada de adicionar un producto a un documento
cargarEncabezadoAdicionarAction()	Responsabilidad encargada de cargar el encabezado del documento que se va a adicionar
cargarEncabezadoModificarAction()	Responsabilidad encargada de cargar el encabezado del documento que se va a modificar
cargarBajaAction()	Responsabilidad encargada de cargar los documentos de Baja
crearBajaAction()	Responsabilidad encargada de adicionar una nueva baja
modificarBajaAction()	Responsabilidad encargada de modificar una baja
eliminarBajaAction()	Responsabilidad encargada de eliminar una baja
contabilizarAction()	Responsabilidad encargada de contabilizar una baja
confirmarAction()	Responsabilidad encargada de confirmar una baja
cancelarEstadoBajaAction()	Responsabilidad encargada de cancelar el estado a una baja
aprobarAction()	Responsabilidad encargada de aprobar una baja
noaprobarAction()	Responsabilidad encargada de no aprobar una baja
cargarComboDestinoAction	Responsabilidad encargada de cargar el combo de los destinos finales de la baja

cargarComboAprobadoPorAction()	Responsabilidad encargada de cargar el combo de los usuarios que han aprobado documentos de baja
cargarCreadosAction()	Responsabilidad encargada de cargar el combo de los usuarios que han creado documentos de baja
cargarNrosSeriesAction()	Responsabilidad encargada de cargar los números de series que tiene registrado el producto
cargarNrosSeriesOtroAction()	Responsabilidad encargada de cargar los números de series que fueron registrador para el producto por una recepción o una apertura
addEquipoAction()	Responsabilidad encargada de adicionar números de series a los productos que por su condición los requiera
deleteEquipoAction()	Responsabilidad encargada de eliminar los números de series que fueron registrados por una baja
cargarComboDepositoAction	Responsabilidad encargada de cargar el combo de los depósitos.
verificarBloqueoAction()	Responsabilidad encargada de verificar si el documento se encuentra bloqueado.
desbloquearDocAction()	Responsabilidad encargada de desbloquear el documento.

Tabla 4 Clases de Baja

2.5 Conclusiones parciales

Con la realización de este capítulo, se arribó a la conclusión, que los artefactos propuestos por los analistas permitieron adquirir una comprensión de todo lo relacionado con las funcionalidades a implementar. Además, la importancia de los estilos de código que hacen más legible el programa fuente, lo cual ayuda en la comunicación entre desarrolladores, y permite una temprana detección de errores. También se aborda en este capítulo las principales clases que fueron utilizadas así como sus funcionalidades. Se dio además una explicación de los componentes que fueron reutilizados para aumentar la comprensión de la solución propuesta. Dentro de las cosas más importantes del capítulo se encuentra lo referente a la integración de los componentes mediante el IoC y la arquitectura utilizada, así como los objetos de automatización y la información que se maneja, obteniéndose los módulos de ajuste y baja con todas las funcionalidades detectadas por los analistas en la captura de requisitos.

CAPÍTULO III. VALIDACIÓN DE LA SOLUCIÓN

PROPUESTA.

3.1 Introducción

En la implementación de los sistemas informáticos juega un papel fundamental el uso de las técnicas de evaluación dinámica o pruebas. Para lograr esto se debe llevar a cabo estrategias a la hora de evaluar dinámicamente el software, ya que para la evaluación del mismo se debe comenzar por los componentes más simples y más pequeños e ir avanzando progresivamente hasta probar todo el software en su conjunto. Es por eso que la técnica utilizada es las Pruebas de Unidad ya que son una forma de probar el correcto funcionamiento de un módulo de código. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado. En este capítulo se realizará la validación a la solución propuesta en el capítulo anterior, de manera que se pueda comprobar la eficiencia de las clases u operaciones utilizadas para dar respuesta a los distintos requisitos planteados por el cliente.

3.2 Pruebas de Software

Pruebas de software es el conjunto de técnicas que permiten determinar la calidad de un producto software. Las Pruebas de software se integran dentro de las diferentes fases del ciclo del software dentro de la Ingeniería de software. La prueba es el proceso de ejecución de un programa con la intención de descubrir un error. La calidad de un sistema software es algo subjetivo que depende del contexto y del objeto que se pretenda conseguir. Para determinar dicho nivel de calidad se deben efectuar unas medidas o pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema. Una prueba tiene éxito si descubre un error no detectado hasta entonces.

3.5.3 Objetivos

El objetivo de las pruebas de un programa es el de detectar todo posible malfuncionamiento antes de que entre en producción. Un error detectado en el laboratorio puede ser costoso de reparar; pero siempre es peor que el error le aparezca al usuario final. En esta idea, un proceso de pruebas será de mayor calidad cuantos menos errores queden por descubrir tras haberla pasado. Y, viceversa, si un programa aún tiene muchos fallos tras haber superado este proceso de pruebas, se dirá que estas pruebas fueron de poca calidad. *“La prueba no puede asegurar la ausencia de defectos: sólo puede demostrar que existen defectos en el software”.*

3.5.4 Alcance

Se lleva a cabo la prueba y se evalúan los resultados obtenidos frente a los resultados esperados. Si se descubren datos erróneos implica que hay un error y hay que corregirlo y empieza el proceso de depuración de errores. Se basa en las estructuras de control del diseño procedimental para generar los casos de prueba que:

- ✓ *Garanticen que se recorren por lo menos una vez todos los caminos independientes de cada módulo.*
- ✓ *Se ejecutan todas las decisiones lógicas en su parte verdadera y en su parte falsa.*
- ✓ *Se recorren todos los bucles.*
- ✓ *Se utilizan las estructuras de datos internas para garantizar su validez.*
- ✓ *Se invierte tiempo y esfuerzo en los detalles de control debido a que:*
- ✓ *Los errores suelen estar en situaciones fuera de las normales.*
- ✓ *A menudo caminos que se piensa que tienen pocas posibilidades de recorrerse, son recorridos regularmente.*
- ✓ *Los errores tipográficos son aleatorios. Puede que no sean detectados por los procesadores de la sintaxis del lenguaje particular y estar presentes en cualquier camino lógico.*

3.3 Pruebas de Unidad

La prueba de unidad se centra en el módulo. Usando la descripción del diseño detallado como guía, se prueban los caminos de control importantes con el fin de descubrir errores dentro del ámbito del módulo. La prueba de unidad hace uso intensivo de las técnicas de prueba de caja blanca aunque para realizar uno de estas pruebas es necesario probar el flujo de datos desde la interfaz del componente (prueba de caja negra). Estas unidades pueden ser clases, métodos, propiedades, componentes entre otros. Si los datos no se comportan correctamente al ser introducidos en la aplicación, todas las demás pruebas no tienen sentido.

A continuación un breve comentario acerca de estos tipos de pruebas:

Técnicas de caja blanca o estructurales: se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa.

Técnicas de caja negra o funcionales: realizan pruebas sobre la interfaz del programa a probar, entendiendo por interfaz las entradas y salidas de dicho programa. No es necesario conocer la lógica del programa, únicamente la funcionalidad que debe realizar.

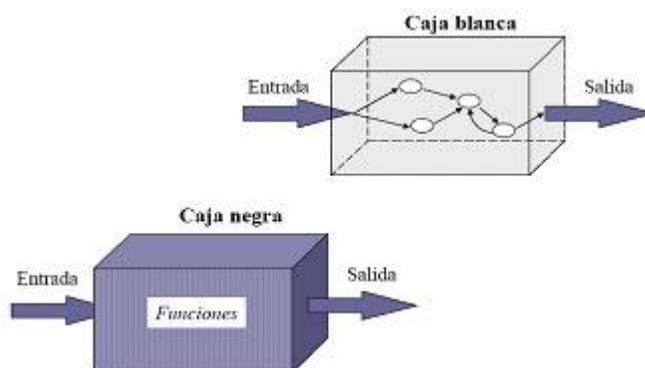


Figura 9 Pruebas de Caja blanca y Caja negra

3.3.1 Prueba de Caja Blanca o Estructurales

Permiten examinar la estructura interna del programa. Se diseñan casos de prueba para examinar la lógica del programa. Este método se centra en cómo diseñar los casos de prueba atendiendo al comportamiento interno y la estructura del programa. Se examina así la lógica interna del programa sin considerar los aspectos de rendimiento.

Es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivar casos de prueba que garanticen que:

- *Se ejercitan todos los caminos independientes de cada módulo.*
- *Se ejercitan todas las decisiones lógicas.*
- *Se ejecutan todos los bucles.*
- *Se ejecutan las estructuras de datos internas.*

Entre algunas de las técnicas de prueba de Caja Blanca podemos citar:

1. **Prueba de Condición**

Es un método de diseño de casos de prueba que ejercita las condiciones lógicas contenidas en el módulo de un programa.

2. **Prueba de Flujo de Datos**

Se selecciona caminos de prueba de un programa de acuerdo con la ubicación de las definiciones y los usos de las variables del programa.

3. **Prueba de Bucles**

Es una técnica de prueba de caja blanca que se centra exclusivamente en la validez de las construcciones de bucles.

4. **Prueba del Camino Básico**

Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el Grafo de Flujo asociado y se calcula su complejidad ciclomática.

Estas técnicas permiten obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico.

3.3.2 **Prueba de Caja Negra o Funcionales**

Las pruebas se llevan a cabo sobre la interfaz del software, y es completamente indiferente el comportamiento interno y la estructura del programa.

Los casos de prueba de la caja negra pretende demostrar que:

- *Las funciones del software son operativas.*
- *La entrada se acepta de forma adecuada.*
- *Se produce una salida correcta.*
- *La integridad de la información externa se mantiene.*

Se derivan conjuntos de condiciones de entrada que ejerciten completamente todos los requerimientos funcionales del programa.

La prueba de la caja negra intenta encontrar errores de las siguientes categorías:

- *Funciones incorrectas o ausentes.*
- *Errores de interfaz.*
- *Errores en estructuras de datos o en accesos a bases de datos externas.*
- *Errores de rendimiento.*
- *Errores de inicialización y de terminación.*

Entre algunas de las técnicas utilizadas en la prueba de caja negra pueden encontrarse:

1. **Partición de Equivalencia**

Técnica que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. La partición equivalente se dirige a una definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que desarrollar.

2. **Análisis de Valores Límite**

La experiencia muestra que los casos de prueba que exploran las condiciones límite producen mejor resultado que aquellos que no lo hacen. Las condiciones límite son aquellas que se hayan en los márgenes de la clase de equivalencia, tanto de entrada como de salida. Por ello, se ha desarrollado el *análisis de valores límite* como técnica de prueba. Esta técnica nos lleva a elegir los casos de prueba que ejerciten los valores límite.

Las pautas para desarrollar casos de prueba con esta técnica son:

- ✓ Si una condición de entrada especifica un rango de valores, se diseñarán casos de prueba para los dos límites del rango, y otros dos casos para situaciones justo por debajo y por encima de los extremos.
- ✓ Si una condición de entrada especifica un número de valores, se diseñan dos casos de prueba para los valores mínimo y máximo, además de otros dos casos de prueba para valores justo por encima del máximo y justo por debajo del mínimo.
- ✓ Aplicar las reglas anteriores a los datos de salida.

- ✓ Si la entrada o salida de un programa es un conjunto ordenado, habrá que prestar atención a los elementos primero y último del conjunto.

3. Grafos de Causa-Efecto

Es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

3.4 Aplicación de pruebas de caja blanca

Para conocer la complejidad del algoritmo es necesario calcular la complejidad ciclomática del proceso al cual se le va a aplicar la prueba, para hacer dicho cálculo es necesario primero tener el código o el diseño del algoritmo, luego enmarcar cada instrucción del código con un número, que representa cada lugar del camino que puede seguir la secuencia del algoritmo, a continuación se representa el código con sus instrucciones enmarcadas.

Después de este paso, es necesario representar el grafo de flujo asociado, en el cual se representan distintos componentes como es el caso de:

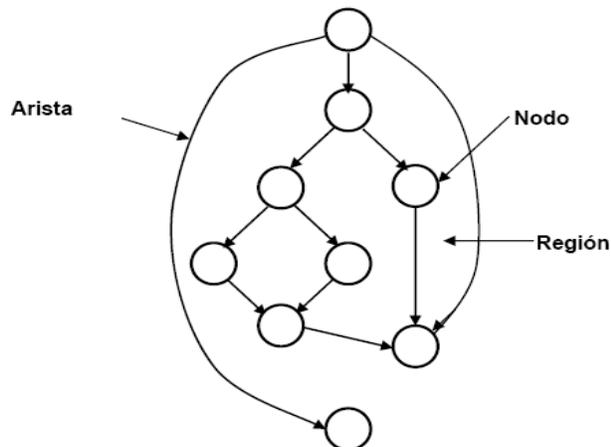


Figura 10 Componentes de los grafos de flujo

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

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

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

```

public function Contabilizar($iddocumento,$version)
{
    $prod= array(); 1
    $operac = '-'; 1
    $prod = $this->pIntegrator->movimientos->Existencia($iddocumento); 1
    foreach ($prod as $index => $temp) 2
    {
        $result= $this->pIntegrator->productos->obtenerProducto()->obtenerProducto($temp->idproducto); 3
        $idproducto=$temp->idproducto; 3
        $cantidad=$temp->cantidad; 3
        $existencia=$result[0]['existencia']; 3
        $signo=$operac; 3
        $importetotalprod=$result[0]['importetotal']; 3
        $cronologia=$this->pIntegrator->documentos->DevolverCronologia(); 3
        $datos=$this->pIntegrator->movimientos->UpdateMovContabilizar($temp->idmovimiento,$cantidad,$operac); 3
        $importetotalprod,$result[0]['preciopromedio'],$existencia,$cronologia); 3
        $arrprod[$index]['idproducto']=$temp->idproducto; 3
        $arrprod[$index]['cantidad']=$datos->cantidad; 3
        $arrprod[$index]['cantidaddisponible']=$result[0]['cantidaddisponible']; 3
        $arrprod[$index]['importetotal']=$datos->importetotal; 3
        $arrprod[$index]['preciopromedio']=$datos->preciopromedio; 3
    } 4
    $ok= $this->pIntegrator->productos->contabilizarProd($arrprod,$operac); 5
    if($ok==3) 6
    {
        foreach ($prod as $index => $temp) 7
        {
            $movimiento = $this->pIntegrator->movimientos->BuscarMov($iddocumento,$temp->idproducto); 8
            $equipos=$this->pIntegrator->movimientos->Buscar_ME($movimiento->idmovimiento); 8
            if($equipos) 9
            {
                foreach ($equipos as $index => $equip) 10
                {
                    $this->pIntegrator->utilesLogistica->ActualizaBaja($equip->idequipo); 11
                } 12
            } 13
        } 14
        $this->pIntegrator->documentos->ContabilizarDocBaja($iddocumento,$version); 15
    } 16
    return $ok; 17
} 18

```

Figura 11 Representación del algoritmo *Contabilizar()*

Seguidamente se construye el grafo de flujo asociado al código anterior.

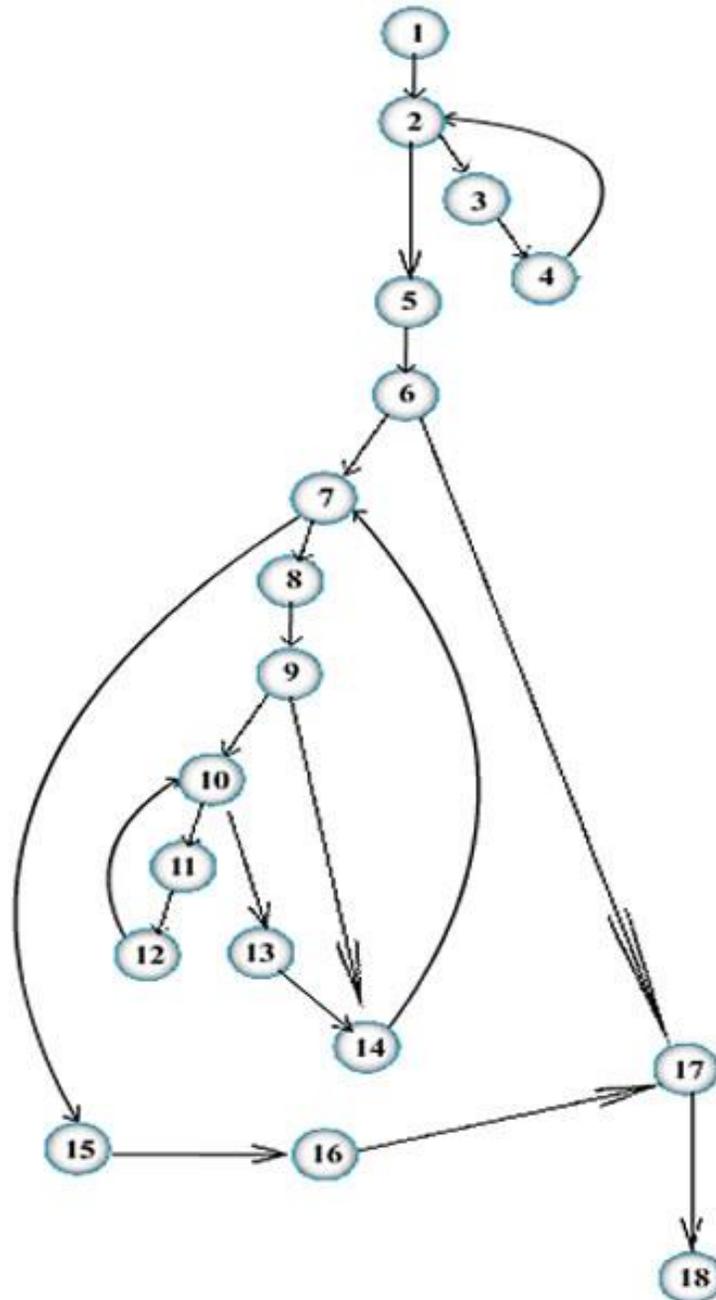


Figura 12 Grafo de flujo asociado al algoritmo *Contabilizar()*

Luego de haber construido el grafo, se realiza el cálculo de la complejidad ciclomática, mediante las tres fórmulas descritas a continuación, las cuales tienen que arrojar el mismo resultado para asegurar que el cálculo de la complejidad es correcto.

Fórmulas para calcular complejidad ciclomática:

1. $V(G) = (A - N) + 2.$

Siendo “A” la cantidad total de aristas y “N” la cantidad total de nodos.

2. $V(G) = P + 1.$

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

3. $V(G) = R.$

Siendo “R” la cantidad total de regiones, para cada fórmula “V (G)” representa el valor del cálculo.

Aplicando estas fórmulas al grafo de flujo de la figura 13 se obtienen los siguientes resultados:

Calculando mediante la fórmula 1:

$$V(G) = (22 - 18) + 2$$

$$V(G) = 6.$$

Calculando mediante la fórmula 2:

$$V(G) = 5 + 1$$

$$V(G) = 6.$$

Calculando mediante la fórmula 3:

$$V(G) = 6.$$

El cálculo efectuado mediante las tres fórmulas ha dado el mismo valor, por lo que se puede decir que la complejidad ciclomática del código es de 6, lo que significa que existen a lo sumo seis posibles caminos por donde el flujo puede circular, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento tratado.

Seguidamente es necesario representar los caminos básicos por los que puede recorrer el flujo.

Camino básico #1

1-2-5-6-17-18

Camino básico #2

1-2-3-4-2-5-6-17-18

Camino básico #3

1-2-3-4-2-5-6-7-8-9-14-7-15-16-17-18

Camino básico #4

1-2-3-4-2-5-6-7-8-9-10-11-12-10-13-14-7-15-16-17-18

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

Para realizar los casos de pruebas es necesario cumplir con las siguientes exigencias:

Descripción: Se hace la entrada de datos necesaria, validando que ningún parámetro obligatorio pase nulo al procedimiento o no se entre algún dato erróneo.

Condición de ejecución: Se especifica cada parámetro para que cumpla una condición deseada para ver el funcionamiento del procedimiento.

Entrada: Se muestran los parámetros que entran al procedimiento

Resultados Esperados: Se expone el resultado que se espera que devuelva el procedimiento.

Caso de prueba para el camino básico # 1.

Descripción: Los datos de entrada cumplirán con los siguientes requisitos: El id del documento y la versión son números enteros.

Condición de ejecución: El id del documento será igual 900000012 y la versión igual a 1.

Entrada: \$iddocumento = 900000012, \$version = 1.

Resultados esperados: Se espera que se lance un mensaje con el siguiente texto: “No se pudo contabilizar, el documento no tiene producto”.

Caso de prueba para el camino básico # 2.

Descripción: Los datos de entrada cumplirán con los siguientes requisitos: El id del documento y la versión son números enteros.

Condición de ejecución: El id del documento será igual 900000013 y la versión igual a 1.

Entrada: \$iddocumento = 900000013, \$version = 1.

Resultados esperados: Se espera que el documento no sea contabilizado, debido a alguna falla en la hora de actualizar los valores de los productos asociados al documento.

Caso de prueba para el camino básico # 3.

Descripción: Los datos de entrada cumplirán con los siguientes requisitos: El id del documento y la versión son números enteros.

Condición de ejecución: El id del documento será igual 900000014 y la versión igual a 1.

Entrada: \$iddocumento = 900000014, \$version = 1.

Resultados esperados: Se espera que el documento sea contabilizado con éxito.

El documento fue contabilizado y no tiene ningún equipo asociado para actualizarle los valores.

Caso de prueba para el camino básico # 4.

Descripción: Los datos de entrada cumplirán con los siguientes requisitos: El id del documento y la versión son números enteros.

Condición de ejecución: El id del documento será igual 900000015 y la versión igual a 1.

Entrada: \$iddocumento = 900000015, \$version = 1.

Resultados esperados: Se espera que el documento sea contabilizado con éxito.

El documento fue contabilizado y los equipos asociados fueron actualizados satisfactoriamente.

3.5 Evaluación de la calidad utilizando métricas

Aunque los términos medida, medición y métricas se utilizan a menudo indistintamente, existe gran confusión a la hora de referirnos a ellos. Dentro del contexto de la ingeniería del software, una medida proporciona una indicación cuantitativa de extensión, cantidad, dimensiones, capacidad y tamaño de algunos atributos de un proceso o producto.

Las métricas de software se centran en cuantificar tanto la complejidad, como la funcionalidad y eficiencia inmersa en el desarrollo de software, inclinando sus objetivos a mejorar la comprensión de la calidad del producto, estimar efectividad del proceso y mejorar la calidad del trabajo.

Atributos de calidad que se abarcan:

1. Responsabilidad o Cohesión. Consiste en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
2. Complejidad del diseño. Consiste en la complejidad que posee una estructura de diseño de clases.

3. Complejidad de implementación. Consiste en el grado de dificultad que tiene implementar un diseño de clases determinado.
4. Reutilización. Consiste en el grado de reutilización de presente en una clase o estructura de clase, dentro de un diseño de software.
5. Acoplamiento. Consiste en el grado de dependencia o interconexión de una clase o estructura de clase, con otras, está muy ligada a la característica de Reutilización.
6. Complejidad del mantenimiento. Consiste en el grado de esfuerzo necesario a realizar para desarrollar un arreglo, una mejora o una rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costes y la planificación del proyecto.
7. Cantidad de pruebas. Consiste en el número o el grado de esfuerzo para realizar las pruebas de calidad (Unidad) del producto (Componente, módulo, clase, conjunto de clases, etc.) diseñado.
8. Nivel de Cohesión. Consiste en el grado de especialización de las clases concebidas para modelar un dominio o concepto específico.
9. Abstracción del diseño. Consiste en la capacidad de modelar lo más cercano posible a la realidad un concepto o dominio determinado.

Las métricas concebidas como instrumento para evaluar la calidad de los Módulos de Ajuste y Baja y su relación con los atributos de calidad definidos en este trabajo son las siguientes:

De la serie Chidamber y Kemner:

- Profundidad de Herencia (PH).
- Número de Descendiente (ND).
- Respuesta para una Clase (RC).

Se utilizaron de la serie definida por Lorenz y Kidd dos métricas:

- Tamaño Operacional de Clases (TOC).
- Número de Operaciones Redefinidas para una Sub-Clase (NOR).

Tamaño operacional de clase (TOC): Está dado por el número de métodos asignados una clase.

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

Figura 13 Tamaño operacional de clase (TOC)

Relaciones entre clases (RC): Está dado por el número de relaciones de uso de una clase con otras.

Atributo que afecta	Modo en que lo afecta
Acoplamiento	Un aumento del RC implica un aumento del Acoplamiento de la clase.
Complejidad del 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.

Figura 14 Relaciones entre clases (RC)

Profundidad de Herencia (PH): Está dado por la profundidad en la herencia de las clases heredadas de un nodo padre.

Atributo que afecta	Modo en que lo afecta
Complejidad del mantenimiento	Un aumento del PH implica una disminución de la complejidad del mantenimiento de la clase.
Complejidad del diseño	Un aumento del PH implica una disminución de la complejidad del diseño de la clase.
Reutilización	Un aumento del PH implica un aumento en el grado de reutilización de la clase.

Figura 15 Profundidad de Herencia (PH)

Número de Descendientes (ND): Está dado por la cantidad de clases que heredan de un padre.

Atributo que afecta	Modo en que lo afecta
Abstracción del diseño	Un aumento del ND puede diluir la abstracción del diseño de clases.
Cantidad de pruebas	Un aumento del ND implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.
Reutilización	Un aumento del ND implica un aumento en el grado de reutilización de la clase.
Nivel de Cohesión	Un aumento del ND implica una disminución en el nivel de cohesión de la clase.

Figura 16 Número de Descendientes (ND)

Número de Operaciones Redefinidas para una clase hija (NOR): Está dado por la cantidad de operaciones redefinidas en cada clase hija.

Atributo que afecta	Modo en que lo afecta
Abstracción del diseño	Un aumento del NOR implica que se ha logrado una buena definición de la abstracción del diseño de clases.
Cantidad de pruebas	Un aumento del NOR implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.
Complejidad del mantenimiento	Un aumento del NOR implica un aumento de la complejidad del mantenimiento de la clase.

Figura 17 Número de Operaciones Redefinidas para una clase hija (NOR)

A continuación se presentan los resultados obtenidos de la aplicación de los instrumentos de evaluación para medir las métricas anteriormente descritas, y una valoración de los mismos.

3.5.1 Resultados del instrumento de evaluación de la métrica Tamaño operacional de clase (TOC)

Ver instrumentos y tabla de resultados en ([Anexo 3 Instrumento de medición de la métrica Tamaño operacional de clase \(TOC\)](#)).

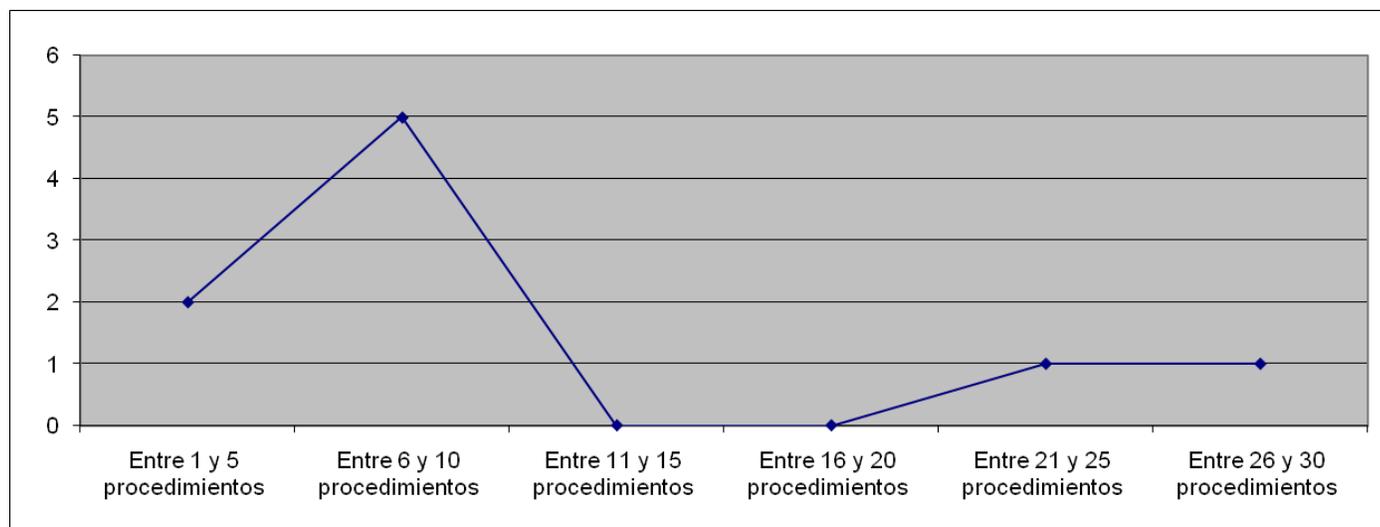


Figura 18 Representación de los resultados obtenidos en el instrumento agrupados en los intervalos definidos

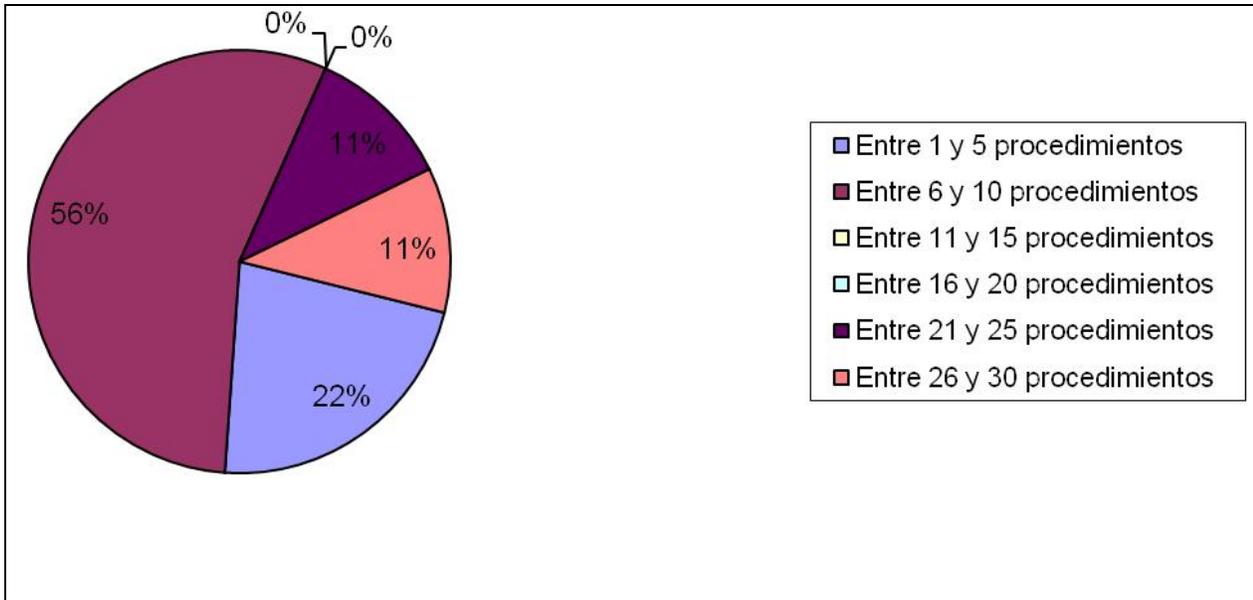


Figura 19 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos

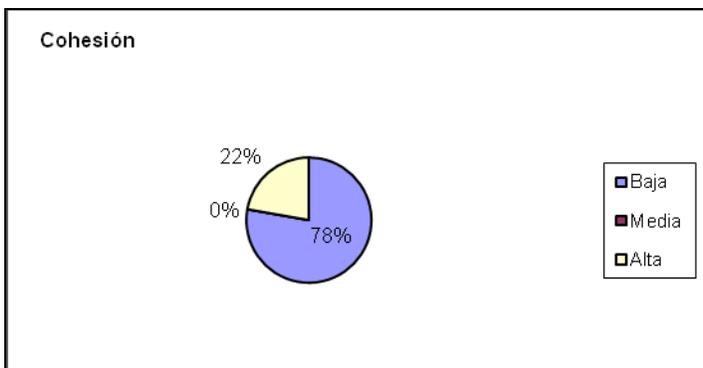


Figura 20 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Cohesión

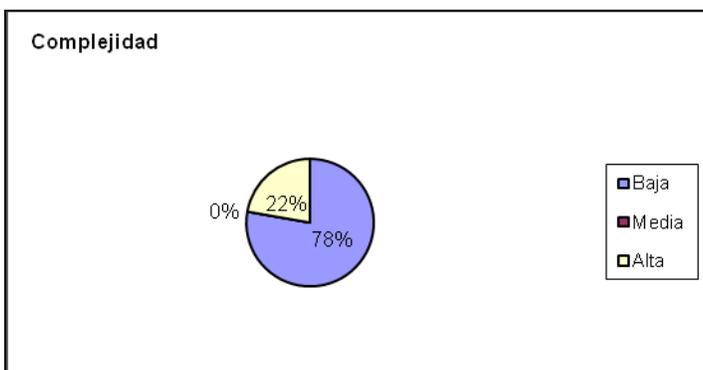


Figura 21 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de Implementación

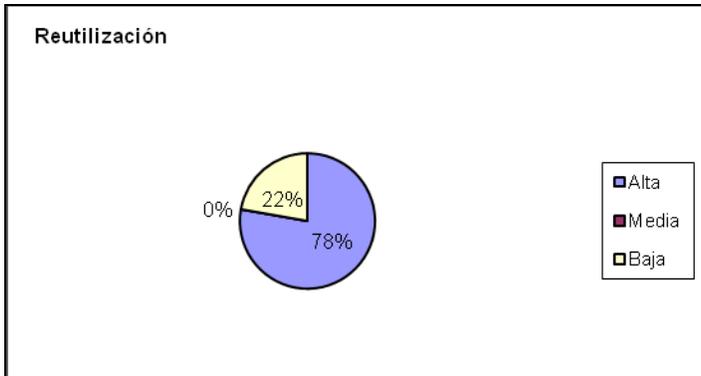


Figura 22 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica TOC, se puede concluir que la implementación de los subsistemas Ajuste de Inventario y Baja Técnica tienen una calidad buena teniendo en cuenta que el 78 % de las clases incluidas en estos subsistemas posee menos cantidad de operaciones que la mitad del valor máximo registrado en las mediciones. Además el 78% de las clases poseen evaluaciones positivas en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización).

3.5.2 Resultados del instrumento de evaluación de la métrica Relaciones entre Clases (RC)

Ver instrumentos y tabla de resultados en ([Anexo 4 Instrumento de medición de la métrica Relaciones entre clases \(RC\)](#)).

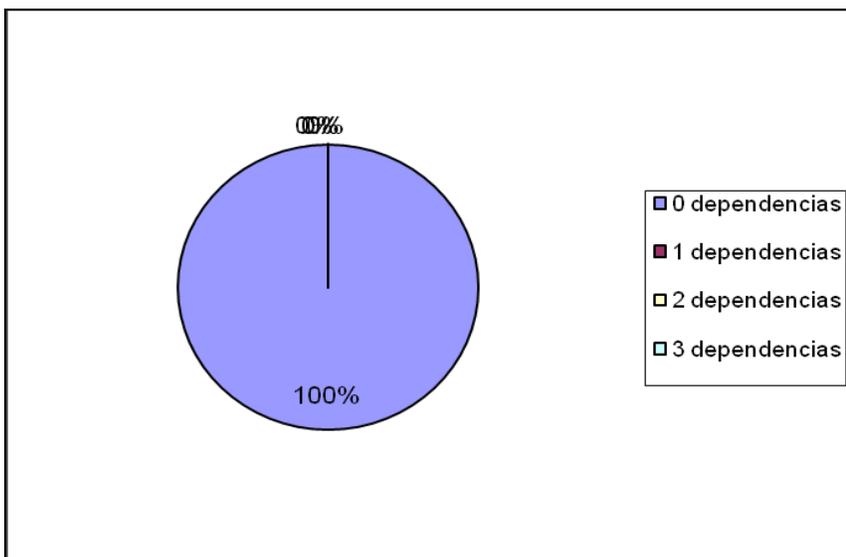


Figura 23 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos



Figura 24 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento



Figura 25 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de Mantenimiento

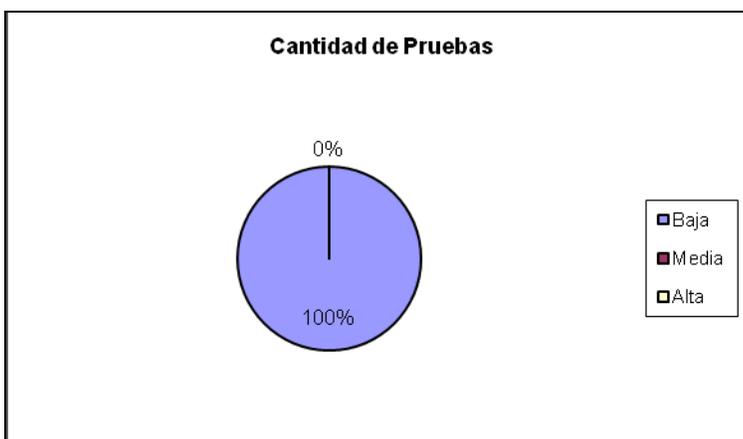


Figura 26 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de Pruebas

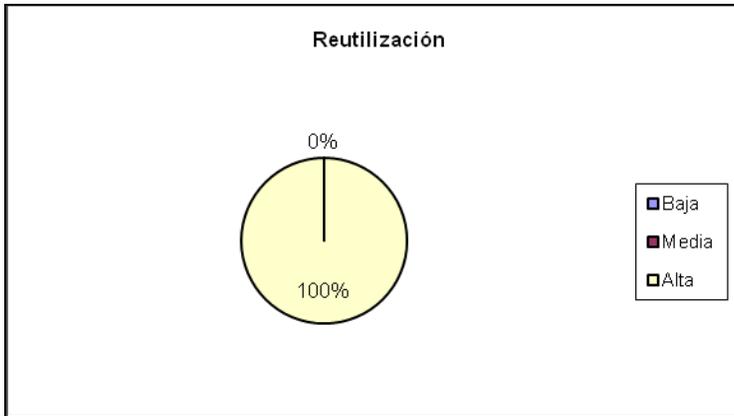


Figura 27 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica RC, se puede concluir que la implementación de los subsistemas Ajuste de Inventario y Baja Técnica tienen una calidad buena teniendo en cuenta que el 100 % de las clases incluidas en estos subsistemas posee 0 dependencias de otras clases. Además el 100% de las clases no poseen acoplamiento con otras. Así mismo los atributos de calidad Complejidad de Mantenimiento, Cantidad de Pruebas y Reutilización se comportan satisfactoriamente en un 100 % de las clases.

3.5.3 Resultados del instrumento de evaluación de la métrica Profundidad de Herencia (PH)

Ver instrumentos y tabla de resultados en ([Anexo 5 Instrumento de medición de la métrica Profundidad de Herencia \(PH\)](#)).

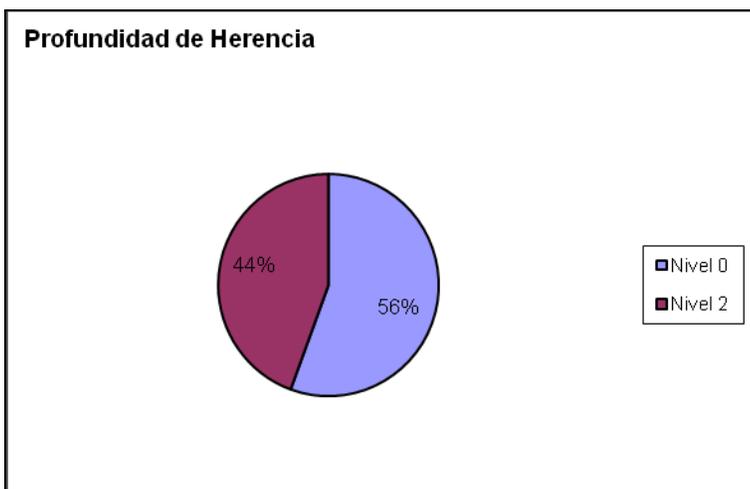


Figura 28 Representación en % de los resultados obtenidos en el instrumento agrupados por nivel



Figura 29 Representación de la incidencia de los resultados de la evaluación de la métrica PH en el atributo Complejidad de Mantenimiento

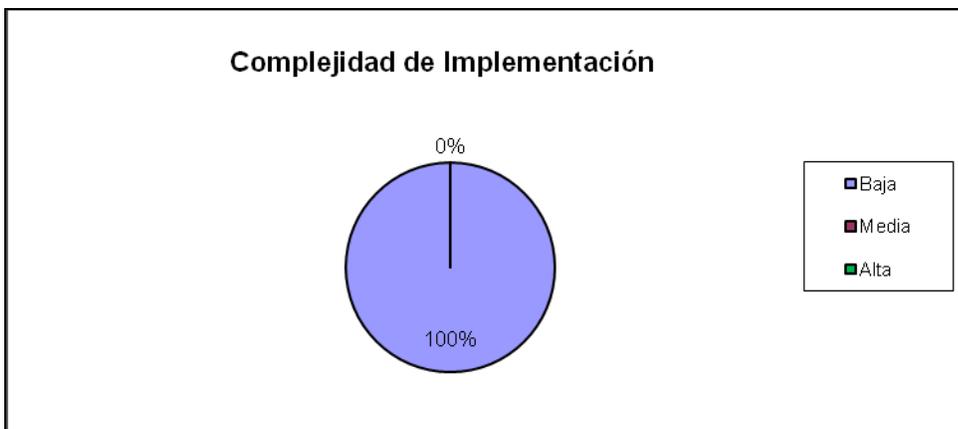


Figura 30 Representación de la incidencia de los resultados de la evaluación de la métrica PH en el atributo Complejidad de Implementación



Figura 31 Representación de la incidencia de los resultados de la evaluación de la métrica PH en el atributo Reutilización

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica PH, se puede concluir que la implementación de los subsistemas Ajuste de Inventario y Baja Técnica tienen una calidad buena teniendo en cuenta que la profundidad de herencia presente en los subsistemas desarrollados es de 2 y en el caso de las clases de

validación no tienen herencia. Analizando los atributos de calidad Complejidad de Mantenimiento, Complejidad de implementación se puede decir que sin duda se tienen buenos índices debido a que solo se cuenta como máximo con un nivel 2 de profundidad en la herencia. Sin embargo el atributo Reutilización no posee indicadores positivos debido al bajo nivel de profundidad de herencia.

3.5.4 Resultados del instrumento de evaluación de la métrica Número de Descendientes (ND)

Ver instrumentos y tabla de resultados en ([Anexo 6 Instrumento de medición de la métrica Número de Descendientes \(ND\)](#)).

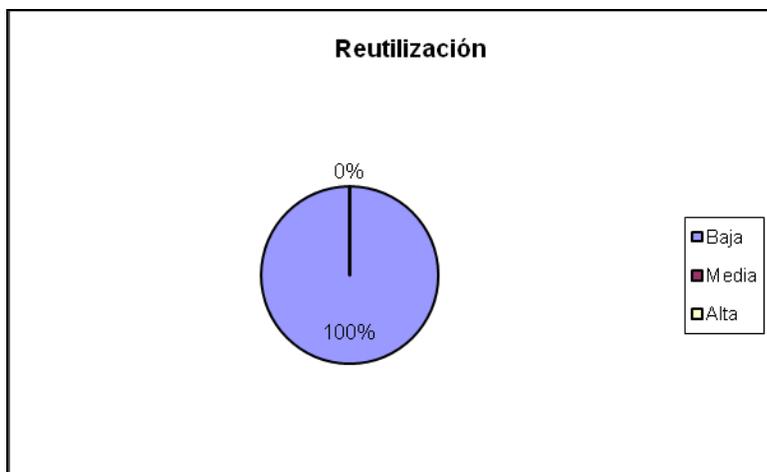


Figura 32 Representación de la incidencia de los resultados de la evaluación de la métrica ND en el atributo Reutilización

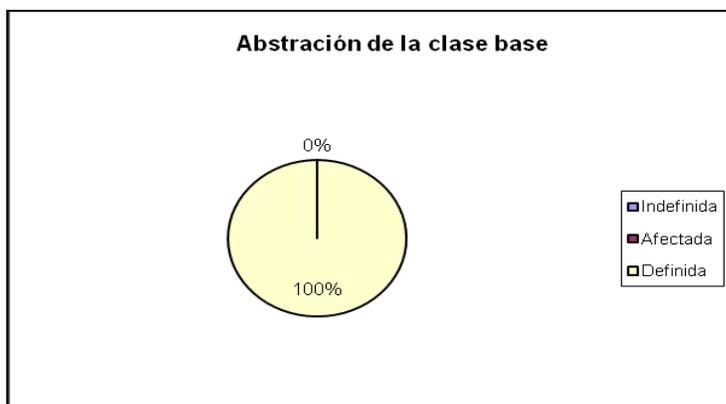


Figura 33 Representación de la incidencia de los resultados de la evaluación de la métrica ND en el atributo Abstracción de la clase base

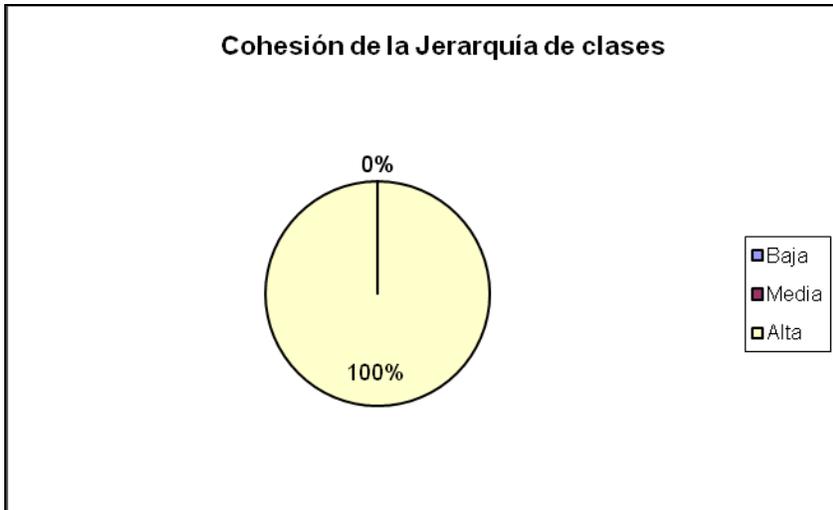


Figura 34 Representación de la incidencia de los resultados de la evaluación de la métrica ND en el atributo Cohesión de la Jerarquía de clases

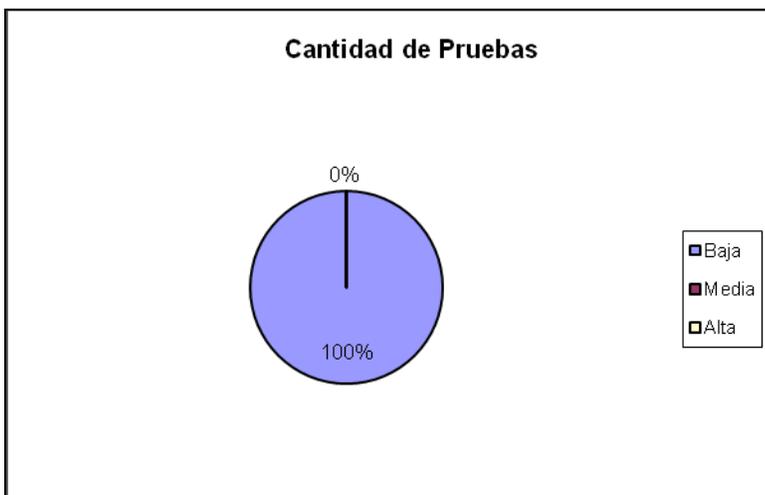


Figura 35 Representación de la incidencia de los resultados de la evaluación de la métrica ND en el atributo Cantidad de Pruebas

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica ND, se puede concluir que la implementación de los subsistemas Ajuste de Inventario y Baja Técnica tienen una calidad buena teniendo en cuenta que solo se emplea la herencia con la clase ZendExt_Model por necesidades del negocio y además no tiene descendientes. Valorando el atributo de calidad Reutilización se puede decir que por lo antes explicado los índices de Reutilización se mantienen bajos al 100%. En cuanto a los atributos Abstracción, Cohesión de la jerarquía de clases como para el atributo Cantidad de Pruebas los índices son positivos favoreciendo esto su implementación.

3.5.5 Resultados del instrumento de evaluación de la métrica Número de Operaciones Redefinidas (NOR)

Ver instrumentos y tabla de resultados en ([Anexo 7 Instrumento de medición de la métrica Número de Operaciones Redefinidas \(NOR\)](#)).



Figura 36 Representación en % de los resultados obtenidos en el instrumento agrupados en los valores existentes



Figura 37 Representación de la incidencia de los resultados de la evaluación de la métrica NOR en el atributo Complejidad del Mantenimiento

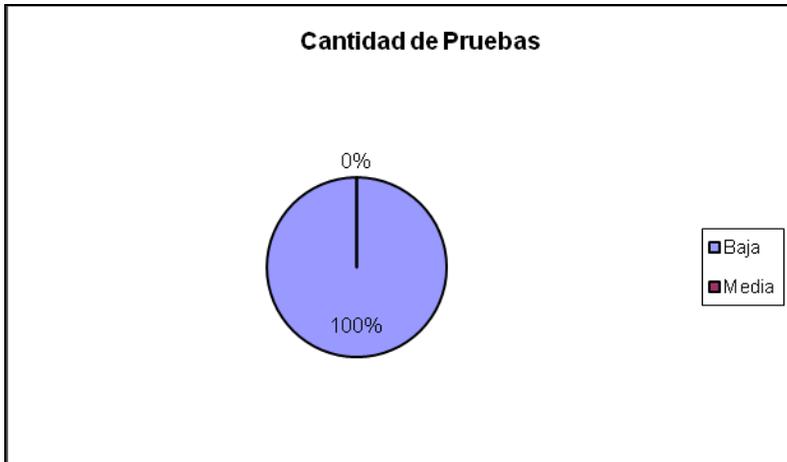


Figura 38 Representación de la incidencia de los resultados de la evaluación de la métrica NOR en el atributo Cantidad de Pruebas



Figura 39 Representación de la incidencia de los resultados de la evaluación de la métrica NOR en el atributo Violación de la Abstracción representada por la superclase

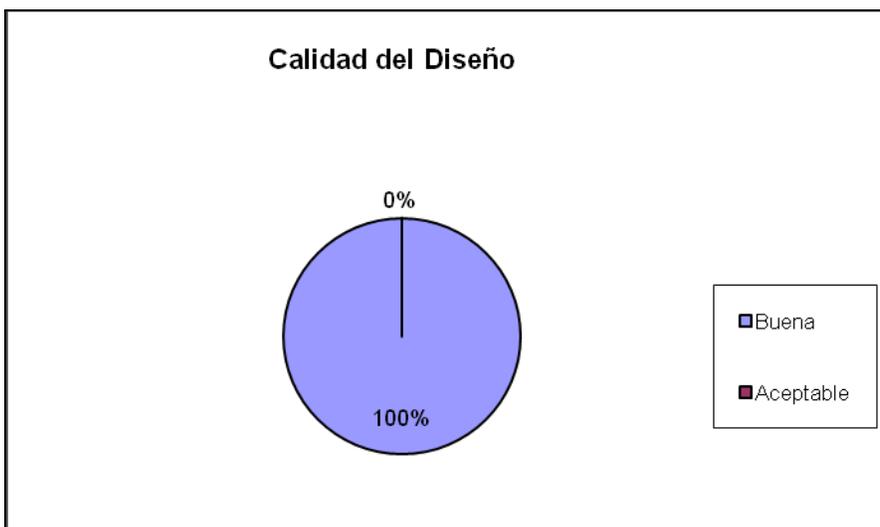


Figura 40 Representación de la incidencia de los resultados de la evaluación de la métrica NOR en el atributo Calidad del Diseño

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica NOR, se puede concluir que la implementación de los subsistemas Ajuste de Inventario y Baja Técnica tienen una calidad buena teniendo en cuenta que ninguna clase redefine métodos heredados. Además los indicadores se comportan de forma adecuada para los atributos de calidad Complejidad del Mantenimiento, Cantidad de Pruebas, Violación de la Abstracción representada por la superclase.

3.6 Matriz de inferencia de indicadores de calidad

La matriz de inferencia de indicadores de calidad permite conocer si el resultado obtenido de la relación entre los atributos y las métricas asociadas a ellos para el componente, es positivo o negativo. Es una representación estructurada de las métricas y los atributos utilizados para evaluar la calidad de la solución propuesta. Aplicando una escala numérica a estos resultados donde 1 corresponde a positivo, 0 a negativo y nulo (-) si no existe relación y realizando un cálculo donde se promedia la suma de los valores obtenidos de un atributo por cada métrica y dividiendo el resultado por la cantidad de métricas evaluadas, se pueden clasificar los atributos en buenos, regulares y malos.

Atributos	ND	RC	PH	TOC	NOR	Total
Complejidad Implementación	-	-	1	0,78	-	0,89
Reutilización	0	1	0	0,78	-	0,445
Acoplamiento	-	1	-	-	-	1
Complejidad Mantenimiento	-	1	1	-	1	1
Cantidad Pruebas	1	1	-	-	1	1
Abstracción	1	-	-	-	1	1
Cohesión	1	-	-	0,22	-	0,61
Calidad Diseño	-	-	-	-	1	1

Tabla 5 Matriz de inferencia de indicadores de calidad

Donde los valores:

0.8 a 1.0 es considerado **Bueno**

0.4 a 0.7 es considerado **Regular**

0.0 a 0.3 es considerado **Malo**

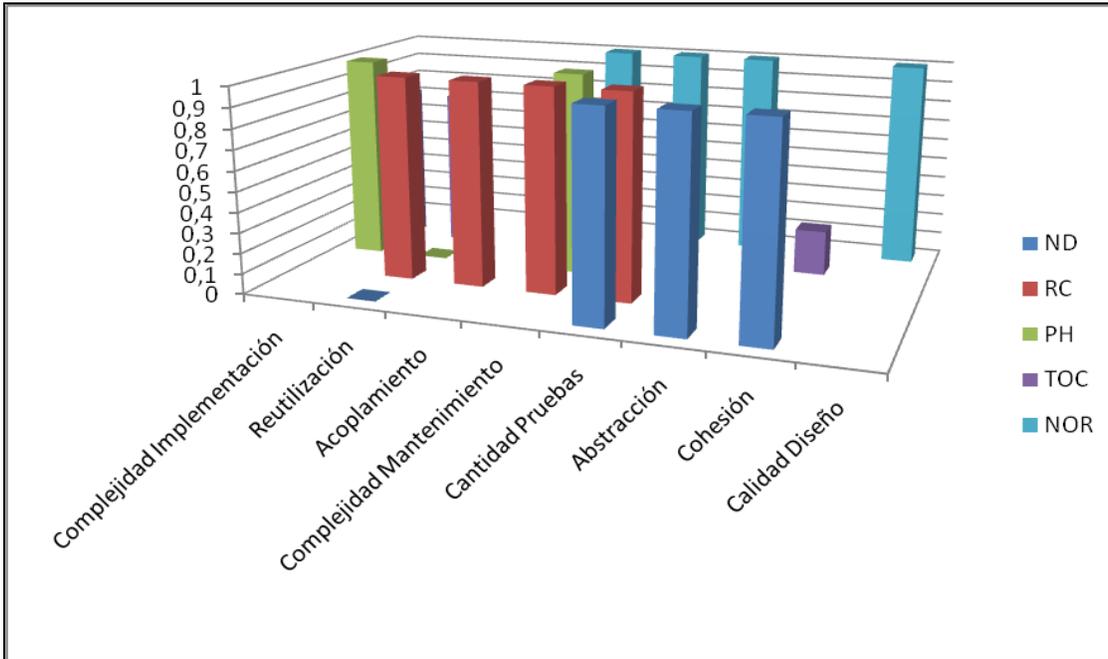


Figura 41 Gráfico resumen

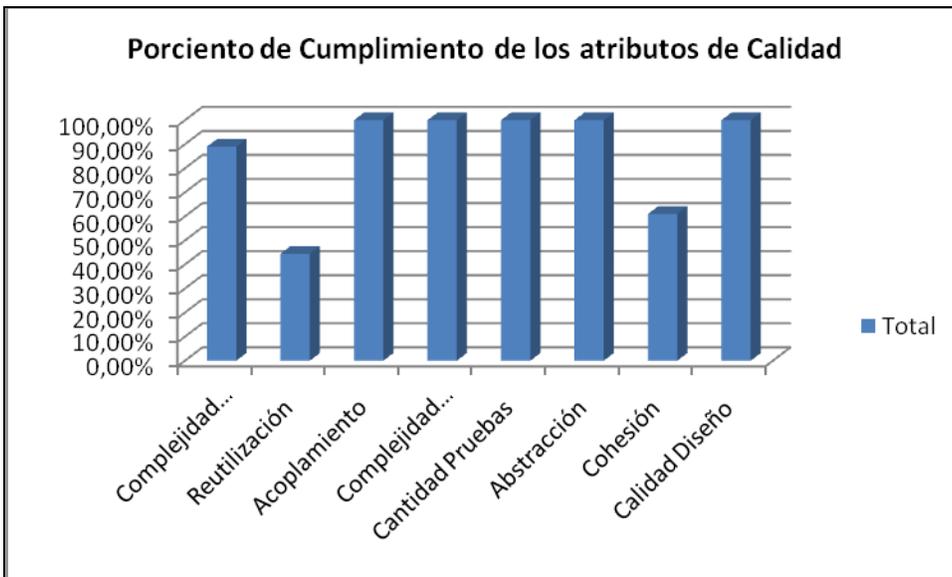


Figura 42 Porcentaje de Cumplimiento de los atributos de Calidad

Analizando la gráfica generada a partir de la matriz de inferencia de indicadores de calidad y el rango de valores para la evaluación técnica de los atributos de calidad evaluados por cada métrica se puede valorar que la solución propuesta mantiene una factible funcionalidad y valores positivos para cada uno de los atributos de calidad relacionados con las métricas evaluadas menos el atributo *reutilización*, debido a la no existencia de herencia con otras clases.

3.7 Conclusiones parciales

Con el desarrollo de este capítulo se pudo apreciar la importancia que tiene el proceso de prueba en el desarrollo de software, para el buen entendimiento del mismo se analizaron diferentes aspectos como el significado de las pruebas de software, sus objetivos y alcance, se realizó una descripción de las pruebas de unidad, dentro de los cuales se analizaron los tipos de prueba: Caja Blanca y Caja Negra, y la ejecución de una de ellas, analizando los resultados obtenidos. Se abordaron las métricas de software que fueron aplicadas en la solución. Basándose en los atributos de calidad trazados se pudo realizar una medición de los componentes de Ajuste y Baja, la cual arrojó como resultado que la solución se puede valorar de aceptable teniendo en cuenta que las métricas aplicadas arrojaron valores positivos para cada uno de los atributos de calidad utilizados.

CONCLUSIONES GENERALES

A manera de conclusión se plantea que:

- Se analizaron soluciones existentes, así como las herramientas propuestas por el equipo de arquitectura para crear las bases teóricas de la investigación.
- Se mostraron los aspectos más significativos de la solución propuesta como el análisis de reutilización de componentes, la descripción de las principales clases a utilizar, la descripción de la implementación y los estándares de codificación utilizados evidenciado la obtención de un producto funcional a partir de los requisitos propuestos por los analistas.
- Se realizó la validación de la solución propuesta mediante el diseño y aplicación de las pruebas de caja blanca para validar la calidad de la solución propuesta arrojando resultados positivos. También se hizo una evaluación de la implementación mediante métricas donde arrojaron valores aceptables en los atributos de reutilización, facilidad de mantenimiento, complejidad del diseño, complejidad de implementación, cohesión, acoplamiento y cantidad de pruebas.

Por todo lo antes mencionado se evidencia el cumplimiento de los objetivos propuestos en el presente trabajo de diploma, lo cual conlleva al cumplimiento del objetivo general.

RECOMENDACIONES

Las recomendaciones propuestas para la continuidad del presente trabajo son:

- Corregir las no conformidades detectadas en las pruebas pilotos con el objetivo de refinar la solución.
- Ampliar las funcionalidades de los módulos con los nuevos requerimientos que surjan por necesidades del cliente.

REFERENCIA BIBLIOGRÁFICA

1. Software de Gestión y Estrategia SAP. [En línea] 2009.[Citado el: 3 de Abril de 2009.]
http://www.sap.com/spain/software/gestion_de_inventarios.epx
2. Openbravo ERP . [En línea] 2007.[Citado el: 4 de Abril de 2009.] <http://www.openbravo.com/es/>
3. OpenERP Sistema de Gestión Empresarial. [En línea] [Citado el: 4 de Abril de 2009.]
<http://www.estudio42.com.ar/index.php/openerp>
4. **Dr. José Carlos del Toro, Ing. Henry Raúl González Brito.** *Documento Visión del proyecto ERP Cuba.* 2008
5. **Colectivo de Autores. 2008.** *ERP-PROD Modelo de Desarrollo orientado a componentes.* [Digital] La Habana : Universidad de las Ciencias Informáticas, 2008.
6. Herramientas Web. *El modelo cliente – servidor.* [En línea] 2007.[Citado el: 5 de Abril de 2009.]
<http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/cliente-servidor.html>
7. **Valdés, Damián Pérez .** Maestros del Web. *¿Qué es Javascript?* [En línea] 3 de Julio 2007 .[Citado el: 4 de Abril de 2009.] <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>
8. HTML.net. *¿Qué es HTML?* [En línea] [Citado el: 4 de Abril de 2009.]
<http://es.html.net/tutorials/html/lesson2.asp>
9. **Álvarez, Miguel Angel .** Desarrollo Web. *¿Qué es XML?* [En línea] [Citado el: 4 de Abril de 2009.]
<http://www.desarrolloweb.com/articulos/449.php>
10. Web Taller. *¿Qué es AJAX?* [En línea]2008. [Citado el: 5 de Abril de 2009.]
<http://www.webtaller.com/maletin/articulos/que-es-ajax.php>
11. ExtJs. [En línea]2006. [Citado el: 11 de Abril de 2009.] <http://extjs.com/depoy/dev/docs>
12. Desarrollo Web. *¿Qué es PHP?* [En línea] [Citado el: 11 de Abril de 2009.]
<http://www.desarrolloweb.com/articulos/392.php>
13. MailxMail. *¿Qué es apache?* [Online] 10 de Enero 2006. [Cited: Abril 15, 2009.]
<http://www.mailxmail.com/curso-configuracion-apache/que-es-apache-que-alternativas-hay-mercado>
14. **Caro, Patricio Salinas .** Universidad de Chile. *Tutorial de UML.* [En línea] [Citado el: 20 de Abril de 2009.] <http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>
15. UtilZone.net. *ZendStudio.* [En línea]13 de Marzo 2008. [Citado el: 22 de Abril de 2009.]
<http://utilzone.net/index.php?s=5144f7c47c6584c98cd6857128950893&showtopic=1655&pid=39239&st=0&#entry39239>

16. **Pajón, Javier.**BaulSoft. *Mozilla Firefox*. [En línea] 2006.[Citado el: 24 de Abril de 2009.]
<http://www.javinet.com.ar/firefox.htm>
17. **Küng, Stefan, Onken, Lübbe y Large, Simon. 2009.** TortoiseSVN. Un cliente de Subversion para Windows. [En línea] 9 de mayo de 2009. [Citado el: 7 de junio de 2009.]
<http://hivelocity.dl.sourceforge.net/sourceforge/tortoisesvn/TortoiseSVN-1.6.2-es.pdf>

BIBLIOGRAFÍA

1. **Pressman, S. 2002.** *Ingeniería de Software: Un enfoque Práctico*. Quinta edición. 2002.
2. **Chidamber and Kemerer.** *Management Use of Metrics for Object-Oriented Software: An Exploratory Analysis*. s.l. : IEEE Trans. Software Engineering.
3. **Lorenz, M. and Kidd, J. 1994.** *Object Oriented Metrics*. s.l. : Englewood, 1994.
4. **Colectivo de Autores. 2008.** *ERP-PROD Modelo de Desarrollo orientado a componentes*. [Digital] La Habana : Universidad de las Ciencias Informáticas, 2008.
5. **Yzquierdo Herrera, Raykenler y Lazo Ochoa, René. 2007.** *El modelo de diseño del sistema HyperWeb. Módulos de Tratamiento Farmacológico y Configuración*. La Habana : Universidad de las Ciencias Informáticas, mayo de 2007.
6. **Dr. José Carlos del Toro, Ing. Henry Raúl González Brito.** *Documento Visión del proyecto ERP Cuba*. 2008
7. **Colectivo de Autores. 2008.** *ERP-PROD Modelo de Desarrollo orientado a componentes*. [Digital] La Habana : Universidad de las Ciencias Informáticas, 2008.
8. **Garlan, M. S. Y. D. 1996.** *Software Architecture: Prespectives on an emerging discipline*. s.l. : Upper Saddle River, Prentice Hall, 1996.
9. **González, H. 2006.** *ERP cubano, un paso estratégico para la consolidación del Software Libre en Cuba*. 2006. Vol. I.
10. **Hernández León, Rolando Alfredo y Coello González, Sayda. 2002.** *El paradigma cuantitativo de las investigación científica*. La Habana : Editorial Universitaria, 2002. ISBN: 959-16-0343-6.
11. **Kairos. 2008.** *Sistema de gestión de Inventario y Activo Fijo: Presentación de producto*. 2008.
12. Software de Gestión y Estrategia SAP. [En línea] 2009.[Citado el: 3 de Abril de 2009.] http://www.sap.com/spain/software/gestion_de_inventarios.epx
13. Openbravo ERP . [En línea] 2007.[Citado el: 4 de Abril de 2009.] <http://www.openbravo.com/es/>
14. OpenERP Sistema de Gestión Empresarial. [En línea] [Citado el: 4 de Abril de 2009.] <http://www.estudio42.com.ar/index.php/openerp>

15. Herramientas Web. *El modelo cliente – servidor*. [En línea] 2007.[Citado el: 5 de Abril de 2009.] <http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/cliente-servidor.html>
16. **Valdés, Damián Pérez** .Maestros del Web. *¿Qué es Javascript?* [En línea] 3 de Julio 2007 .[Citado el: 4 de Abril de 2009.] <http://www.maestrosdelweb.com/editorial/%C2%BFque-es-javascript/>
17. HTML.net. *¿Qué es HTML?* [En línea] [Citado el: 4 de Abril de 2009.] <http://es.html.net/tutorials/html/lesson2.asp>
18. **Álvarez, Miguel Angel** .Desarrollo Web. *¿Qué es XML?* [En línea] [Citado el: 4 de Abril de 2009.] <http://www.desarrolloweb.com/articulos/449.php>
19. Web Taller. *¿Qué es AJAX?* [En línea]2008. [Citado el: 5 de Abril de 2009.] <http://www.webtaller.com/maletin/articulos/que-es-ajax.php>
20. ExtJs. [En línea]2006. [Citado el: 11 de Abril de 2009.] <http://extjs.com/deploy/dev/docs>
21. Desarrollo Web. *¿Qué es PHP?* [En línea] [Citado el: 11 de Abril de 2009.] <http://www.desarrolloweb.com/articulos/392.php>
22. MailxMail. *¿Qué es apache?* [Online] 10 de Enero 2006. [Cited: Abril 15, 2009.] <http://www.mailxmail.com/curso-configuracion-apache/que-es-apache-que-alternativas-hay-mercado>
23. **Caro, Patricio Salinas** .Universidad de Chile. *Tutorial de UML*. [En línea] [Citado el: 20 de Abril de 2009.] <http://www.dcc.uchile.cl/~psalinas/uml/introduccion.html>
24. UtilZone.net. *ZendStudio*. [En línea]13 de Marzo 2008. [Citado el: 22 de Abril de 2009.] <http://utilzone.net/index.php?s=5144f7c47c6584c98cd6857128950893&showtopic=1655&pid=39239&st=0&#entry39239>
25. **Pajón, Javier**.BaulSoft. *Mozilla Firefox*. [En línea] 2006.[Citado el: 24 de Abril de 2009.] <http://www.javinet.com.ar/firefox.htm>
26. Sistema de Contabilidad Material para la Actividad Presupuestada de las FAR. 2000.
27. **Küng, Stefan, Onken, Lübbe y Large, Simon**. 2009. TortoiseSVN. Un cliente de Subversion para Windows. [En línea] 9 de mayo de 2009. [Citado el: 7 de junio de 2009.] <http://hivelocity.dl.sourceforge.net/sourceforge/tortoisesvn/TortoiseSVN-1.6.2-es.pdf>

ANEXOS

- Anexo1 Modelo de Ajuste de Inventario

ENTIDAD:		AJUSTE DE INVENTARIO					No.	
		CANT. POSICIONES:						
AREA		ALMACEN					FECHA	
							D	M
								A
___ MERMAS		___ MAL ESTADO			___ VENCIMIENTO			
___ ROTURAS EN ALMACEN		___ FALTANTES EN CONTEO			___ BAJAS UTENS. Y HERRAMIENTAS			
___ ROTURAS EN LA TRANSPORT		___ SOBRAINTES EN CONTEO			___ OTROS (DETALLAR)			
CODIGO	DESCRIPCION	UM	CANTIDAD	PRECIO UNITARIO	IMPORTE		SALDO EN EXISTENCIAS	
OBSERVACIONES:								
JEFE ALMACEN:		AUTORIZADO POR:			ANOTADO POR:		COTABILIZADO POR:	
NOMBRE Y APELLIDOS		NOMBRE Y APELLIDOS			T.ESTIBA		S.INV	
FIRMA		CARGO						
FIRMA		FIRMA						
		FECHA						
		D	M	A				

Figura 43 Modelo de Ajuste de Inventario

- Anexo3 Instrumento de medición de la métrica Tamaño operacional de clase (TOC)

	Categoría	Criterio
Cohesión	Baja	< =Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	> 2* Prom.

Complejidad implementación	Baja	< =Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	> 2* Prom.

Reutilización	Baja	> 2*Prom.
	Media	Entre Prom. y 2* Pom.
	Alta	<= Prom.

Tabla 6 Rango de valores de para la evaluación técnica de los atributos de calidad

No	Subsistema	Clase	Cantidad de Procedimientos	Cohesión	Complejidad	Reutilización
1	Ajuste	GestajusteinventarioModel	28	Alta	Alta	Baja
2	Ajuste	RegistrarproductosajusteModel	3	Baja	Baja	Alta
3	Ajuste	DocumentoValidator	7	Baja	Baja	Alta
4	Ajuste	MovimientoValidator	8	Baja	Baja	Alta
5	Baja	GestbajainventarioModel	21	Alta	Alta	Baja
6	Baja	RegistrarproductosModel	7	Baja	Baja	Alta
7	Baja	DocumentoValidator	7	Baja	Baja	Alta
8	Baja	MovimientoValidator	7	Baja	Baja	Alta
9	Baja	MovEquipoValidator	4	Baja	Baja	Alta

Tabla 7 Resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad

- Anexo4 Instrumento de medición de la métrica Relaciones entre clases (RC)

Acoplamiento	Categoría	Criterio
	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2
Complejidad Mant.	Categoría	Criterio
	Baja	<= Prom.
	Media	Entre Prom. y 2*Prom.
	Alta	> 2*Prom.

Reutilización	Categoría	Criterio
	Baja	>2* Prom.
	Media	Entre Prom. y 2*Prom.
	Alta	<= Prom.

Cantidad de Pruebas	Categoría	Criterio
	Baja	<= Prom.
	Media	Entre Prom. y 2*Prom.
	Alta	> 2*Prom.

Tabla 8 Rango de valores de para la evaluación técnica de los atributos de calidad relacionados con la métrica RC

No	Subsistema	Clase	Cantidad de Relaciones de Uso	Acoplamiento	Complejidad Mant.	Reutilización	Cantidad de Pruebas
1	Ajuste	GestajusteinventarioModel	0	Ninguno	Baja	Alta	Baja
2	Ajuste	RegistrarproductosajusteModel	0	Ninguno	Baja	Alta	Baja
3	Ajuste	DocumentoValidator	0	Ninguno	Baja	Alta	Baja
4	Ajuste	MovimientoValidator	0	Ninguno	Baja	Alta	Baja
5	Baja	GestbajainventarioModel	0	Ninguno	Baja	Alta	Baja
6	Baja	RegistrarproductosModel	0	Ninguno	Baja	Alta	Baja
7	Baja	DocumentoValidator	0	Ninguno	Baja	Alta	Baja
8	Baja	MovimientoValidator	0	Ninguno	Baja	Alta	Baja
9	Baja	MovEquipoValidator	0	Ninguno	Baja	Alta	Baja

Figura 45 Resultados de la evaluación de la métrica RC y su influencia en los atributos de calidad

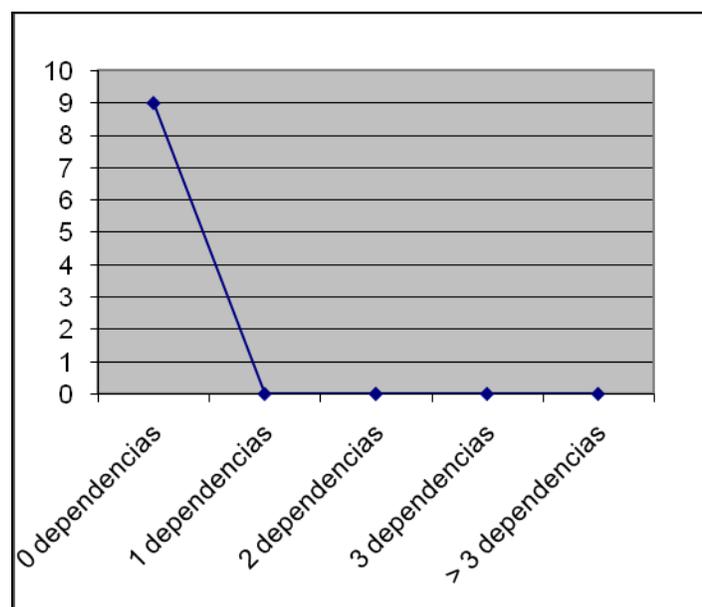


Figura 46 Gráfica de los resultados de la evaluación de la métrica RC agrupados por la tendencia de los valores

- Anexo5 Instrumento de medición de la métrica Profundidad de Herencia (PH)

No	Subsistema	Clase	Clase padre	Niveles del árbol de herencia
1	Ajuste	GestajusteinventarioModel	ZendExt_Model	2
2	Ajuste	RegistrarproductosajusteModel	ZendExt_Model	2
3	Ajuste	DocumentoValidator		0

4	Ajuste	MovimientoValidator		0
5	Baja	GestbajainventarioModel	ZendExt_Model	2
6	Baja	RegistrarproductosModel	ZendExt_Model	2
7	Baja	DocumentoValidator		0
8	Baja	MovimientoValidator		0
9	Baja	MovEquipoValidator		0

Tabla 9 Resultados de la evaluación de la métrica PH

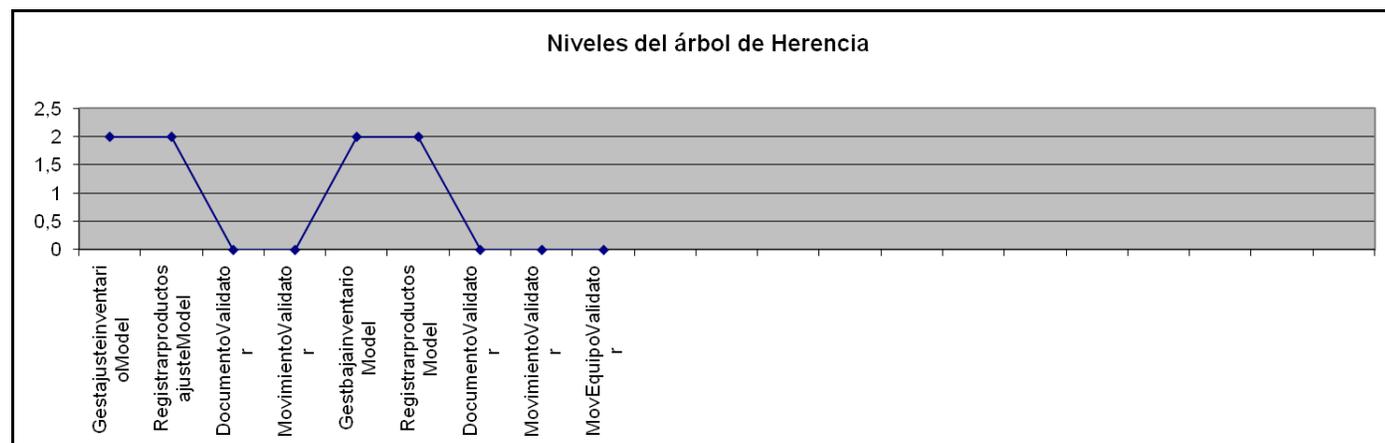


Figura 47 Gráfica de los resultados de la evaluación de la métrica PH

- Anexo6 Instrumento de medición de la métrica Número de Descendientes (ND)

	Categoría	Criterio
Reutilización	Baja	\leq Prom.
	Media	Entre Prom. y $2 \times$ Prom.
	Alta	$> 2 \times$ Prom.

	Categoría	Criterio
Abstracción	Indefinida	> 5
	Afectada	Entre 2 y 5
	Definida	≤ 2

	Categoría	Criterio
Cohesión	Baja	> 5
	Media	Entre 2 y 5
	Alta	≤ 2

	Categoría	Criterio
Cantidad de Pruebas	Baja	≤ 2
	Media	Entre 2 y 5
	Alta	> 5

Tabla 10 Rango de valores de para la evaluación técnica de los atributos de calidad relacionados con la métrica ND

No	Subsistema	Clase	Clase padre	Número de descendientes	Reutilización	Abstracción de la clase base	Cohesión de la JC	Cantidad de Pruebas
1	Ajuste	GestajusteinventarioModel	ZendExt_Model	0	Baja	Definida	Alta	Baja
2	Ajuste	RegistrarproductosajusteModel	ZendExt_Model	0	Baja	Definida	Alta	Baja
3	Ajuste	DocumentoValidator		0	Baja	Definida	Alta	Baja
4	Ajuste	MovimientoValidator		0	Baja	Definida	Alta	Baja
5	Baja	GestbajainventarioModel	ZendExt_Model	0	Baja	Definida	Alta	Baja
6	Baja	RegistrarproductosModel	ZendExt_Model	0	Baja	Definida	Alta	Baja
7	Baja	DocumentoValidator		0	Baja	Definida	Alta	Baja
8	Baja	MovimientoValidator		0	Baja	Definida	Alta	Baja
9	Baja	MovEquipoValidator		0	Baja	Definida	Alta	Baja

Figura 48 Resultados de la evaluación de la métrica ND y su influencia en los atributos de calidad

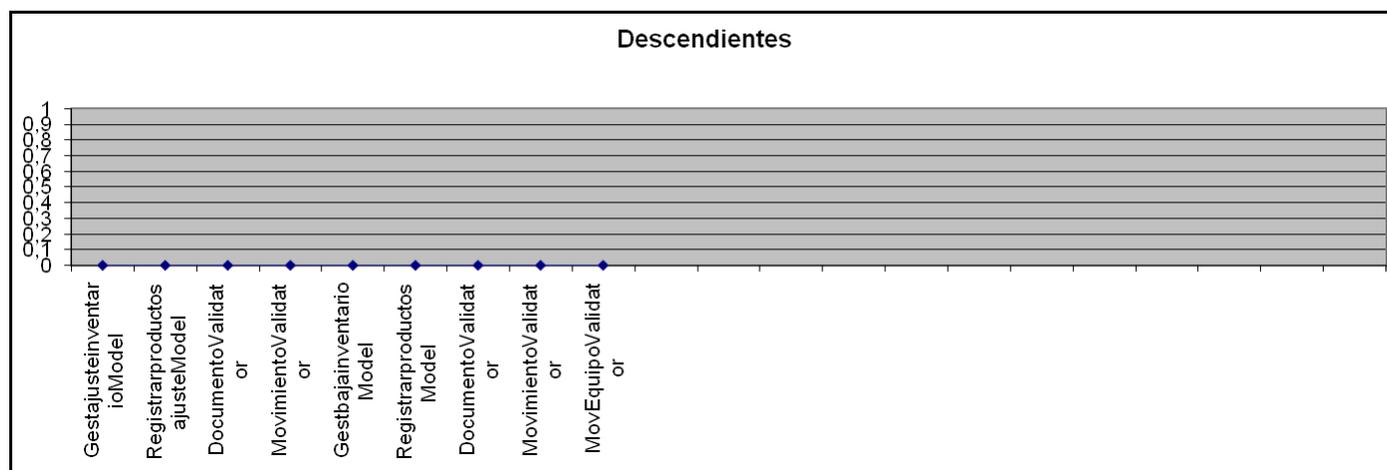


Figura 49 Representación de los resultados obtenidos al aplicar los instrumentos que evalúan la métrica ND

- Anexo7 Instrumento de medición de la métrica Número de Operaciones Redefinidas (NOR)

	Categoría	Criterio
Complejidad de Mantenimiento	Baja	0
	Media	Entre 0 y 2

	Categoría	Criterio
Cantidad de Pruebas	Baja	0
	Media	Entre 0 y 2

	Categoría	Criterio
Violación de la Abstracción representada por la superclase	Baja	0
	Media	Entre 0 y 2

Tabla 11 Rango de valores de para la evaluación técnica de los atributos de calidad relacionados con la métrica NOR

No	Subsistema	Clase	Clase padre	No de Operaciones redefinidas	Calidad del Diseño	Complejidad de Mant.	Cantidad de Pruebas	Violación de ARS
1	Ajuste	GestajusteinventarioModel	ZendExt_Mode	0	Buena	Baja	Baja	Baja
2	Ajuste	RegistrarproductosajusteModel	ZendExt_Mode	0	Buena	Baja	Baja	Baja
3	Ajuste	DocumentoValidator		0	Buena	Baja	Baja	Baja
4	Ajuste	MovimientoValidator		0	Buena	Baja	Baja	Baja
5	Baja	GestbajainventarioModel	ZendExt_Mode	0	Buena	Baja	Baja	Baja
6	Baja	RegistrarproductosModel	ZendExt_Mode	0	Buena	Baja	Baja	Baja
7	Baja	DocumentoValidator		0	Buena	Baja	Baja	Baja
8	Baja	MovimientoValidator		0	Buena	Baja	Baja	Baja
9	Baja	MovEquipoValidator		0	Buena	Baja	Baja	Baja

Figura 50 Resultados de la evaluación de la métrica NOR y su influencia en los atributos de calidad

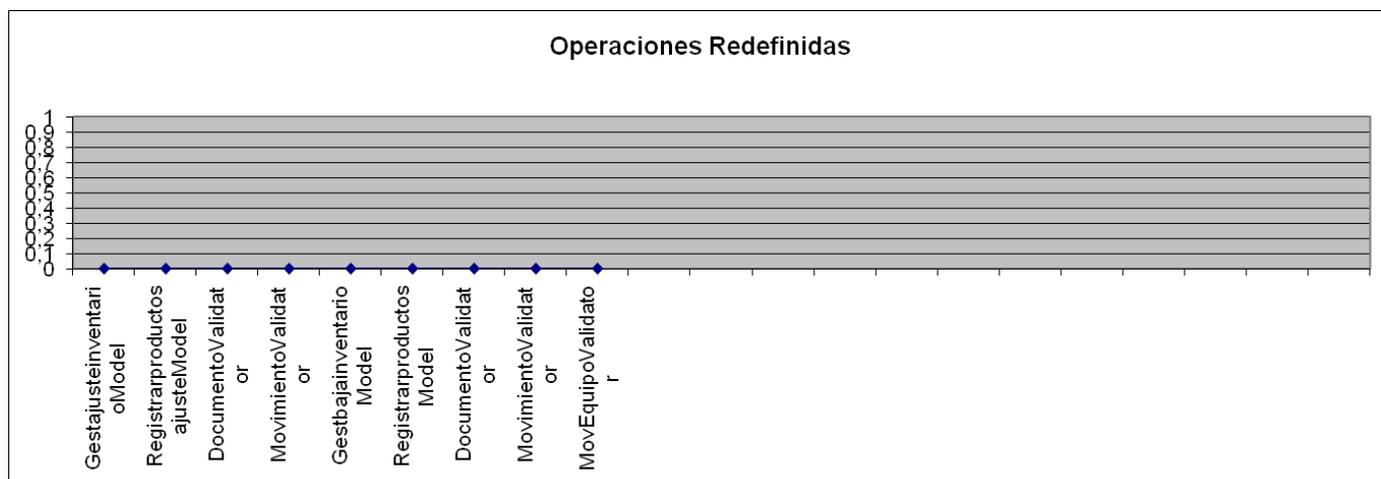


Figura 51 Representación de los resultados obtenidos al aplicar los instrumentos que evalúan la métrica NOR

- o Anexo8 Interfaz Principal del Sistema



Figura 52 Interfaz para el acceso a los módulo de Inventario

- Anexo9 Interfaz Principal para la gestión de los documentos de Ajuste

The screenshot shows a web application window titled 'Gestionar ajustes de inventario'. It features a search bar with fields for 'Número', 'Año', and 'Estado', and a 'Búsqueda avanzada' button. Below the search bar is a table with 10 rows of data. The table columns are: Número, Número del documento, Año, Estado, Motivo, Productos, Fecha, Nro inventario, Creado por, and Aprobado por. The data is as follows:

Número	Número del documento	Año	Estado	Motivo	Productos	Fecha	Nro inventario	Creado por	Aprobado por
1	47	2009	Elaboración	Ajuste por sobrante	1	22/05/2009		logistica	-
2	46	2009	Contabilizado	Faltante	1	28/05/2009		logistica	logistica
3	45	2009	Contabilizado	Rotura	1	28/05/2009		logistica	logistica
4	44	2009	Contabilizado	Importe (+)	1	28/05/2009		logistica	logistica
5	43	2009	Elaboración	Ajuste por sobrante	0	22/05/2009		logistica	-
6	42	2009	Elaboración	Faltante	1	22/05/2009		logistica	-
7	41	2009	Preparado	Ajuste por sobrante	1	22/05/2009	15	logistica	-
8	40	2009	Preparado	Ajuste por sobrante	1	22/05/2009	13	logistica	-
9	39	2009	Preparado	Ajuste por sobrante	2	22/05/2009	5	logistica	-
10	38	2009	Preparado	Faltante	4	22/05/2009	5	logistica	-

At the bottom of the window, there is a pagination control showing 'Página 1 de 5' and a status bar indicating 'Resultados del 1 - 10 de 42'.

Figura 53 Interfaz principal para la gestión de los documentos de Ajuste

- Anexo10 Interfaz adicional documento de Ajuste

The screenshot shows a form titled 'Adicionar ajuste de inventario'. The form contains the following fields and elements:

- Datos del documento:**
 - Almacén: Déposito A
 - Entidad: UCI
 - Número: 48
 - Fecha: 01/09/2009
 - Estado: Elaboración
- Motivo de ajuste:** A dropdown menu with the following options: Ajuste por sobrante, Faltante, Merma, Deterioro, Rotura, Otros (+), Otros (-), Importe (+), and Importe (-).
- Observaciones:** A text area for entering notes.
- Buttons:** Cancelar, Aplicar, and Aceptar.

Figura 54 Interfaz adicional documento de Ajuste

- Anexo11 Interfaz para la gestión de los productos

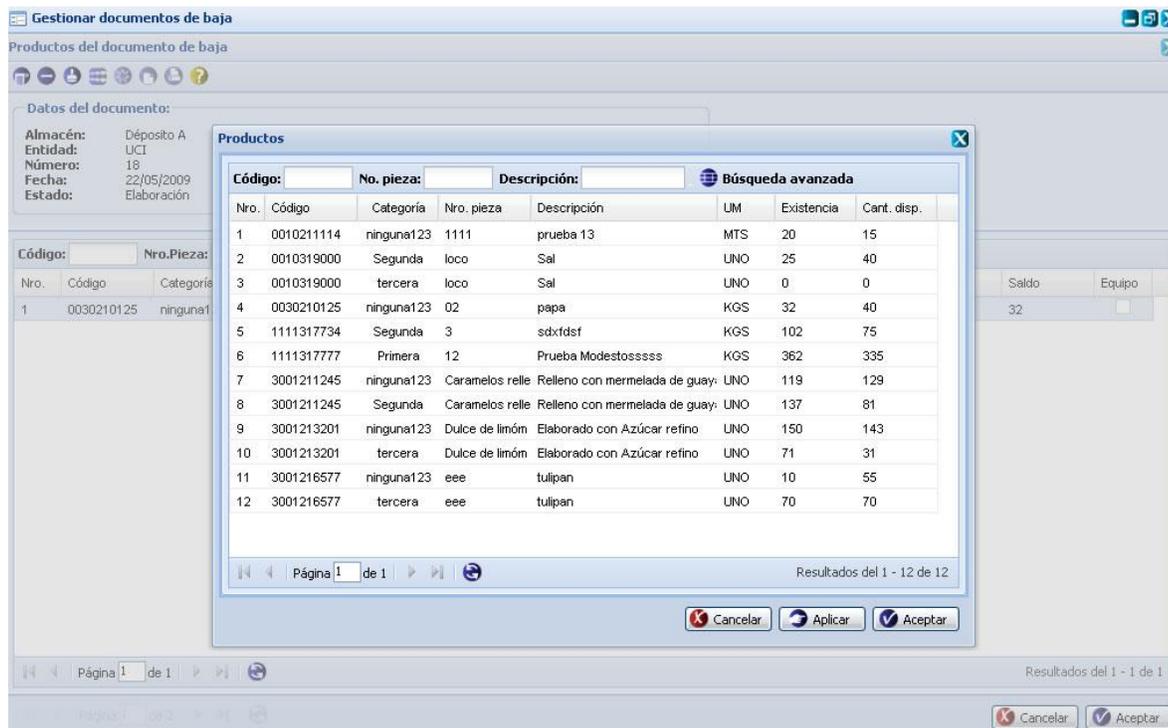


Figura 55 Interfaz para la gestión de los productos

- Anexo12 Interfaz para la especificación de los números de series

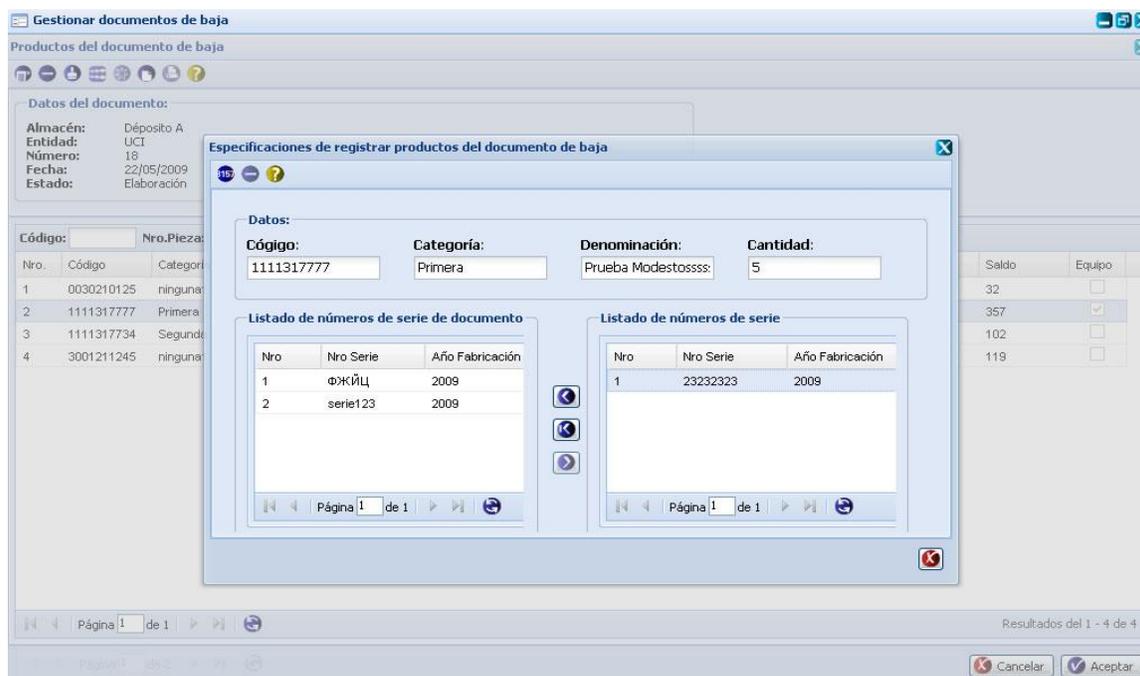


Figura 56 Interfaz para la especificación de los números de series

GLOSARIO DE TÉRMINOS

Ajuste de inventario: Consiste en anotar en los registros de control de existencias (tarjetas, libros o registros) los resultados de los conteos físicos realizados, cuando se detectan diferencias entre las existencias físicas, según los saldos de los mencionados registros. Las diferencias pueden ser originadas por faltantes, mermas, vencimiento, avería, sobrante, hecho extraordinario.

Avería: Daño material que consiste en la destrucción, rotura o deterioro de un medio material como consecuencia de su manipulación o de accidentes durante la carga, la transportación, la descarga, el almacenaje o el despacho.

Baja técnica de los medios materiales: Consiste en anotar en los registros de control de las existencias el resultado de las bajas aprobadas para el armamento, la técnica de todo tipo, las municiones, los equipos y demás medios materiales cuando por la pérdida de condiciones físicas, funcionales o ambas, quedan inhabilitados para el uso al que estaban destinados.

Faltante: Diferencia en defecto que se produce por sustracción, desorden (colocación de un medio material fuera de su lugar), mal conteo en la recepción o en el despacho y errores de anotación o aritméticos en los registros.

Hecho extraordinario: Daño material de consideración que se produce por acción delictiva o accidente, así como por un desastre provocado por fuerzas naturales o por acciones u omisiones negligentes o intencionales.

Merma: Pérdida del peso o volumen, así como de las cualidades de un producto por causas naturales. Incluye la evaporación.

Sobrante: Diferencia en exceso que se produce por mal conteo en la recepción o en el despacho y por errores de anotación o aritméticos en los registros.

Vencimiento: Pérdida de las cualidades de un producto al arribar al término fijado para su utilización.