

**Universidad de las Ciencias Informáticas**  
**Centro de Soluciones de Gestión**



**Título:** Implementación del Módulo Recepción del  
Subsistema Inventario del Sistema Integral de Gestión  
CedruX.

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

**Autores:** Reinier Lugo Díaz.  
Yuriel Martínez Ordaz.

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

**Ciudad de La Habana, Junio de 2009.**  
**“Año 50 del triunfo de la Revolución.”**

**DATOS DEL CONTACTO:**

Tutora: Lic. Arismayda Dorado Risco

Clasificación: Profesional.

Clasificación del área de desarrollo: Facultad 3.

Síntesis de la Tutora: Licenciada en ciencias de la Computación, 4 años de graduada.

Actualmente es Arquitecta de Sistema de los Subsistemas Inventario y Facturación del proyecto ERP-Cuba.

Categoría Científica: Licenciada

Categoría Docente: Instructor

Tutora: Ing. Meylin Martínez Chong

Clasificación: Profesional.

Clasificación del área de desarrollo: UCID

Síntesis de la Tutora: Ingeniera Informática, tres años de graduada.

Actualmente es Arquitecta de Sistema de los Subsistemas Inventario y Facturación del proyecto ERP-Cuba.

Categoría Científica: Ingeniero

AGRADECIMIENTOS:

*Agradecemos a todas aquellas  
personas y entidades que nos ayudaron  
para la realización de esta tesis y  
que sin el ánimo , comprensión y amor  
de nuestros familiares y amigos, no podría  
haber existido*

**DEDICATORIA:**

*A mi mamá y a mi abuelita, por su gran apoyo y confianza en todo momento, por ser los ángeles que guían mis pasos, siendo la inspiración de mi vida, y por haberme dado los mejores consejos del mundo ayudándome a ser una mejor persona cada día.*

*A mi hermana, a la cual quiero con todas las fuerzas de mi corazón, por ser una personita tan buena y tan especial conmigo, que sirva esta tesis de diploma como ejemplo de sacrificio profesional, y demostración de que ella lo puede hacer mucho mejor que yo.*

*A mi compañero y hermano de tesis Yuriel, que tantas horas de insomnio le ha dedicado a la confección de este trabajo tan importante para la vida de los dos. A ti e más sincero agradecimiento hermano.*

*A mi papito Julio, por ser un ejemplo a seguir, por todos sus consejos, por haberme enseñado que la vida es sacrificio y esfuerzo, y sobre todo por ser el gran padre que es.*

*A mi padrastro Eddy que ha llegado a ocupar un gran lugar dentro mi corazón que día a día lo quiero más y más.*

*A toda mi familia que ha estado al tanto de cada momento de mi trayectoria y resultados.*

*A mis amigos: Ray, Jose, Fidel, Toste, Ivian, Lara, Liobel, Edel, Daiyan por ser tan incondicionales en todo momento y haberme aceptado tal y como soy, por todos sus consejos y sus muestras de lo que es verdaderamente una gran amistad.*

*Al equipo de desarrollo que tantas horas de esfuerzo dedicó a la realización de esta tarea de tan alta importancia para nuestro país.*

*A nuestras tutoras que tanto nos han ayudado en la realización de este Trabajo de Diploma, a ellas va también nuestro gran agradecimiento.*

*A nuestra gran revolución y a nuestro invicto y glorioso comandante Fidel, que hicieron posible que existiera esta gran Universidad.*

**Reinier Lugo Díaz.**

*A mis padres que tanto me han malcriado, por sus consejos, por su apoyo incondicional en todo y por estar siempre presentes, por malos que fueran los tiempos.*

*A mi alocada familia que siempre me ha incitado a estudiar, me ha apoyado y ha contribuido, en su gran mayoría, ha que yo esté hoy aquí.*

*A nuestro eterno comandante que con su visión futura creó este centro de estudios, dando la posibilidad de estudiar en él a jóvenes como yo, que en otros tiempos, no habrían tenido la oportunidad siquiera de imaginarlo.*

*A todos los pesaos de mi grupo, los que me estiman, por tener siempre la mano extendida cuando los necesité, y a los que no, por tenerme siempre presente, y haberme incitado día a día a terminar mis estudios en esta escuela.*

*A mis tutoras y compañeros del proyecto por su ayuda en la realización de este Trabajo de Diploma.*

*A alguien muy especial que no por ser la última significa que es menos importante, sino todo lo contrario, a ella: por su apoyo, sus consejos, por comprenderme y apoyarme en los peores momentos de mi vida de estudiante, por enseñarme que la familia es lo más importante que tiene una persona, por mostrarme tantas cosas de la vida que no me alcanzan las palabras para decirlas, así como tampoco me alcanza el aliento para expresar todo lo que ha significado para mi familia: mi abuela **Meya** a quien la vida no le dio la oportunidad de verme graduado, dedico este trabajo, con todo mi cariño.*

**Yuriel Martínez Ordaz**

**PENSAMIENTO:**



*En Cuba nadie ha hecho tanto en tan poco tiempo.  
Fidel Castro.*

**RESUMEN:**

En la actualidad el mundo está guiado por el uso y desarrollo de las (TIC)<sup>1</sup>, estas a su vez están regidas por la informática como ciencia fundamental y por el uso de programas de cómputo o software, que facilitan el trabajo en las diferentes actividades de la vida diaria. Las empresas vienen dando mayor importancia a las Tecnologías de Información y su alineación con las estrategias de negocio para mejorar sus procesos claves. Prueba de ello, es el incremento tan sustancial de adquisiciones de paquetes de software empresariales tales como los ERP<sup>2</sup>. Un sistema de ERP involucra una gran diversidad de áreas y procesos empresariales entre los cuales se encuentra la gestión de los inventarios, encargada de gestionar las entradas y salidas de materiales en el almacén y su ubicación en este, los posteos y la obtención de la información necesaria para la entidad.

El presente trabajo tiene como propósito informatizar el proceso de Recepción de productos o Entrada de materiales a un almacén (sean productos adquiridos, productos fabricados o terminados en la propia empresa, o productos o mercancías devueltas por las áreas que las extrajeron con anterioridad). Esperándose que el sistema sea usado en las instituciones y empresas del país, posibilitando entre otras ventajas que la información que se maneje sea real, clara, verdadera y completa de forma tal que pueda ser utilizada en la toma de decisiones.

Palabras claves: Control de inventarios, Producto, Recepción.

---

<sup>1</sup> **TIC:** Tecnologías de la Información y las Comunicaciones.

<sup>2</sup> **ERP:** Planificación de Recursos Empresariales (del inglés Enterprise Resource Planning).

**ÍNDICE.**

**INTRODUCCIÓN:**..... 1

Capítulo 1 ..... 5

Fundamentación Teórica ..... 5

    1.1. Introducción ..... 5

    1.2. La gestión de inventarios en la planeación de recursos empresariales ..... 5

    1.3. Sistemas informáticos existentes vinculados al campo de acción ..... 6

        1.3.1. Valoración crítica de los sistemas analizados ..... 11

    1.4. Proceso de Desarrollo de software ..... 12

    1.5. Tecnologías, lenguajes y Herramientas ..... 14

        1.5.1. Herramientas ..... 15

        1.5.2. Lenguajes ..... 21

        1.5.3. Tecnologías ..... 25

            1.5.3.1. Marcos de Trabajo ..... 25

            1.5.3.2. Arquitectura ..... 27

            1.5.3.3. Servidor Web ..... 30

            1.5.3.4. Navegador ..... 31

    1.6. Conclusiones parciales ..... 34

Capítulo 2 ..... 35

Características del Sistema ..... 35

    2.1. Introducción ..... 35

    2.2. Flujo del proceso ..... 35

        2.2.1. Valoración crítica de los artefactos propuestos por los analistas ..... 35

        2.2.2. Objetos de automatización ..... 38

        2.2.3. Información que se maneja ..... 38

        2.2.4. Propuesta del sistema ..... 39

    2.3. Descripción de la implementación ..... 41

        2.3.1. Diagrama de despliegue ..... 41

        2.3.2. Diagrama de componentes ..... 42



2.3.3.	Integración entre componentes .....	45
2.3.4.	Normas y Estándares de codificación. ....	46
2.4.	Análisis de reutilización de componentes, código o módulos .....	49
2.5.	Descripción de las principales clases a utilizar.....	51
2.6.	Descripción de los algoritmos no triviales a implementar .....	57
2.6.1.	Análisis de complejidad de los mismos .....	58
2.7.	Conclusiones parciales .....	62
Capítulo 3	.....	63
Validación de la solución propuesta	.....	63
3.1.	Introducción .....	63
3.2.	Pruebas de Software .....	63
3.2.1.	Objetivos de las Pruebas .....	64
3.2.2.	Pruebas de unidad .....	64
3.2.3.	Técnicas de Diseño de Pruebas.....	65
3.2.4.	Prueba de Caja Blanca o Estructurales.....	65
3.2.5.	Aplicación de pruebas de caja blanca .....	67
3.3.	Evaluación de la solución obtenida mediante métricas .....	73
3.3.1.	Métricas aplicadas .....	78
3.3.2.	Resultados de los instrumentos de evaluación de las métricas .....	80
3.3.2.1.	Resultados del instrumento de evaluación de la métrica Tamaño operacional de clase (TOC) .....	80
3.3.2.2.	Resultados del instrumento de evaluación de la métrica Relaciones entre Clases (RC) ...	83
3.3.2.3.	Resultados del instrumento de evaluación de la métrica Profundidad de Herencia (PH)...	86
3.3.2.4.	Resultados del instrumento de evaluación de la métrica Número de Operaciones Redefinidas (NOR).....	89
3.3.3.	Matriz de cubrimiento de los parámetros de calidad evaluados con las métricas propuestas.....	91
3.4.	Conclusiones parciales .....	93
Conclusiones generales	.....	94

Recomendaciones .....	95
Referencias bibliográficas .....	96
Glosario de términos .....	99
Anexos.....	100

**ÍNDICE DE FIGURAS.**

Figura 1 Representación de los roles definidos por el Modelo de desarrollo. .... 13

Figura 2 Representación del Patrón MVC. .... 28

Figura 3 Arquitectura Cliente/Servidor. .... 30

Figura 4 Diagrama de despliegue. .... 42

Figura 5 Diagrama de Integración entre Componentes. .... 43

Figura 6 Diagrama de Componentes. .... 44

Figura 7 Diagrama de colaboración entre clases de la arquitectura. .... 46

Figura 8 Comentariado de la declaración de una clase. .... 49

Figura 9 Algoritmo no trivial a implementar contabilizar (\$idestructuracomun, \$alias, \$iddocumento). .... 59

Figura 10 Notación de grafos de flujo para las instrucciones: Secuenciales, If, While. .... 60

Figura 11 Notación de grafos de flujo para la instrucción Case. .... 60

Figura 12 Grafo de flujo asociado al algoritmo contabilizar (). .... 61

Figura 13 Representación de pruebas de Caja Blanca y Caja Negra. .... 65

Figura 14 Representación del algoritmo modifDual (). .... 67

Figura 15 Grafo de flujo asociado al algoritmo modifDual (). .... 68

Figura 16 Representación de los resultados obtenidos en el instrumento agrupados en los intervalos definidos. .... 80

Figura 17 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos. .... 81

Figura 18 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad. .... 81

Figura 19 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Complejidad de Implementación. .... 82

Figura 20 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Reutilización. .... 82

Figura 21 Representación de los resultados de la evaluación de la métrica RC agrupados por la tendencia de los valores. .... 83

Figura 22 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos. .... 84

Figura 23 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento. ....84

Figura 24 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de Mantenimiento. ....85

Figura 25 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de Pruebas.....85

Figura 26 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización.....86

Figura 27 Representación en % de los resultados obtenidos en el instrumento agrupados por nivel. ....87

Figura 28 Representación de la incidencia de los resultados de la evaluación de la métrica PH en el atributo Complejidad de Mantenimiento. ....87

Figura 29 Representación de la incidencia de los resultados de la evaluación de la métrica PH en el atributo Complejidad de Implementación. ....88

Figura 30 Representación de la incidencia de los resultados de la evaluación de la métrica PH en el atributo Reutilización.....88

Figura 31 Representación en % de los resultados obtenidos en el instrumento agrupados en los valores existentes.....89

Figura 32 Representación de la incidencia de los resultados de la evaluación de la métrica NOR en el atributo Complejidad del Mantenimiento. ....90

Figura 33 Representación de la incidencia de los resultados de la evaluación de la métrica NOR en el atributo Cantidad de Pruebas.....90

Figura 34 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.....91

Figura 35 Factura Comercial..... 100

Figura 36 Factura – Recepción de operaciones comerciales con productores individuales agropecuarios. .... 101

Figura 37 Informe de recepción. .... 102

Figura 38 Informe de reclamación..... 103

Figura 39 Vale de entrega o devolución..... 104

Figura 40 Entrega de productos terminados al almacén. .... 105

Figura 41 Submayor de inventario. ....	106
Figura 42 Modelo de tarjeta de estiba. ....	107
Figura 43 Conduces.....	108
Figura 44 Gráfica de los resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización).....	115
Figura 45 Gráfica de los resultados de la evaluación de la métrica RC agrupados por la tendencia de los valores.....	117
Figura 46 Gráfica de los resultados de la evaluación de la métrica PH. ....	118
Figura 47 Representación de los resultados obtenidos al aplicar los instrumentos que evalúan la métrica NOR.....	120

**ÍNDICE DE TABLAS.**

Tabla 1 prefijos a utilizar en la creación de variables. ....48

Tabla 2 Clase utilizada en la solución (GestinfrecepcionController) .....52

Tabla 3 Clase utilizada en la solución (GestinfrecepcionModel) .....53

Tabla 4 Clase utilizada en la solución (GestindiferenciaController) .....55

Tabla 5 Clase utilizada en la solución (GestindiferenciaModel) .....56

Tabla 6 Clase utilizada en la solución (GestproductoModel) .....57

Tabla 7 Requisitos asociados al módulo de recepción. ....77

Tabla 8 Tamaño operacional de clase (TOC).....78

Tabla 9 Relaciones entre clases (RC).....79

Tabla 10 Profundidad de Herencia (PH). ....79

Tabla 11 Número de Operaciones Redefinidas para una clase hija (NOR).....80

Tabla 12 Clasificación de los parámetros de calidad según impacto en el diseño propuesto .....92

Tabla 13 Rangos para evaluar el impacto de los parámetros de calidad en la solución. ....92

Tabla 14 Matriz de cubrimiento para los parámetros de calidad evaluados con las métricas aplicadas a la solución. ....93

Tabla 15 Documento de Recepción Model..... 110

Tabla 16 Documento de Diferencia Model. .... 111

Tabla 17 Movimiento de productos del Informe de Recepción. .... 112

Tabla 18 Movimiento de productos del Informe de Diferencia. .... 113

Tabla 19 Rango de valores de para la evaluación técnica de los atributos de calidad (Responsabilidad, 114

Tabla 20 Resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización)..... 115

Tabla 21 Rango de valores de para la evaluación técnica de los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas) relacionados con la métrica RC. . 116

Tabla 22 Resultados de la evaluación de la métrica RC y su influencia en los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas) ..... 117

Tabla 23 Resultados de la evaluación de la métrica PH..... 118

Tabla 24 Rango de valores de para la evaluación técnica de los atributos de calidad (Abstracción del diseño, Cantidad de pruebas, Complejidad de Mantenimiento) relacionados con la métrica NOR. .... 119

Tabla 25 Resultados de la evaluación de la métrica NOR y su influencia en los atributos de calidad (Abstracción del diseño, Cantidad de pruebas, Complejidad de Mantenimiento). ..... 120

## INTRODUCCIÓN:

Hoy en día se hace indispensable para cualquier empresa del mundo tener informatizada cada una de las áreas que la componen, requiriendo para ello de herramientas que les proporcionen control y centralización de su información, todo esto con el fin de tomar las mejores y más oportunas decisiones para sus procesos y estrategias de negocio. Los ERP son una solución robusta para aquellas empresas que buscan una solución universal a la centralización de su información.

La implementación de un sistema de ERP por lo general es un proceso largo y complejo, ya que implica rediseñar los esquemas de trabajo. Su implementación es de alto riesgo, ya que envuelve complejidad, tamaño, altos costos, un equipo considerable de desarrollo, además de inversión de tiempo.

Nuestro país sin estar ajeno a esta tendencia mundial, guiado por el uso y desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC), desde hace ya varios años invierte grandes recursos y esfuerzos en el proceso de informatización de la sociedad, constituyendo la actividad económica un área priorizada en esta contienda. A nivel nacional existen actualmente gran cantidad de sistemas ERP nacionales y extranjeros en los cuales se ha invertido grandes cifras de dinero anualmente, en término de pagos de licencias, mantenimiento, reparación y capacitación de personal. La inmensa mayoría de estos productos no se corresponden totalmente con las actuales necesidades de la economía nacional, en muchos casos debido a que su construcción fue enfocada a un sector específico de la economía, se desarrollaron en el seno de economías totalmente diferentes a la cubana o con tecnologías obsoletas. A todo lo anteriormente expuesto se adiciona el surgimiento en la economía nacional de otra moneda contable que posee los mismos privilegios de la antigua moneda, dando lugar a lo que se conoce como la Dualidad Monetaria.

Esta situación conllevó a que la Dirección de Política Contable perteneciente al Ministerio de Finanzas y Precios en conjunto con los demás Ministerios y Organismos del Estado Cubano, tomaran como acuerdo la realización de un Sistema Integral de Gestión, adecuado a las exigencias y características de la economía nacional, permitiendo la recopilación de toda la información generada durante la gestión de los procesos de forma estándar, ayudando a los directivos en la toma de decisiones.

Los sistemas ERP involucran e integran varias áreas y procesos empresariales entre los cuales se encuentra la gestión del área de la Logística, destinada a tramitar los flujos de materiales e información,



siendo su objetivo fundamental la satisfacción de las necesidades en bienes y servicios de un cliente y/o mercado, en calidad, cantidad, lugar y momento; maximizando la satisfacción del cliente y la flexibilidad de respuesta minimizando el tiempo de la misma y los costes. La Logística agrupa las actividades que ordenan los flujos de materiales, coordinando recursos y demandas para asegurar un nivel determinado de servicio al menor coste posible.

Dentro de las actividades de la Logística, la Gestión de Inventario es sin duda alguna uno de los aspectos más desafiantes por su complejidad y las implicaciones de riesgo financiero que involucra. Uno de los procesos involucrados en el control de inventarios es la Recepción de productos destinados a almacenamiento, sean productos que se compran, productos fabricados por la propia empresa o devoluciones de clientes.

Analizando la situación anteriormente expuesta, se impone el siguiente **problema científico**: ¿Cómo obtener un producto funcional a partir de los requerimientos identificados para los procesos de Recepción del Subsistema Inventario para las entidades empresariales y unidades presupuestadas del país? Para enfrentar esta problemática fueron **objeto de estudio** de la investigación los procesos para el Control de Inventario, enmarcando el **campo de acción** en los procesos para la recepción en las entidades empresariales y unidades presupuestadas del país.

Basado en la idea anteriormente expuesta se puede definir el **objetivo general** del presente trabajo como: Implementar los procesos de recepción de productos en almacenamiento para las entidades empresariales y unidades presupuestadas del país.

Se plantean además como **objetivos específicos**:

- Analizar los procesos de Recepción de productos de un almacén y sistemas que existen actualmente para su control, así como las herramientas que se utilizarán para el desarrollo de la solución.
- Implementar un módulo para la recepción de los productos de un almacén.
- Validar la solución propuesta.

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

- Análisis de los procesos para la Recepción de productos de un almacén y su relación con el resto de los procesos asociados al inventario.

- Análisis de los sistemas existentes identificando sus características y deficiencias fundamentales.
- Análisis de las tecnologías, lenguajes y herramientas propuestas para el desarrollo de la aplicación.
- Realización de un análisis de los artefactos entregados por los analistas para el proceso Recepción.
- Implementación de las interfaces a partir del prototipo entregado por los analistas.
- Implementación de los requerimientos especificados para los procesos de Recepción.
- Realización de las pruebas de unidad e integración.
- Aplicación de métricas para validar la calidad de la solución.

Para desarrollar este trabajo se ha planteado la siguiente **idea a defender**: Si se realiza la implementación de los procesos de recepción 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.

Los siguientes **métodos teóricos** sustentan el trabajo de investigación:

- Histórico-Lógico: Su empleo permitió el desarrollo evolutivo y coherente en el estudio de la metodología orientada a objetos, patrones de diseño, herramientas CASE<sup>3</sup> y sistemas ERP para el desarrollo de los artefactos que proponen los flujos estudiados.
- Analítico-Sintético: Permitted integrar y descomponer el conocimiento, resultó importante a la hora de fundamentar los elementos más significativos de la investigación determinando los aspectos esenciales y el arribo a conclusiones prácticas y teóricas.
- Inductivo-deductivo: Permitted pasar de la particularidad del proceso a lo general y viceversa beneficiando objetivamente el vínculo que se establece en la realidad entre lo singular que ocurre en cada entidad nacional y lo general que esta estandarizado para ellas, ya que ambas se complementan mutuamente en el proceso de desarrollo.
- Modelación: Pues se crean abstracciones que explican la realidad, por ejemplo, todos los modelos y diagramas presentados.

Luego de realizada la implementación de los procesos de recepción de productos para almacenamiento, proporcionará a las entidades los siguientes aportes prácticos:

---

<sup>3</sup>**CASE**: *Ingeniería de Software Asistida por Ordenador (del inglés Computer Aided Software Engineering).*

- La elaboración del Informe de Recepción que ampara la entrada de los productos.
- La generación de los Informes de Diferencias a partir de Informes de Recepción.
- La gestión de la Dualidad Monetaria.
- La ubicación de los productos recibidos durante el proceso de Recepción.
- El registro de los números de serie de los productos que tengan definida esta característica.
- La gestión por lotes del producto.

La estructura del documento está definida de la siguiente manera:

- **En el Capítulo 1. Fundamentación teórica:** Se exponen los conceptos fundamentales relacionados con el tema de investigación y se describen los principales aspectos de los lenguajes, herramientas y metodologías a utilizar para el desarrollo de la aplicación.
- **En el Capítulo 2. Descripción y análisis de la solución propuesta:** Se realiza una valoración crítica del diseño propuesto por los analistas, se analizan las implementaciones, componentes o módulos ya existentes y que puedan ser rehusados, se examinan las estrategias de integración. Se describen los algoritmos no triviales a implementar analizándose la complejidad de los mismos. Se efectúa la selección de las estructuras de datos apropiadas para la implementación de estos algoritmos y se realiza la descripción de las clases u operaciones que se utilicen para representar computacionalmente dichas estructuras.
- **En el Capítulo 3. Validación de la solución propuesta:** Se desarrolla el estudio y diseño de las pruebas de unidad para validar la solución propuesta. Se realiza una descripción de estas teniendo en cuenta objetivo, alcance, tipo de prueba y detalles de la misma. Se efectúa la descripción de los valores utilizados para las pruebas, así como una evaluación de la ejecución de estas y de los resultados obtenidos. Además se emite una propuesta de iteración y de posibles mejoras a realizar en la implementación.

# Capítulo 1

## Fundamentación Teórica

### 1.1. Introducción

Las tecnologías en la sociedad de hoy se encuentran en constante avance, desarrollo y evolución. Nuestro país apuesta por tener una independencia tecnológica del software y aplicaciones propietarias, por los grandes beneficios que esto traería a la economía nacional, destacándose el ahorro de miles de dólares en compra de licencias. En el presente capítulo se describen las tendencias, herramientas y tecnologías usadas para el desarrollo de la aplicación propuestas por el equipo de arquitectura del proyecto. También se realiza un análisis de sistemas que permiten el control de inventarios para evaluar sus ventajas y deficiencias que propicien un mejor entendimiento para una propuesta de solución.

### 1.2. La gestión de inventarios en la planeación de recursos empresariales

Los sistemas ERP son herramientas estratégicas de trabajo, se registra su surgimiento aproximadamente hace cuatro décadas atrás. La evolución de estos sistemas ha estado muy relacionada con el desarrollo que actualmente manifiestan las tecnologías de la información y las técnicas de gestión. Provocando que las empresas comenzaran a verlos como aplicaciones capaces de hacerlas mejorar. Estos son sistemas estructurados que buscan satisfacer las demandas de soluciones de gestión empresarial, basados en el concepto de una solución completa que permita a las empresas unificar las diferentes áreas de productividad.

Entre los procesos que gestionan estos sistemas informáticos se encuentran los de Control de Inventario. Este 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. (1) En particular los procesos de Recepción tienen como objetivo dar entrada en el almacén a los productos soportados por documentos que pueden ser: Transferencia, Vale de Devolución, Entrega de Producción Terminada, Conduce o Factura, actualizándose la tarjeta de estiba y el submayor de inventario.

### 1.3. Sistemas informáticos existentes vinculados al campo de acción

A nivel mundial existen proveedores que marcan la pauta en el mercado ERP y que incluyen en el paquete de software el Control de Inventarios y la recepción de productos. Algunas de estas grandes empresas proveedoras de estos productos de software a nivel mundial son:

- **SAP<sup>4</sup>**: Fue fundada en 1972 en Alemania por cinco ingenieros de la IBM<sup>5</sup>, siendo hoy la mayor empresa de su rama. Su sistema fue optimizado para gestionar los procesos de producción y gestión, logística y recursos humanos. Hoy día, pasados más de 30 años, cuenta ya con más de 12 millones de usuarios, 64.500 instalaciones, 1.500 socios y 23 soluciones informáticas. Es considerada la mayor empresa proveedora de ERP a nivel mundial, contribuyendo para eso, haber sido una de las pioneras. En Cuba existen compañías como la Empresa de Telecomunicaciones de Cuba, Sociedad Anónima (ETECSA) que usan sus soluciones implementadas (2)

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.

Para el Control de Inventario, puede realizar un seguimiento de la cantidad y el valor de todos sus materiales, realizar un inventario físico y optimizar los recursos del almacén. Los empleados pueden planificar, introducir y documentar movimientos de almacén interno gestionando las

---

<sup>4</sup> **SAP**: *Sistemas, aplicaciones y productos para el proceso de datos*

<sup>5</sup> **IBM**: *Del inglés, International Business Machines, empresa multinacional que fabrica y comercializa herramientas, programas y servicios relacionados con la informática.*

entradas y las salidas de mercancías, el almacenamiento, la recogida y el embalaje, los traslados físicos, entre otras.

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. (3)

- **Exact Globe:** Establecida en Holanda 1984, Exact Software es uno de los proveedores líderes de soluciones de gestión empresarial en todo el mundo. Entre Su oferta de soluciones integradas está el tradicional sistema de gestión administrativo-financiero (ERP) (4), que cubre las funcionalidades en las áreas de contabilidad, distribución, logística, servicios, RRHH<sup>6</sup>, Nóminas, Proyectos, Fabricación, entre otras.

Es un software modular diseñado para cualquier tipo de empresa que ha sido traducido a múltiples idiomas. Además, soporta regímenes legales y comerciales en más de 20 legislaciones diferentes, y se encuentra traducida a más de 25 idiomas por lo que resulta altamente recomendable para empresas internacionales. (4)

Exact Globe permite automatizar los procesos de negocio en el ámbito del Control de Inventario y gestión de productos, además permite el trabajo con varios almacenes en distintas localidades. Puede registrar los movimientos de productos de forma fácil y rápida, trabajar con el precio real de compra al hacer la recepción de productos y mercancías, además de permitir el tratamiento que se le realiza a los productos en cuanto a su registro por lote y números de serie.

- **OpenBravo:** Es un sistema de gestión empresarial en software libre, completamente funcional, integrado y basado en web. Dispone de soporte para bases de datos PostgreSQL y Oracle.

---

<sup>6</sup> **RRHH:** Recursos humanos.

Openbravo sigue un esquema de licenciamiento tipo software libre, sin costes ocultos o ataduras a un proveedor concreto. La licencia del producto asegura el acceso público al código fuente y la posibilidad de modificar dicho código libremente. Los clientes únicamente pagan por los servicios que ellos realmente quieren, cuando los necesitan.

Está construido bajo los módulos

- Compra
- Venta
- Almacén
- Contabilidad

El proceso de 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 industrias.

La gestión de almacenes incluye:

- Almacenes y ubicaciones (multi-almacén).
- Stock por producto en doble unidad (por ejemplo, en kilogramos y cajas).
- Atributos del producto en almacén personalizables (color, talla, descripción de calidad, etc.).
- Lote y número de serie.
- Impresión de etiquetas. Códigos de barras (EAN, UPC, UCC, Code, otras.).
- Gestión de bultos en almacén.
- Control de reposición.
- Trazabilidad configurable por producto.
- Movimiento entre almacenes.
- Gestión automática de salidas de stock (vaciado según existencias, con reglas de prioridad por caducidad, ubicación, etc.).
- Inventario físico. Planificación de inventarios. Inventario continuado.

- Informes de movimientos, seguimiento, stocks, entradas/salidas, caducidades, inventario, ubicaciones, etc. Informes personalizables.
- Integrado con Openbravo POS.
- Sincronización y control del stock en la misma tienda. (5)

Es un software libre que presenta las siguientes limitaciones:

- No permite manejar simultáneamente varios idiomas para los clientes: la versión estándar no permite indicar un idioma de comunicación distinto para cada cliente.
- El software es gratuito pero la implementación no es barata: el coste de implementación inicial ronda los 27.000 €.
- Presenta una usabilidad pobre: para una herramienta que será usada por unos pocos usuarios profesionales, las teclas de acceso directo y la navegación por tabulador son fundamentales; por lo que OpenBravo al ser 100% web tiene este aspecto muy poco trabajado.

En Cuba también existen alternativas de software nacionales certificadas, se describen a continuación una síntesis de las funcionalidades generales de algunos de ellos:

- **Versat-Sarasola:** El programa 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 (2), uno de ellos es el Control de inventarios.

En este módulo 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.

Actualmente lo utilizan alrededor de 200 entidades de varias provincias y en lo adelante lo introducirán más de dos mil 500 unidades presupuestadas del país, entre las que figuran organismos de la Administración Central del Estado, las direcciones municipales de finanzas, tesorerías, la ONAT y otros. (2)

- **RODAS XXI Versión 3.0:** Sistema multiempresa y multiusuario creado por CITMATEL para la automatización de la gestión empresarial.

El producto está presente en todas las provincias y en el municipio especial de la Isla de la Juventud. En estos momentos lo utilizan más de 450 entidades de instituciones como los ministerios de Ciencia, Tecnología y Medio Ambiente, y el de Economía y Planificación, los grupos Islazul e Inversiones Gran Caribe, así como GEOCUBA, entre otros. (6)

Contiene diferentes módulos que pueden usarse integrados o independientes, uno de ellos es el Inventario.

Además, cuenta con el módulo Administrador, que brinda mayor integralidad al sistema y garantiza facilidades adicionales durante su instalación y explotación. (2)

En este módulo se realiza la entrada o recepción de productos al almacén generando un documento según el tipo de entrada asociada y su comprobante de operaciones de forma automática, previa configuración del sistema para ello.

- **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.

Está formado por varios módulos entre los que se encuentra el de Inventarios. 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. (2)

El sistema maneja toda la información referida al Submayor de Inventarios de la entidad, garantizando el cuadro permanente con las respectivas cuentas de la Contabilidad General. Esta preparado para controlar el saldo de cada material en dos monedas (Moneda Libremente Convertible y la Moneda Nacional), a partir de los procedimientos vigentes en el país en cuanto a política monetaria. El tratamiento de las contabilizaciones se lleva a cabo de forma transaccional, por resúmenes diarios o de forma personalizada según se defina por parámetros. Incorpora la gestión de lotes de productos que posean dicho tratamiento y la entrada de medios materiales a almacenes mediante la gestión de claves de movimientos y las operaciones con los movimientos de entrada.

El sistema presenta las siguientes desventajas:

- Toda la gestión de la aplicación se hace a través de ventanas independientes haciendo más engorroso el trabajo por la cantidad de clic para realizar una operación.
- Solo trabaja sobre el sistema operativo Windows. (7)

### 1.3.1. Valoración crítica de los sistemas analizados

Aún cuando en el ámbito nacional se encuentran en explotación varios software nacionales y extranjeros que gestionan el proceso Control de inventario, se observo que la solución internacional SAP usa una tecnología que puede ser riesgosa en la situación actual de bloqueo que sufre nuestra nación, pues este está basado en la plataforma J2EE<sup>7</sup>, cuya máquina virtual es propiedad de la empresa SUN<sup>8</sup>

---

<sup>7</sup> **J2EE:** Java Platform, Enterprise Edition o Java EE es una plataforma de programación para desarrollar y ejecutar aplicaciones en lenguaje de programación Java.

<sup>8</sup> **SUN:** Stanford University Network, Red de la Universidad de Stanford.

perteneciente a los Estados Unidos país que nos bloquea por todos los medios el acceso a la tecnología informática. Estos sistemas en su mayoría son soluciones propietarias, lo que implica que se realicen pagos de licencias de software, capacitación, mantenimiento y reparación, aspectos que no son muy factibles por las condiciones económicas actuales, además no se tributa al cumplimiento del ideal de independizarnos completamente de las compañías que desarrollan soluciones propietarias. Por otro lado estas soluciones están desarrolladas en economías totalmente diferentes a la nuestra lo implicando que no se incluyan en sus soluciones aspectos como la dualidad monetaria y multimoneda, características totalmente nacionales.

En las soluciones nacionales se detectaron que las mismas se desarrollaron sobre plataformas obsoletas como es el caso del Versat-Sarasola que fue desarrollado en Delphi o sobre plataformas donde los programas son propietarios atentando contra el proceso de migración hacia software libre en el que esta enmarcado el estado, además son aplicaciones de escritorio, por lo que es necesario para su uso, la instalación de la misma en todas las PC de la entidad. En cambio, con una aplicación web solo bastaría tener acceso a la red para acceder al sistema.

Los sistemas SAP, Versat Sarasola manejan reportes de Inventario, pero lo realizan de una manera estática, cuando lo ideal es darle la posibilidad al usuario de filtrar datos escogidos según las características deseadas en la Recuperación. Por otro lado el sistema Rodas XXI no implementa reportes.

#### **1.4. Proceso de Desarrollo de software**

Un proceso de desarrollo de software tiene como propósito la producción eficaz y eficiente de un producto software que reúna los requisitos del cliente. Este proceso es intensamente intelectual, afectado por la creatividad y juicio de las personas involucradas. Aunque un proyecto de desarrollo de software es equiparable en muchos aspectos a cualquier otro proyecto de ingeniería, en el desarrollo de software hay una serie de desafíos adicionales, relativos esencialmente a la naturaleza del producto obtenido (8)

#### **ERP-PROD Modelo de desarrollo orientado a componentes**

El Modelo de desarrollo fue definido por la dirección del proyecto, con el objetivo de lograr un modelo estandarizado entre los equipos de desarrollo, que define de forma clara y precisa las responsabilidades de cada uno de los roles que se ven involucrados en el desarrollo de la solución.

La misma cuenta con los **Roles**: Jefe de línea, planificador, arquitecto principal, analista principal, especialista de calidad, desarrolladores, analistas y funcionales; organizados como se muestra en la figura 1.

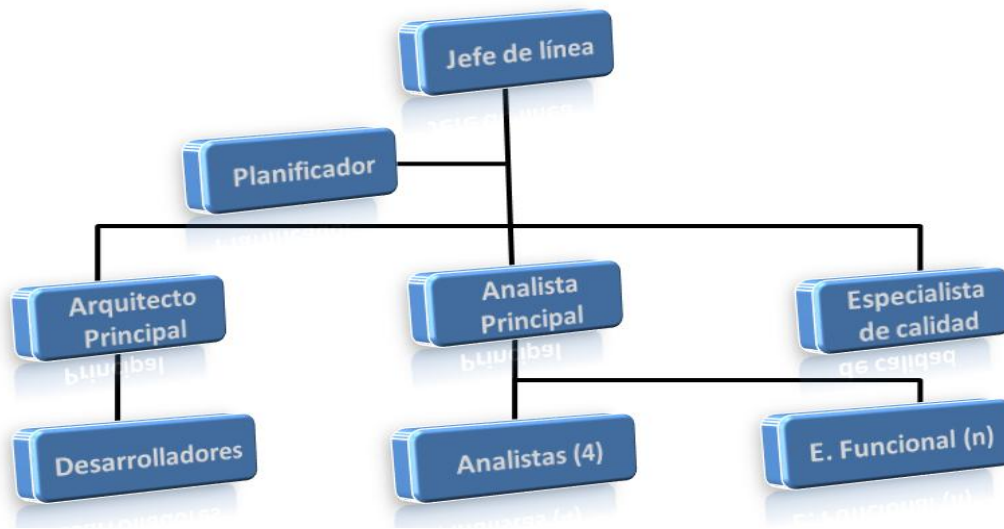


Figura 1 Representación de los roles definidos por el Modelo de desarrollo.

### Flujos de Trabajo/Artefactos:

#### ➤ 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 errores (En teoría es el Dpto. de Implantación)

La propuesta es desarrollada en colaboración con las Líneas de desarrollo del proyecto ERP-Cuba, y de acuerdo con las necesidades que han presentado cada una de ellas y teniendo en cuenta los principales riesgos con los que se cuentan en el proyecto. (9)

Consultar el ERP-PROD Modelo de Desarrollo orientado a componentes.

### **1.5. Tecnologías, lenguajes y Herramientas**

Es impresionante como cada día el Internet se desarrolla y avanza con grandes pasos dando lugar a nuevas e interesantes tecnologías, lenguajes y herramientas para la construcción y desarrollo de aplicaciones WEB<sup>9</sup>, que posibilitan la interacción y personalización de la información con el usuario

---

<sup>9</sup> **Web:** Término más usado para referirse al World Wide Web que es la Red Mundial de páginas o Documentos de texto entrelazados.

mediantes páginas WEB que haciendo uso de un navegador y la red conectora pueden ser visualizadas en cualquier lugar.

Las herramientas 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.

### 1.5.1. Herramientas

Actualmente se consideran a las Herramientas de Desarrollo de Software (HDS) como herramientas basadas en computadoras que asisten el proceso de ciclo de vida de una aplicación, consolidadas en la literatura en la forma de Ingeniería de software asistida por computadora (CASE, por sus siglas en inglés). Las HDS automatizan metodologías de software y desarrollo de sistemas. (10)

#### Visual Paradigm 6.3

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una rápida construcción de aplicaciones de calidad, mejores y a un menor costo. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. (11)

Visual Paradigm forma parte del IDE<sup>10</sup> de Eclipse. Está diseñado para desarrollar software con Programación Orientada a Objetos, examina y permite reducir la duración del ciclo de desarrollo brindando ayuda tanto a arquitectos, analistas, diseñadores y desarrolladores. Busca también automatizar tareas tediosas que pueden distraer a los desarrolladores.

Dentro de sus características fundamentales están:

- Multiplataforma: Soportada en plataformas Java para Sistemas Operativos Windows, Linux y Mac OS X

---

<sup>10</sup> **IDE:** Entorno de Desarrollo Integrado del inglés *Integrated Development Environment*.

- Interoperabilidad: Intercambia diagramas UML y modelos con otras herramientas. Soporta exportar e importar a XMI, XML y archivos Excel. Importa archivos de proyectos de Rational Rose. Integración con Microsoft Office Visio.
- Modelamiento de los Requisitos: Captura de requisitos con diagrama de requisitos, modelamiento de casos de uso y análisis textual.
- Colaboración de Equipo: Realiza el modelado en colaboración y simultáneamente con el Visual Paradigm TeamWork Server y Subversion.
- Generación de Documentación: Comparte y genera los diagramas y diseños en formatos como PDF, HTML y Microsoft Word.
- Editor de Detalles de Casos de Uso: Entorno todo en uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.
- Ingeniería de Código: Permite generación de código e ingeniería inversa en lenguajes como Java, C++, CORBA, IDL, PHP, XML Schema, Ada, Python, C#, VB .NET, ODL, Flash ActionScript, Delphi, Perl y Rugby. También permite ingeniería inversa.
- Modelado de Procesos de Negocio: Visualiza, comprende y mejora los procesos de negocio con la herramienta muy completa para estos procesos.
- Integración con Entornos de Desarrollo: Apoyo al ciclo de vida completo de desarrollo del software: análisis, diseño e implementación, en IDE como Eclipse, Microsoft Visual Studio, NetBeans, Sun ONE, Oracle JDeveloper, JBuilder y otros.
- Modelamiento de Bases de Datos: Generación de bases de datos, conversión de diagramas entidad-relación a tablas de base de datos, mapeos de objetos y relaciones, ingeniería inversa desde gestores de bases de datos.

### **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 desarrollado en Java (excesiva lentitud y consumo de memoria en algunos casos), ahora está basado en Eclipse, plataforma de

software de código abierto para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido" (12).

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).
- Detección de errores en tiempo real.
- 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.
- Zend Guard integration.
- Zend Studio Migration.
- Zend Studio Keymaps.
- PHPDocumentor.
- Fácil de depuración. (12)

### **Notepad++ 4.7.5**

Notepad++ es un editor gratuito de código fuente, que soporta varios lenguajes de programación y se ejecuta en MS Windows.



Este proyecto, basado en el componente de edición Scintilla<sup>11</sup>, está escrito en C++ utilizando directamente la API de win32 y STL (lo que asegura una velocidad mayor de ejecución y un tamaño más reducido del programa final), se distribuye bajo los términos de la Licencia Pública General de GNU (13).

Los lenguajes de programación admitidos son: C, C++, Java, C#, XML, HTML, PHP, JavaScript, archivos de recursos RC, makefile, Arte ASCII, doxygen, archivos INI, archivos por lotes (BAT), ASP, archivos VB/VBS, Shell script de UNIX, SQL, Objective-C, CSS, Pascal, Perl, Python, Lua, TeX, TCL, lenguaje ensamblador, Ruby, Lisp, Scheme, Smalltalk, PostScript, VHDL, FORTRAN, Ada, Caml, Autolt, KiXtart, Matlab y Verilog. Además, los usuarios pueden definir su propio lenguaje usando User Language ensamblador, Ruby, Lisp, Scheme, Smalltalk, PostScript, VHDL, FORTRAN, Ada, Caml, Autolt, iXtart, Matlab y Verilog. Además, los usuarios pueden definir su propio lenguaje usando User Language Define System incorporado, el cual hace al Notepad++ extensible, para tener resaltado de sintaxis y plegamiento de sintaxis.

Notepad++ no es solamente editor de código fuente, también, gracias a su velocidad, es una alternativa al bloc de notas. Con su implementación de navegación por pestañas, moverse entre los archivos de texto abiertos hace más confortable su uso para el usuario.

A continuación se enumeran algunas características de Notepad++:

- Sintaxis coloreada y envoltura de la misma.
- Sintaxis coloreada a elección del usuario.
- Autocompletado.
- Multidocumento.
- Multivista.
- Soporte para buscar/reemplazar expresiones regulares.
- Soporte completo para arrastrar y colocar.
- Posición dinámica en las vistas.
- Detección automática del estado del documento.
- Herramienta de Zoom.
- Funcionamiento bajo entornos multilingües.
- Puntos de marca.

---

<sup>11</sup> **Scintilla**: *Un componente libre de edición de código fuente.*

- Resalto de paréntesis y sangría.
- Grabación y reproducción de marcos (13).

### **PGAdmin III**

PgAdmin III es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Está escrita en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en sistemas operativos como: Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres.

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas \*nix), y puede encriptarse mediante SSL para mayor seguridad (14).

### **Postgre SQL Manager 2007**

Es una aplicación de alto desempeño para la administración y desarrollo de PostgreSQL Database Server. El programa trabaja con cualquier versión de PostgreSQL y soporta todas las últimas características de PostgreSQL, incluyendo espacios de tablas, nombres de argumentos en funciones y más. Su interfaz gráfica es sumamente atractiva e incluye un modo guiado de trabajo. Incluye las herramientas básicas de mantenimiento y administración.

Características:

- Soporte completo para PostgreSQL hasta la versión 8.3
- Administración y navegación rápida de bases de datos.
- Administración fácil de todos los objetos PostgreSQL.

- Administración efectiva de seguridad.
- Capacidades de exportación e importación de datos.
- Modo guiado para labores de mantenimiento. (15)

### SVN (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 lo 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. (16)

### TortoiseSVN 1.3.0

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)

### 1.5.2. Lenguajes.

#### Lenguaje Unificado de Modelado (Unified Modeling Language, UML)

UML es ante todo un lenguaje. Un lenguaje proporciona un vocabulario y una regla para permitir una comunicación. En este caso, este lenguaje se centra en la representación gráfica de un sistema.

Este lenguaje nos indica cómo crear y leer los modelos, pero no dice cómo crearlos. Esto último es el objetivo de las metodologías de desarrollo.

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

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

Aunque UML está pensado para modelar sistemas complejos con gran cantidad de software, el lenguaje es lo suficientemente expresivo como para modelar sistemas que no son informáticos, como flujos de trabajo (workflow) en una empresa, diseño de la estructura de una organización y por supuesto, en el diseño de hardware.

Está compuesto por tres clases de bloques de construcción:

- Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)
- Relaciones: relacionan los elementos entre sí.
- Diagramas: Son colecciones de elementos con sus relaciones. (18)

#### HTML (HyperText Markup Language)

El lenguaje HTML fue diseñado para estructurar documentos y presentarlos en forma de hipertexto, estableciendo relaciones unidireccionales entre ellos (hipervínculos). Inicialmente fue concebido para visualizar e interconectar el contenido de documentos electrónicos, considerando por ello un pequeño conjunto de etiquetas. Posteriormente, la ventaja que representaba la simplicidad de HTML se convirtió en un inconveniente, ya que su marcado no siempre cubría todos los aspectos de presentación que los

usuarios requerían. La solución adoptada fue el desarrollo de extensiones privadas del lenguaje, lo que complicó su estandarización. Las luchas comerciales entre las principales empresas de desarrollo de navegadores web condujeron a un lenguaje HTML que, aunque universalmente utilizado, carece de estandarización real. Desde finales de 1993 existen comités de estandarización, impulsados por el World Wide Web Consortium<sup>20</sup>, que tienden a un modelo único de HTML. (19)

### **JavaScript**

JavaScript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Nos permite ejecutar instrucciones como respuesta a las acciones del usuario, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

Es un lenguaje con muchas posibilidades, permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones, estructuras de datos complejas, etc. Además, JavaScript pone a disposición del programador todos los elementos que forman la página web, para que éste pueda acceder a ellos y modificarlos dinámicamente. (20)

Ventajas de JavaScript:

- Los programas escritos en este lenguaje no requieren de mucha memoria ni tiempo adicional de transmisión, por ser pequeños y compactos.
- JavaScript no requiere un tiempo de compilación; ya que los scripts se pueden desarrollar en un período de tiempo relativamente corto.
- Es independiente de la plataforma hardware o sistema operativo, y funciona correctamente siempre y cuando exista un navegador con soporte JavaScript.
- Durabilidad: Asegura la permanencia de una operación realizada, y aunque falle el sistema esta no podrá deshacerse.

## PHP 5.2

PHP, acrónimo de "PHP: Hypertext Preprocessor", es un lenguaje "Open Source" interpretado de alto nivel, especialmente pensado para desarrollos web y el cual puede ser embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl y es fácil de aprender. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil, aunque se pueda hacer mucho más con PHP. (21)

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. 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. (22)

Características de PHP:

- Es un lenguaje multiplataforma.

- Capacidad de conexión con la mayoría de los manejadores de base de datos que se utilizan en la actualidad como: MySQL, PostgreSQL y Oracle.
- Es un producto de código abierto, soportado por una gran comunidad de desarrolladores que se encargan de encontrar y reparar los fallos de funcionamiento.
- Permite las técnicas de Programación Orientada a Objetos.
- No requiere definición de tipos de variables.
- Tiene manejo de excepciones
- Posee una amplia documentación en su página oficial<sup>12</sup>, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Incorpora bibliotecas que contienen funciones integradas para realizar útiles tareas relacionadas con la Web. Puede generar imágenes GIF al instante, establecer conexiones a otros servicios de red, enviar correos electrónicos, trabajar con cookies y generar documentos PDF, todo con unas pocas líneas de código.

PHP 5.2.4 es una versión de PHP que además incluye:

- Soporte sólido para Programación Orientada a Objetos (OOP) con PHP Data Objects.
- Mejoras de rendimiento.
- Mejor soporte a XML.

### PostgreSQL 8.3

Es un servidor de base de datos relacional, libre. Ofrece soporte total para transacciones, disparadores, vistas, procedimientos almacenados, almacenamiento de objetos de gran tamaño. Se destaca en ejecutar consultas complejas, consultas sobre vistas, subconsultas 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 ofrece entre otras ventajas la 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, está disponible en casi cualquier sistema operativo. Además de presentar gran estabilidad y confiabilidad legendarias, en contraste a muchos

---

<sup>12</sup> Página oficial de PHP <http://www.php.net/manual/es/>

sistemas de bases de datos comerciales, es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad (23).

### 1.5.3. Tecnologías

Para los desarrolladores es de vital importancia el adecuado uso de las tecnologías a utilizar durante el proceso de implementación. Los marcos de trabajo y el uso de varias técnicas novedosas de programación viabilizan el desarrollo del software, apoyándose en el uso de una arquitectura adecuada que propicie una robustez al sistema.

#### 1.5.3.1. Marcos de Trabajo

Los Frameworks (Marcos de Trabajo), no son más que arquitecturas definidas para un determinado dominio de la aplicación que contiene un conjunto de componentes implementados y sus interfaces bien definidas, estos componentes se pueden utilizar, redefinir y crear nuevos componentes. (24)

#### Zend Framework

Es un Framework open 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 IoC<sup>13</sup> 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 inglés) para PHP 5.2+ con una base de datos con capas de abstracción incorporadas. 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 y el multilinguaje.

### **IoC**

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.

---

<sup>13</sup> **IoC:** *Inversión de Control del inglés Inversion of Control*

## Librería ExtJS 2.2

ExtJS empezó siendo un conjunto de librerías y extensiones para YUI (Yahoo! User Interface). Con el tiempo se convirtió en un FrameWork independiente y a principios de 2007 se creó una compañía para comercializar y dar soporte del FrameWork Ext. De esta forma Ext tiene dos tipos de licencias, LGPL y comercial. Básicamente, se puede usar para desarrollar, pero para obtener soporte hay que tener una licencia comercial.

Desde la versión 1.1 puede ejecutarse como una aplicación independiente. Posee las siguientes características:

- Ext.Element: Representa un elemento del árbol DOM. Muchas de las funciones de manipulación de los elementos tienen un parámetro opcional que permite realizar el cambio mediante un efecto de animación. El parámetro de animación puede ser un dato booleano o un objeto que incluye las opciones de la animación.
- Ext.BorderLayout: Esta clase representa un diseño común para ser usado en aplicaciones de escritorio.
- Ext.DomHelper: Utilidades para trabajar plantillas o DOM. Soporta el uso de DOM o fragmentos de HTML de forma transparente.
- Ext.UpdateManager: Proporciona soporte para actualización AJAX de los objetos Element. (25)

### 1.5.3.2. Arquitectura

El concepto de arquitectura software incluye los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura de software es el conjunto de decisiones significativas sobre la organización de un sistema, la selección de los elementos estructurales y sus interfaces de los cuales el sistema está compuesto junto con su comportamiento. Describe los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.

“... es la estructura de las estructuras del sistema, la cual comprende los componentes del software, las propiedades de esos componentes visibles externamente, y las relaciones entre ellos (...)” (26)

PRESSMAN

“Conjunto de decisiones significativas acerca de la organización de un sistema software, la selección de los elementos estructurales a partir de los cuales se compone el sistema, y las interfaces entre ellos, junto con su comportamiento, (...)” (27)

RUMBAUGH

### Patrón Modelo-Vista-Controlador (MVC)

El patrón fue descrito por primera vez a finales de la década de los setenta en los laboratorios de investigación de Xerox<sup>14</sup>. A partir de entonces su uso se ha ido extendiendo cada día más para la construcción de sistemas software con interfaz gráfica.

Este patrón arquitectónico separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes, como se representa en la figura 2:

**Modelo:** Administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador).

**Vista:** Maneja la visualización de la información.

**Controlador:** Controla el flujo entre la vista y el modelo (los datos).

Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases. Esta separación permite construir y probar el modelo, independientemente de la representación visual.

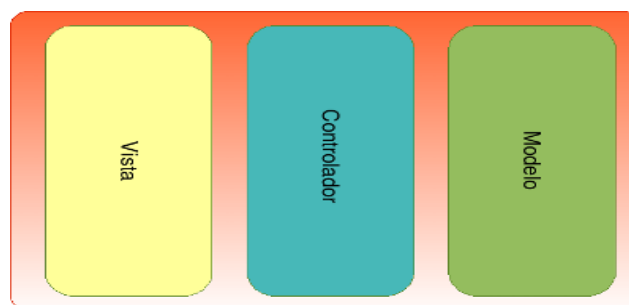


Figura 2 Representación del Patrón MVC.

<sup>14</sup> **Xerox Corporation:** es el proveedor más grande del mundo de fotocopiadoras de tóner (tinta seca) y sus accesorios

Entre las ventajas del estilo Modelo-Vista-Controlador están las siguientes:

**Soporte de múltiples vistas:** Dado que la vista se halla separada del modelo y no hay dependencia directa del modelo con respecto a la vista, la interfaz de usuario puede mostrar múltiples vistas de los mismos datos simultáneamente. Por ejemplo, múltiples páginas de una aplicación Web pueden utilizar el mismo modelo de objetos mostrado de maneras diferentes.

**Adaptación al cambio:** Los requerimientos de interfaz de usuario tienden a cambiar con mayor rapidez que las reglas de negocios. Los usuarios pueden preferir distintas opciones de representación, o requerir soporte para nuevos dispositivos como teléfonos celulares o PDAs<sup>15</sup>. Dado que el modelo no depende de las vistas, agregar nuevas opciones de presentación generalmente no afecta al modelo.

Una desventaja que tiene este modelo es el costo de actualizaciones frecuentes: Si el modelo experimenta cambios frecuentes, por ejemplo, podría desbordar las vistas con una lluvia de requerimientos de actualización.

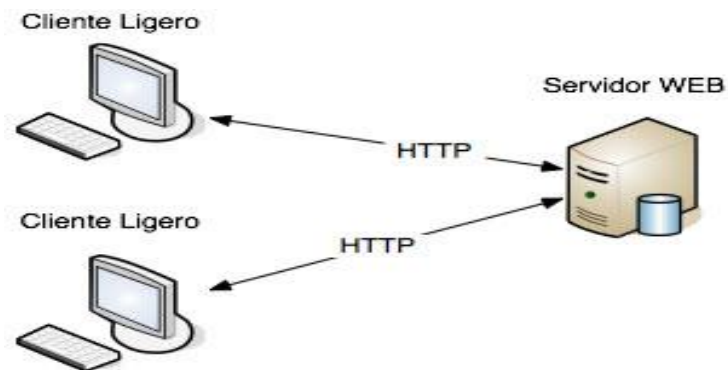
### **Arquitectura Cliente/Servidor**

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. (28)

Un ejemplo de esta arquitectura se representa en la figura 3 donde se observan dos clientes ligeros comunicándose con un servidor WEB mediante la conexión HTTP.

---

<sup>15</sup> *PDAs: Asistente Digital Personal (del inglés Personal Digital Assistant).*



**Figura 3 Arquitectura Cliente/Servidor.**

Características de la arquitectura Cliente/Servidor:

- Un servidor da servicio a múltiples clientes en forma concurrente.
- El servidor presenta una interfaz única y bien definida a todos sus clientes.
- El cliente no necesita conocer la lógica del servidor, sólo su interfaz externa.
- El cliente no depende de la ubicación física del servidor, ni del tipo de equipo físico en el que se encuentra, ni de su sistema operativo.
- Las funciones de Cliente y Servidor pueden estar en plataformas separadas, o en la misma plataforma
- Los cambios en el servidor no afectan al cliente.

### **1.4.3.3 Servidor Web**

El servidor Web es un programa que corre sobre el servidor que escucha las peticiones HTTP que le llegan y las satisface. Dependiendo del tipo de la petición, el servidor Web buscará una página Web o bien ejecutará un programa en el servidor. De cualquier modo, siempre devolverá algún tipo de resultado HTML al cliente o navegador que realizó la petición. El servidor Web es factor fundamental en el desarrollo de las aplicaciones del lado del servidor, que vayamos a construir, ya que se ejecutarán en él.

(29)

## Apache 2.0

Apache es un servidor web gratuito, potente y que nos ofrece un servicio estable y sencillo de mantener y configurar. Es indiscutiblemente uno de los mayores logros del Software Libre.

Destacaremos las siguientes características:

- Es multiplataforma, aunque idealmente está preparado para funcionar bajo Linux.
- Muy sencillo de configurar.
- Es de código abierto.
- 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, etc. (30)

Apache es una muestra, al igual que el sistema operativo Linux (un Unix desarrollado inicialmente para PC), de que el trabajo voluntario y cooperativo dentro de Internet es capaz de producir aplicaciones de calidad profesional difíciles de igualar.

La licencia Apache es una descendiente de la licencia BSD, no es GPL. Esta licencia te permite hacer lo que quieras con el código fuente (incluso forks y productos propietarios) siempre que les reconozcas su trabajo. (31)

### 1.4.3.4 Navegador

Un navegador, navegador red o navegador web (del inglés, *web browser*) es un programa que permite visualizar la información que contiene una página web (ya esté esta alojada en un servidor dentro de la World Wide Web o en uno local).

El navegador interpreta el código, HTML generalmente, en el que está escrita la página web y lo presenta en pantalla permitiendo al usuario interactuar con su contenido y navegar hacia otros lugares de la red mediante enlaces o hipervínculos.

## Mozilla Firefox 2.0

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 2.0 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 se tienen 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, puedes introducir un término en la barra de búsqueda y recibir respuesta inmediata del motor de búsqueda que hayas elegido. Puedes seleccionar un nuevo motor desde la barra de búsqueda en cualquier momento, y añadirlos desde los cientos de webs favoritas (32).

### 1.5.3.5 Otras tecnologías

#### AJAX

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. (33)

No es un lenguaje de programación en sí mismo sino la combinación de una serie de tecnologías que permiten una relación cliente-servidor más eficaz agilizando la respuesta de este último. Se considera una técnica de desarrollo web para crear aplicaciones interactivas. Las mismas se ejecutan en las computadoras-clientes lo que brinda ciertas ventajas en lo que respecta a la velocidad e interacción y traspase de contenido. Esta unión de tecnologías está compuesta, entonces, por cuatro tecnologías ya existentes, que operan de modo conjunto, a saber:

- El objeto XMLHttpRequest que permite el intercambio asíncrono con el servidor.
- Un formato que permite transferir datos de vuelta al servidor. Si bien el que más se utiliza es el XML, otros formatos pueden ser utilizados.
- DOM (Document Object Model), al cual los usuarios deben acceder a través de un lenguaje scripting como JavaScript para interactuar con la información solicitada.
- XHTML y hojas en estilo “cascada” (CSS) para el diseño de la página web, es decir, para mostrar el contenido. (34)

## XML

XML<sup>16</sup> es una tecnología en realidad muy sencilla que tiene a su alrededor otras tecnologías que la complementan y la hacen mucho más grande y con posibilidades mucho mayores.

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. El XML juega un papel fundamental 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 y fácil. Además 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. (35)

Ventajas de XML:

- Las aplicaciones se pueden generar rápidamente y su mantenimiento es más sencillo.

---

<sup>16</sup> XML: Lenguaje de marcado extensible (Del Inglés Extensible Markup Language).



- Separa los datos de la presentación y del proceso, lo que permite mostrar y procesar los datos al gusto deseado con sólo aplicar distintas hojas de estilo y aplicaciones.
- La información es más accesible y reutilizable, por la flexibilidad de las etiquetas de XML que permiten su utilización sin tener que amoldarse a reglas específicas de un fabricante.

### 1.6. Conclusiones parciales

Con el estudio de algunos sistemas que permiten el control de inventarios existentes en el mundo y en Cuba se llega a la conclusión de que los mismos han servido para agilizar los procesos de recepción de productos en un almacén, teniendo un mayor control de los recursos y erradicando en gran medida los errores que se cometían cuando se hacía todo de forma manual aunque tienen la limitante de abordar solamente partes del problema de la gestión de inventario y en particular de la recepción de medios materiales en una entidad. Por tanto se decidió elaborar un software multiplataforma, que brinde facilidades de uso a los usuarios, que reúna características que permitan su implantación en todas las entidades empresariales y unidades presupuestadas del país y que lleve un control del proceso.

En la confección del mismo se utilizará como metodología de desarrollo ERP-PROD, como lenguaje de programación del lado del servidor PHP, para una mejor interacción entre el usuario y la aplicación se utilizaron las potencialidades que brinda ExtJS, como gestor de base de datos PostgreSQL aprovechando todas las potencialidades que este brinda, la programación por el lado del cliente XHTML, Java Script, CCS. Con auxilio de el ZendStudio para el desarrollo del módulo y el Visual Paradimg para el modelado de los diagramas de componente.

# Capítulo 2

## Características del Sistema

### 2.1. Introducción

En este capítulo se tratan los elementos inmersos en el desarrollo del subsistema. Se realiza una valoración de la propuesta del diseño del sistema exponiendo las principales ventajas y deficiencias del mismo. Se expone la explotación de la arquitectura y las posibilidades proporcionadas por el framework y las librerías utilizadas en la programación de la aplicación, con el objetivo de brindar mayor comprensión de las funcionalidades de los componentes implementados. Además se describen y analizan las complejidades de algoritmos no triviales estimados entre los más importantes dentro del sistema y por último se realizan las descripciones de clases y operaciones utilizadas.

### 2.2. Flujo del proceso

#### 2.2.1. Valoración crítica de los artefactos propuestos por los analistas

Los requisitos funcionales descritos por los analistas del sistema facilitaron el entendimiento de los procesos a implementar permitiendo una buena comprensión del problema y posibilitando la identificación de las clases y las funcionalidades a desarrollar. Tomándose como base la descripción detallada de los requisitos funcionales, se puede establecer una estrategia de trabajo que permita la implementación de la capa lógica, de presentación y de datos de la aplicación.

#### Especificación de los requerimientos del software

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

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

- R1- Visualizar informe de diferencias
- R2- Visualizar informe de recepción
- R3- Gestionar informe de recepción
  - R3.1- Crear informe de recepción
  - R3.2- Buscar informes de recepción
  - R3.3- Búsqueda avanzada de informe de recepción
  - R3.3- Modificar informe de recepción

- R3.4- Eliminar informe de recepción
- R3.5- Cancelar informe de recepción
- R3.6- Contabilizar informe de recepción
- R4- Gestionar productos al informe de recepción
  - R4.1- Adicionar productos al informe de recepción
  - R4.2- Buscar productos del informe de recepción
  - R4.3- Modificar productos del informe de recepción
  - R4.4- Eliminar productos del informe de recepción
  - R4.5- Registrar diferencias en el informe de recepción
  - R4.6- Modificar diferencias en el informe de recepción
  - R4.4- Eliminar diferencias en el informe de recepción
  - R4.7- Crear tarjeta de estiba
- R5- Gestionar informes de diferencias
  - R5.1- Crear informe de diferencias
  - R5.1- Eliminar informe de diferencias
  - R5.2- Modificar informe de diferencias
  - R5.3- Buscar un informe de diferencia
  - R5.4- Búsqueda avanzada de un informe de diferencias
  - R5.5- Aprobar informe de diferencias
- R6- Gestionar productos al informe de diferencias
  - R6.1- Adicionar productos al informe de diferencias
  - R6.2- Buscar productos del informe de diferencias
  - R6.3- Modificar productos del informe de diferencias
  - R6.4- Eliminar productos del informe de diferencias
- R7- Gestionar lotes de los productos
  - R7.1- Registrar lotes del producto
  - R7.2- Eliminar lotes del producto
  - R7.3- Modificar lotes del producto
- R8- Gestionar números de serie de los equipos

- R8.1- Adicionar número de serie del equipo
- R8.2- Modificar número de serie del equipo
- R8.3- Eliminar número de serie del equipo
- R9- Ubicar productos en las áreas de almacenamiento
  - R9.1- Asignar ubicación al producto
  - R9.2- Modificar ubicación
  - R9.3- Eliminar ubicación

### 2.2.2. Objetos de automatización

Para el proceso de Recepción de productos se quiere automatizar la generación de documentos tanto de recepción por diferentes criterios (Factura, Transferencia, Vale de devolución, Producción terminada, Conduce) como los de diferencia, además buscar por diferentes criterios dichos documentos. [\(Para más información ver Anexos 3 y 4\)](#)

### 2.2.3. Información que se maneja

- **Documento de compra:** Este documento puede ser una factura o Conduce, documentos usados en todas las entidades. También puede ser una Transferencia o Vale de devolución, cuando entre las entidades involucradas no median relaciones de pago en la entrega-recepción. [\(Para más información ver Anexos 1,5 y 9\)](#)
- **Factura:** Permite anotar de forma manual los datos principales de los documentos que originan movimientos en los registros contables, materiales y financieros, registrar la habilitación de todos los libros, registros y demás modelos vinculados tanto al sistema de contabilidad material como al sistema de contabilidad financiera, incluidos los modelos de riguroso control. Permite dejar definida la responsabilidad individual por la entrega y recepción de los libros, registros o modelos que han sido habilitados, confeccionados, recibidos o entregados, así como registrar la recepción y entrega de los modelos pre-numerados. [\(Para más información ver Anexo 1\)](#)
- **Transferencia:** Permite amparar desde el punto de vista documental hasta su destino final la entrega-recepción de productos entre almacenes de una misma entidad sin que medie pago

alguno, pudiéndose utilizar también entre centros de costo y servir de base a las anotaciones de las entregas de productos entre almacenes y centros de costo.

- **Conduce:** Permite formalizar las entregas parciales de productos a incluir en una “Factura”, amparar documentalmente hasta su destino final el tránsito de los productos y justificar las entregas en los casos en que no existan condiciones para efectuar la facturación en los lugares o en el momento de efectuar la entrega. ([Para más información ver Anexo 9](#))
- **Informe de recepción:** Permite formalizar la recepción de los productos en los almacenes, cualquiera que sea su procedencia, respaldar las anotaciones de las entradas en inventario en los controles de existencia correspondientes. ([Para más información ver Anexo 3](#))
- **Tarjeta de identificación y control de estiba:** Permite controlar las existencias en unidades físicas de los productos en almacén, mediante el registro del movimiento de entradas y salidas de estos, identificar de forma rápida los que se encuentran almacenados, controlar la fecha de vencimiento, categoría y otras especificidades de los mismos. ([Para más información ver Anexo 8](#))
- **Informe de diferencias:** Permite notificar al transportador las diferencias por sobrantes, faltantes, averías o mermas en exceso de las normas, determinadas en la recepción detallada, en caso de que se establezca su responsabilidad, notificar al suministrador, errores de cálculos, sumas y diferencias en precios, registrar los bultos sobrantes, faltantes o averiados que se detecten al recibir estos, servir de base para aplicar la responsabilidad material cuando corresponda. ([Para más información ver Anexo 4](#))

#### 2.2.4. Propuesta del sistema

Para llevar a cabo el proceso recepción de productos de un almacén, inicialmente el sistema debe permitir la creación del Informe de recepción que ampara la entrada de productos al almacén o la búsqueda de los documentos de recepción, ya existentes, por diferentes criterios. Si se desea verificar la existencia de un Informe de recepción, para ejecutar cualquier acción sobre el mismo, se deciden los distintos criterios, por cuales se desee realizar una búsqueda, estas pueden ser: por año, estado, número del documento, creado por y por rango de fecha. Luego de haber finalizado dicha búsqueda, si se encontró algún Informe de recepción que cumpla con dichos criterios, entonces el sistema deberá verificar si este documento esta siendo usado o no por otro usuario. Si no está siendo usado, se deberá verificar

el estado y en dependencia de este y según los privilegios del usuario registrado se podrán ejecutar distintas acciones.

Se le dará la posibilidad al usuario de visualizar el documento en forma de reporte no permitiendo cambios sobre este, se permitirá la actualización de dicho Informe de recepción, es decir, se podrán efectuar cambios, y actualizar los productos. Registrar diferencias a los productos adicionados en el proceso de recepción según la cantidad, ya sea sobrante o faltante y según la calidad de los mismos. Ubicar las cantidades de productos recibidas en las distintas áreas de almacenaje. Registrar los números de serie a los medios materiales que son equipos en correspondencia con la cantidad recepcionada. Se actualizarán las existencias de los productos en el almacén cuando se contabilicen los informes de recepción, en caso de que existan productos con diferencias registradas el sistema generará un informe de diferencias. El sistema permitirá crear nuevos documentos donde una vez elaborados, se podrán visualizar, modificar, eliminar, etc.

Además admitirá crear el Informe de diferencias o la búsqueda de los informes de diferencias, ya existentes, por diferentes criterios. Si se desea verificar la existencia de un Informe de diferencias para ejecutar cualquier acción sobre él, se deciden los distintos criterios por cuales realizar su búsqueda, estos pueden ser: por año, número del documento, año del documento que lo origina, número del documento que lo origina, creado por, aprobado por y por rango de fecha. Luego de haber finalizado dicha búsqueda, si se encontró algún Informe de diferencias que cumpla con dichos criterios, entonces el sistema deberá verificar si este documento está siendo usado o no por otro usuario. Si no está siendo usado, se deberá verificar el estado y en dependencia de este y según los privilegios del usuario registrado se podrán ejecutar distintas acciones.

Se mostrará el documento en formato de reporte no permitiendo cambios sobre este. Se permitirá la actualización de los datos de dicho Informe de diferencias y actualizar los productos. Se consentirá según el estado del informe y la cantidad de productos que este tenga asociado, aprobar el informe de de diferencias. El sistema concederá adicionar productos del informe de recepción que lo originó, y en caso de existir algún producto en la factura, que no se tuvo en cuenta en el proceso de recepción, se permite adicionar el mismo de los productos que existen en el almacén, así como eliminarlos del informe de diferencias e insertarles diferencias a estos por conceptos de importes y precios. El sistema permitirá crear nuevos documentos donde una vez elaborados, se podrán visualizar, modificar, aprobar, eliminar, etcétera.

Luego de analizar detenidamente lo anteriormente expuesto, se concluye que es factible implementar una aplicación Web, basada en el uso del estilo arquitectónico Modelo – Vista – Controlador, usando el lenguaje de programación PHP y como soporte para los datos el Sistema Gestor de Bases de Datos PostgreSQL. Se propone utilizar además, para facilitar el trabajo, el Zend Framework, desarrollado en PHP bajo el estilo arquitectónico MVC, muy utilizado en la actualidad. Dicha aplicación, automatiza los procesos que integran la recepción de medios materiales en los almacenes dentro de las entidades empresariales y unidades presupuestadas del país, además la documentación asociada a su desarrollo sirve de base para futuras implementaciones, mantenimientos o mejoras al software desarrollado con este mismo propósito.

### **2.3. Descripción de la implementación**

#### **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 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.

En la figura 4 se muestra el diagrama de despliegue de la aplicación.



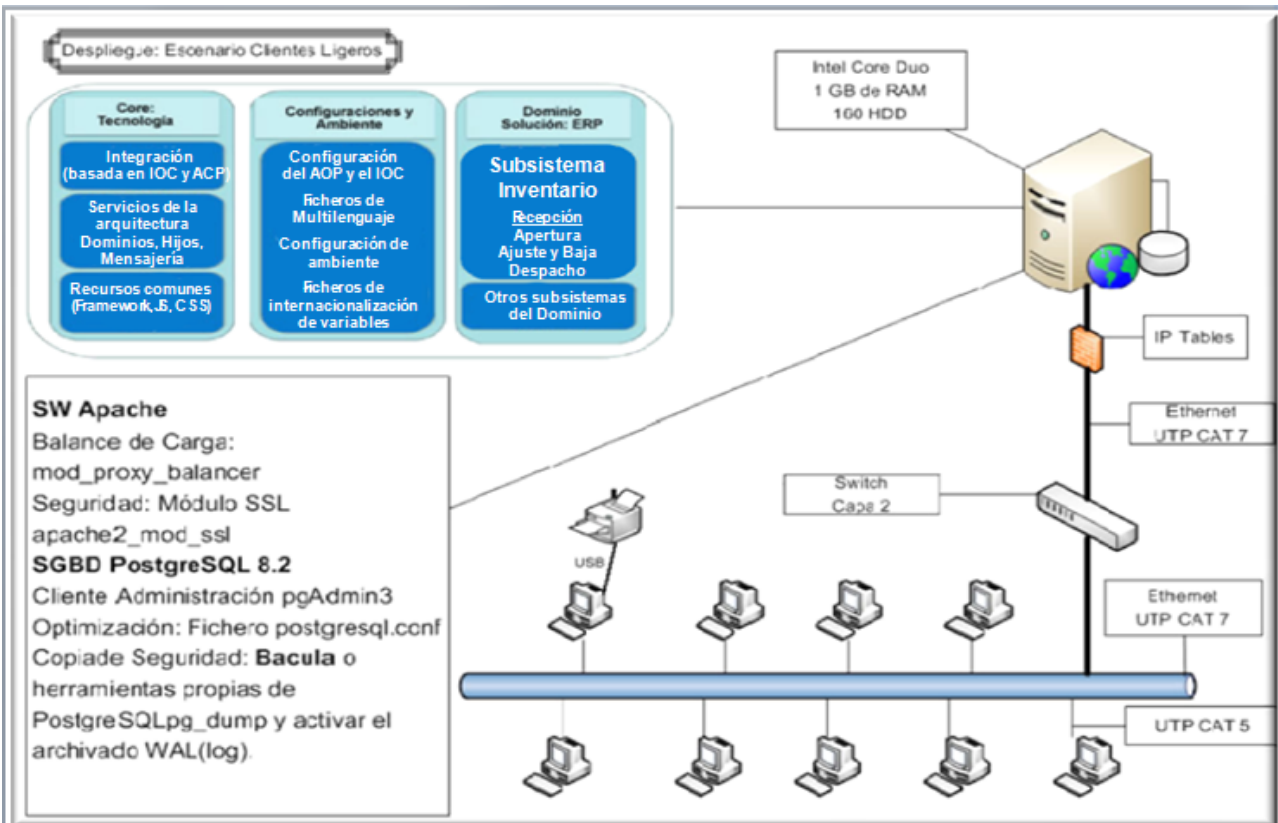


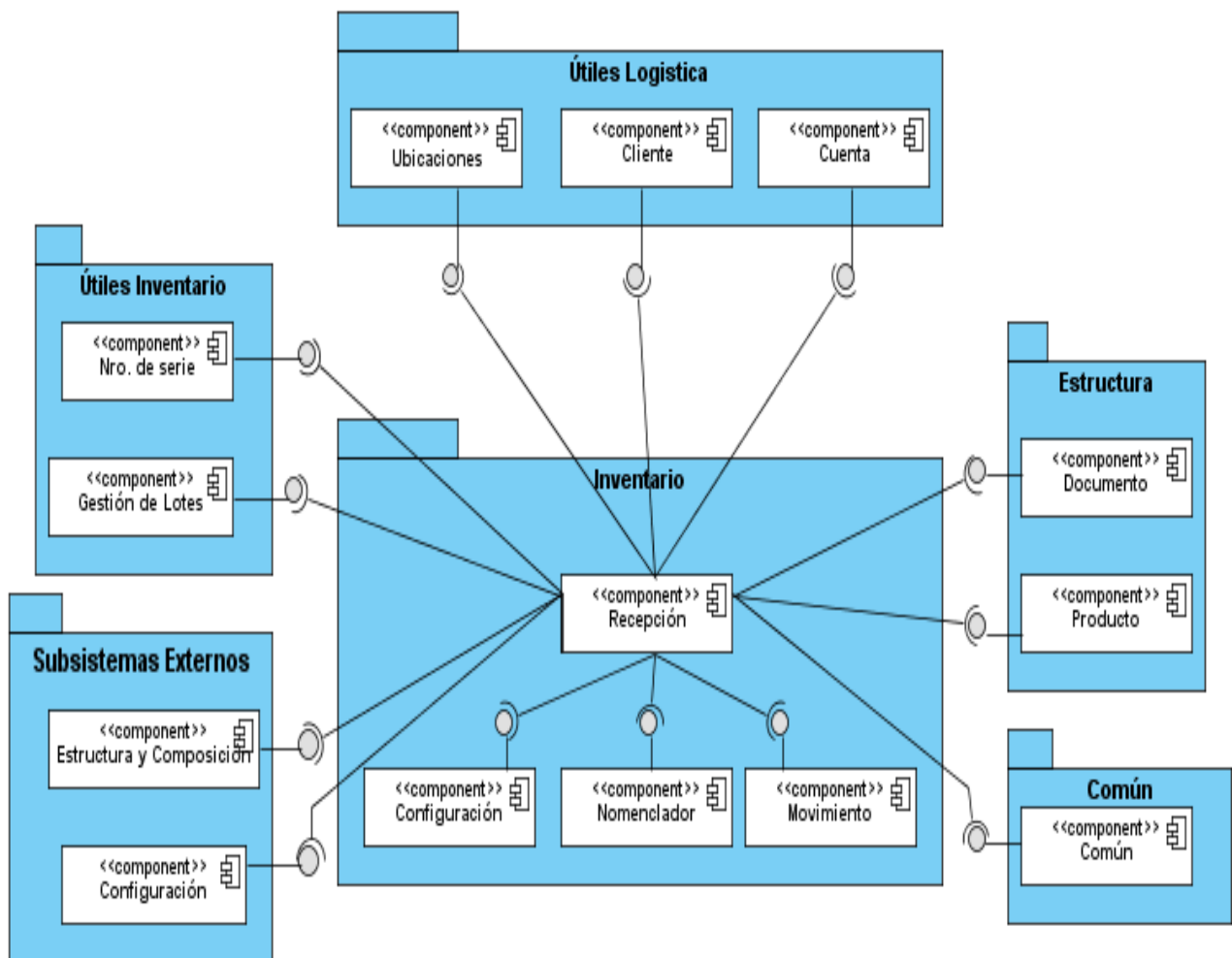
Figura 4 Diagrama de despliegue.

### 2.3.2. Diagrama de componentes

El diagrama de componentes muestra un conjunto de elementos del modelo tales como componentes, subsistemas de implementación y sus relaciones. Se utiliza para modelar la vista estática de un sistema. Muestra la organización y las dependencias lógicas entre un conjunto de componentes software.

Visto de otro modo un diagrama de componentes puede ser un tipo especial de diagrama de clases que se centra en los componentes físicos del sistema.

En las figuras 5 y 6 se muestran el diagrama de integración de componentes y el diagrama de componentes respectivamente correspondientes al módulo de Recepción. Cada paquete en los diagramas representa una división física del sistema.



**Figura 5 Diagrama de Integración entre Componentes.**

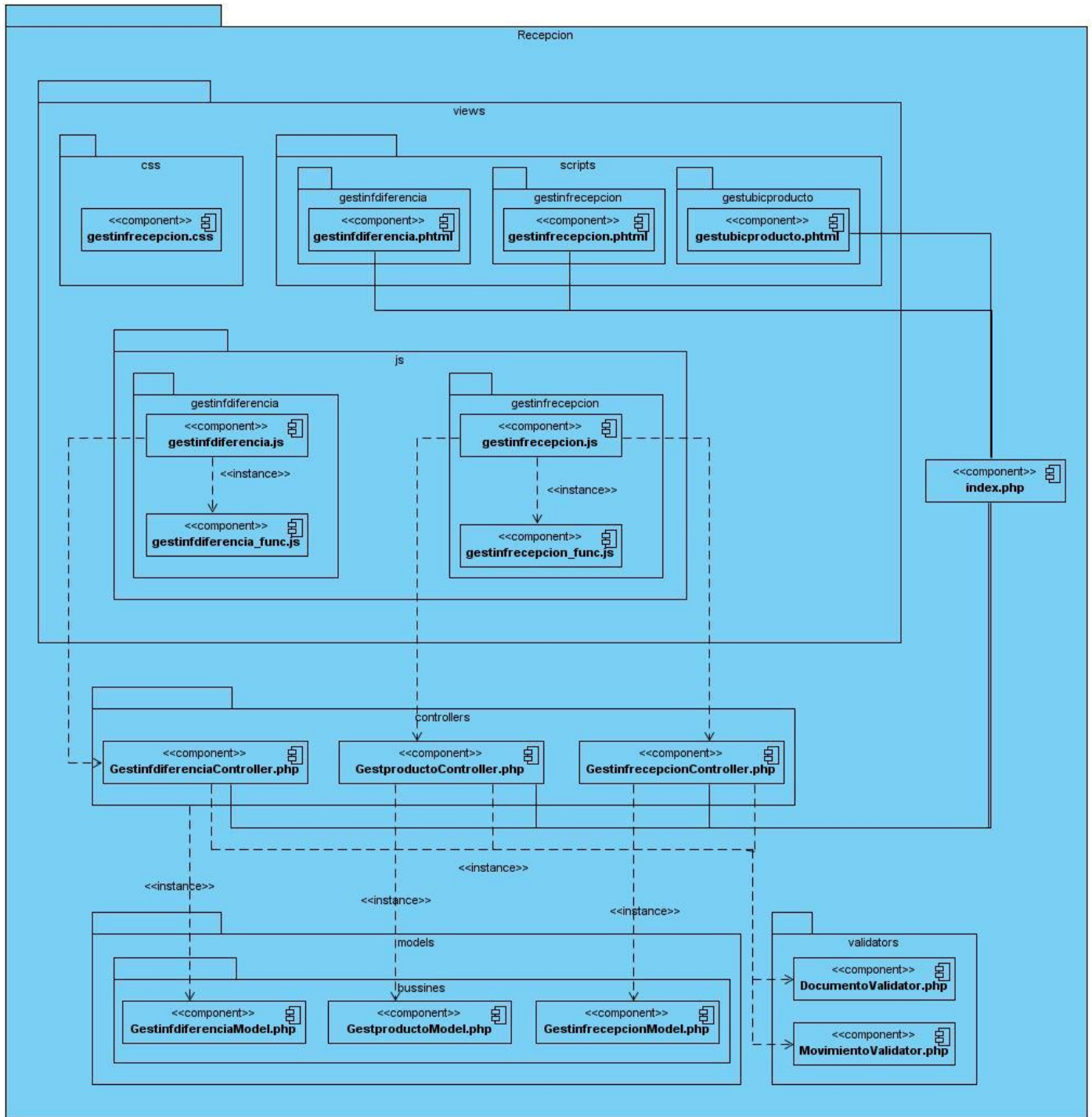


Figura 6 Diagrama de Componentes.

### 2.3.3. Integración entre componentes

La aplicación está definida por capas, en la capa de funcionamiento se utiliza el patrón MVC que consiste en el flujo de los datos desde la vista hacia la capa de datos y viceversa, pasando por los diferentes elementos que componen la arquitectura. Esta consta de cuatros nodos de integración, el que encontramos entre la vista y el controlador, el que está entre 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, tal y como se muestra en la figura 7. Todo el código dentro un de mismo componente utiliza llamadas a métodos o eventos de forma directa. La comunicación entre diferentes módulos y componentes se realiza mediante llamadas a la inversión de control. 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.

Cada componente tiene su registro de los datos de los módulos en un fichero XML que será mapeado por el framework para el funcionamiento del mismo, dicho fichero tiene por nombre IoC y registra las funcionalidades que ofrecen los métodos de las clases control de los componentes del sistema. La base de datos es accedida de forma directa mediante controladoras y los componentes rehusados son integrados mediante interfaces sencillas, garantizando así una total integración de las capas en el sistema.

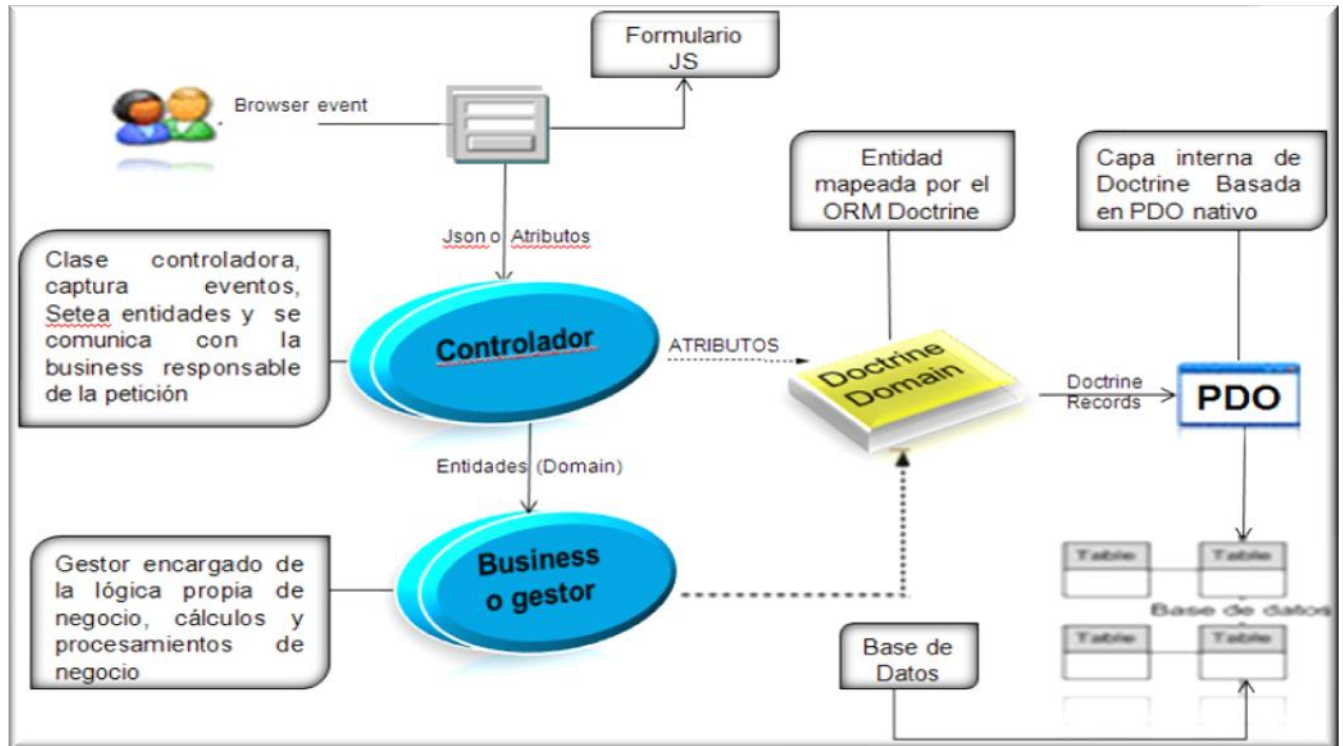


Figura 7 Diagrama de colaboración entre clases de la arquitectura.

### 2.3.4. Normas y Estándares de codificación

Los 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. Este apartado constituye una visión de la estandarización del código que se utilizó en el desarrollo de la aplicación y desde un punto de vista arquitectónico.

Los estándares de codificación en el marco del ERP permiten una mejor integración entre las líneas de producción, estableciendo de un modelo de desarrollo que conlleva a lograr un código más legible y reutilizable, de tal forma que se pueda facilitar su mantenimiento a lo largo del tiempo.

Entre las normas y estándares de codificación utilizados en la implementación de este producto se pueden mencionar:

### ➤ Estándares de Nomenclatura

En el estándar de nomenclatura se define como serán nombradas las clases atendiendo entre otras cosas al tipo de clases que sean. Los nombres de las clases comienzan con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación PascalCasing. Con sólo leerlo se reconoce el propósito de la misma. Las clases controladoras después del nombre llevan la palabra: "Controller" y las clases de los modelos que están dentro de Business llevan la palabra "Model", las de los dominios reciben el nombre de las tablas de la base de datos y las que se encuentran dentro de los dominios bases comienzan con la palabra "Base" seguido del nombre de la tabla en la Base de Datos.

- Clases controladoras.  
Ejemplo: GestionarUsuarioController
- Clases de los modelos  
Ejemplo: MonedaContModel
- Domain (Dominio)  
Ejemplo: User
- Generated (Dominio bases)  
Ejemplo: BaseUser

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

Para las variables se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCasing, y comenzando con un prefijo según el tipo de datos.

Ejemplo: arrMoneda

Los prefijos a utilizar en la creación de variables serán los siguientes:

Tipos de Datos	Prefijos
Arreglos	arr
Objetos	obj
Enteros	int
Cadena	str
float	flt
Boolean	Boo

**Tabla 1 prefijos a utilizar en la creación de variables.**

El nombre a emplear para los atributos se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se empleará notación CamelCasing. Además en caso de ser un objeto se comienza con:”\_” y después se escribe el nombre.

Ejemplo: intMoneda=dinero

objMoneda=\_dinero

➤ **Normas de comentariado**

Es una necesidad comentar todo lo que se haga dentro del desarrollo, es decir, establecer las pautas que conlleven a lograr un código más legible y reutilizable, de manera que se pueda facilitar su mantenimiento a lo largo del tiempo.

Los comentarios deben ser lo bastante claros y precisos de forma tal que se entienda el propósito de lo que se ha desarrollando.

Se pueden encontrar comentarios de código antes de la declaración de una clase, una función complicada y antes de la declaración de la misma como se muestra en el ejemplo de la figura 8, en las llaves indicando el fin de la misma y lo que se está cerrando.

```

/**
 * GestinrecepcionController
 * Clase controladora para la gestion de Informes de Recepción
 * @author Yurriel Martínez Ordaz
 * @package Inventario
 * @subpackage Recepción
 * @copyright UCID-ERP Cuba
 * Versión 1.0-0
 */

```

**Figura 8 Comentariado de la declaración de una clase**

### ➤ Estilo del Código

En la implementación cuando se valla a escribir una sentencia en php la forma de utilizar las etiquetas del mismo es la siguiente:

```

<?php
//código
?>

```

La política de sangría ha utilizar en la implementación es por tabulaciones. Las clases se comienzan a declarar pegado al margen izquierdo, después de poner el nombre de la clase se pone un espacio y se abre llave en la misma línea. Se establecieron pautas para utilización de sangrías dentro de las declaraciones dentro de métodos, bloques y switch, al igual que para la apertura y cierre de llaves, el espaciado en blanco en las declaraciones, los controles, las Expresiones y Arreglos.

Todo este estándar fue tomado del documento Normas y estándares de codificación de la línea de arquitectura del ERP el cual establece estos y otros conjuntos de normas por las cuales se rigió el desarrollo logrando una mejor organización y entendimiento de lo realizado. (36)

## 2.4. Análisis de reutilización de componentes, código o módulos

Dentro del subsistema inventario existen varios componentes que poseen funcionalidades que son necesarias para dar una solución más elegante, reutilizable y factible a los procesos de Recepción.



Haciendo uso de la reutilización de estos componentes se evita el duplicado de líneas de código en los restantes procesos. Dichos componentes y sus principales objetivos se describen a continuación.

### **Componente Documento**

Este componente permite la gestión de distintos tipos de documentos, entre ellos el Informe de recepción, y el Informe de diferencias. Además de brindar facilidades de organización y reutilización reduce considerablemente los esfuerzos de desarrollo y mantenimiento. El diseño del componente Documento mediante el uso de la herencia está destinado a propiciar una mínima dependencia entre clases y una mayor flexibilidad a la hora de realizar modificaciones internas o la aparición de nuevos requisitos ya que solo se afectaría en menor medida la estructura del sistema.

### **Componente Producto**

Este componente es el encargado de proporcionar de forma rápida y precisa el listado de todos los productos existentes en el almacén de la entidad correspondiente, acelerando así el proceso de recepción de productos en dicho almacén. Brinda además la posibilidad de, en caso de no existir un determinado producto en el almacén, registrarlo de forma rápida y sencilla, permitiendo así que se realice la recepción.

### **Componente Movimiento**

Este componente constituye la capa intermedia donde se relacionan los componentes Producto y Documento. Dicho componente facilita todas las operaciones que son necesarias para operar con los movimientos de ese documento. Un movimiento es una copia que se le hace al producto para no afectar los valores reales del mismo y una vez contabilizado el documento serán afectados los valores de producto con los del movimiento. Brinda además facilidades de organización y reutilización. Reduce considerablemente los esfuerzos de desarrollo y mantenimiento, mejora en gran medida la seguridad y eficiencia interna. Proporciona una mayor flexibilidad ante cambios entre dichos componentes y encapsula además funciones específicas evitando así una alta dependencia.

### **Componente Número de Serie**

Este componente es el encargado del registro de los números de serie únicos a los productos del documento que, por su naturaleza se consideran equipos, la nacionalidad y el año de fabricación.

### Componente Ubicaciones

Este componente es el encargado de ubicar una cantidad determinada de un producto en un área que se haya determinado previamente por la configuración general de la entidad, posibilitando una mejor organización y control de la ubicación de los productos en las áreas de almacenaje.

### Componente Lote

Este componente brinda la posibilidad, según la cantidad de producto recepcionada, clasificarla en los distintos lotes que se hayan recibido y registrar la fecha de vencimiento del mismo, para tener un mejor control de las existencias contra fecha de vencimiento.

### Componente Cliente

El componente cliente tiene la funcionalidad de mostrar todos los clientes y proveedores que posee la empresa.

## 2.5. Descripción de las principales clases a utilizar

En este tópico se describirán las principales clases del módulo Recepción así como sus métodos más significativos, los cuales contribuirán al cumplimiento satisfactorio de los requisitos propuestos por los analistas.

### ➤ Gestionar Informe de Recepción

<b>Nombre: GestinfrecepcionController.</b>	
<b>Tipo de clase: Controladora.</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad</b>	
<b>Nombre.</b>	<b>Descripción.</b>
init()	Constructor de la clase.
gestinfrecepcionAction()	Responsabilidad encargada de validar todos los datos necesarios para gestionar Informes de

	Recepción y redireccionar a la interfaz.
cargardocumentosrepcionAction()	Responsabilidad encargada de cargar todos los Informes de Recepción y mostrarlos al usuario para su posterior gestión.
guardarinfrecepAction() modificardocumentoAction() eliminarinfrecepAction()	Responsabilidades encargadas de la gestión de los Informes de Recepción.
contabilizarAction()	Responsabilidad encargada de contabilizar un Informe de Recepción.
cancelarrepcionAction()	Responsabilidad encargada de cancelar un Informe de Recepción previamente contabilizado.
productorepcionAction()	Responsabilidad encargada de mostrar los productos asociados a un Informe de Recepción.
nomprodAction()	Responsabilidad encargada de cargar los datos del nomenclador de productos
buscarproductosdocAction()	Responsabilidad encargada de cargar los productos que tiene un Informe de Recepción.
adicionardatprodAction() adicionardenomprodAction() modificarprodrepcionAction() eliminarprodAction()	Responsabilidades encargadas de gestionar los productos de un Informe de Recepción
cargargridmonedasAction()	Responsabilidad encargada de cargar los datos del grid de monedas del productos de un Informe de Recepción

**Tabla 2 Clase utilizada en la solución (GestinfrepcionController)**

➤ **Gestionar Informe de Recepción Model**

<b>Nombre: GestinfrecepcionModel.</b>	
<b>Tipo de clase: Modelo.</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad</b>	
<b>Nombre.</b>	<b>Descripción.</b>
GestinfrecepcionModel()	Constructor de la clase.
getRecepcion(\$datos)	Responsabilidad encargada de buscar y cargar todos los Informes de Recepción y mostrarlos al usuario para su posterior gestión.
adicionarInfRecep(\$datos) modificarInfRecep(\$iddocumento,\$datos) eliminarInfRecep(\$iddocumento,\$alias)	Responsabilidades encargadas de la gestión de los Informes de Recepción.
contabilizar(\$alias,\$iddocumento)	Responsabilidad encargada de contabilizar un Informe de Recepción.
cancelar(\$res,\$iddocumento)	Responsabilidad encargada de cancelar un Informe de Recepción previamente contabilizado.

**Tabla 3 Clase utilizada en la solución (GestinfrecepcionModel)**

➤ **Gestionar Informe de Diferencias Controller**

<b>Nombre: GestinfdiferenciaController.</b>	
<b>Tipo de clase: Controladora.</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad</b>	
<b>Nombre.</b>	<b>Descripción.</b>

init()	Constructor de la clase.
gestinfdiferenciaAction()	Responsabilidad encargada de validar todos los datos necesarios para gestionar Informes de Diferencias y re direccionar a la interfaz.
cargardocumentosdiferenciaAction()	Responsabilidad encargada de cargar todos los Informes de Diferencias y mostrarlos al usuario para su posterior gestión.
cargarinfdiferenciaAction()	Responsabilidades encargadas de cargar los datos del Informe de Diferencias a crear.
guardarinfdiferenciaAction() modificarinfdiferenciaAction() eliminarinfdiferenciaAction ()	Responsabilidades encargadas de la gestión de los Informes de Diferencias.
productodiferenciaAction()	Responsabilidad encargada de insertar las diferencias por concepto de calidad o cantidad a un producto.
elimdiferenciaAction()	Responsabilidad encargada de eliminar las diferencias a un producto.
precioimportediferenciaAction()	Responsabilidad encargada de insertar las diferencias por concepto de precio e importe a un producto.
cargarcalidadAction()	Responsabilidad encargada de cargar los datos del nomenclador de calidad.
buscarproddifAction()	Responsabilidad encargada de cargar los productos que tiene un Informe de Diferencias.
cargarprodrepcionAction()	Responsabilidades encargadas de cargar los productos de un Informe de Recepción asociado a un Informe de Diferencias.
adicionarprodderecepAction()	Responsabilidad encargada de adicionar productos del Informe de Recepción al Informe de Diferencias.

adicionariedadatprodAction()	Responsabilidad encargada de adicionar productos del nomenclador de productos al informe de Diferencias.
eliminarproductoAction()	Responsabilidad encargada de eliminar productos del Informe de Diferencias.

**Tabla 4 Clase utilizada en la solución (GestinfDiferenciaController)**

➤ **Gestionar Informe de Diferencias Model**

<b>Nombre: GestinfDiferenciaModel.</b>	
<b>Tipo de clase: Modelo.</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad</b>	
<b>Nombre.</b>	<b>Descripción.</b>
GestInfDiferenciaModel()	Constructor de la clase.
getDiferencia(\$datos)	Responsabilidad encargada de buscar y cargar todos los Informes de Diferencias y mostrarlos al usuario para su posterior gestión.
cargarDiferencia(\$iddocumento)	Responsabilidades encargadas de cargar los datos del Informe de Diferencias a crear.
adicionarInfDif(\$datos) modificarInfDif(\$iddocumento,\$datos) eliminarInfDif(\$iddocumento,\$alias)	Responsabilidades encargadas de la gestión de los Informes de Diferencias.
diferenciaProducto(\$idmovimiento,\$idcalidad,\$cantidad,\$cantidadcalidad,\$motivo,\$idcalidadanterior)	Responsabilidad encargada de insertar las diferencias por concepto de calidad o cantidad a un producto.
eliminarDiferenciaProducto(\$idmovimiento,\$idcalidad)	Responsabilidad encargada de eliminar las diferencias a un producto.

preciImporteDiferencia(\$idmovimiento,\$id dualidadMonetaria,\$idmoneda,\$preciodifer encia,\$importediferencia)	Responsabilidad encargada de insertar las diferencias por conceptos de precio e importe y a un producto.
cargarCalidad()	Responsabilidad encargada de cargar los datos del nomenclador de calidad.
buscarProductosDoc(\$iddocumento,\$luga res ,\$codigo, \$nropieza, \$desc, \$start, \$limit,\$idcategoriaproducto,\$existencia,\$id grupo,\$tipo)	Responsabilidad encargada de buscar y cargar los productos que tiene un Informe de Diferencias.
cargarProdRecep(\$iddocumento,\$lugares, \$codigo, \$nropieza, \$desc,\$start, \$limit)	Responsabilidades encargadas de buscar y cargar los productos de un Informe de Recepción asociado a un Informe de Diferencias.
adicionarDeRecep(\$iddocumento,\$idprod ucto)	Responsabilidad encargada de adicionar productos del Informe de Recepción al Informe de Diferencias.
eliminarMovimiento(\$iddocumento,\$idmov imiento)	Responsabilidad encargada de eliminar productos del Informe de Diferencias.
aprobar(\$iddocumento)	Responsabilidad encargada de aprobar el Informe de Diferencias.

Tabla 5 Clase utilizada en la solución (GestinfndiferenciaModel)

➤ Gestionar productos del Informe de Recepción

<b>Nombre: GestproductoModel.</b>	
<b>Tipo de clase: Modelo</b>	
<b>Atributo</b>	<b>Tipo</b>
\$documentos	DatDocumento
Movimiento	DatMovimiento
<b>Para cada responsabilidad</b>	
<b>Nombre.</b>	<b>Descripción.</b>

GestproductoModel()	Constructor de la clase.
buscarProductosDoc(\$iddocumento,\$lugares,\$codigo,\$nropieza,\$desc,\$start,\$limit,\$idcategoriaproducto,\$existencia,\$idgrupo,\$tipo)	Responsabilidad encargada de buscar y cargar los productos que tiene un informe.
adicionarDedatProd(\$iddocumento,\$productos) modificarprod (\$iddocumento,\$datos,\$idestructuraop) eliminarProd(\$iddocumento,\$idproducto,\$datos)	Responsabilidades encargadas de gestionar los productos de un informe.
cargarGridMonedas(\$datos)	Responsabilidad encargada de cargar los datos del grid de monedas.
modificarCuentaRecepcion(\$idproducto,\$datos)	Responsabilidad encargada de modificar el atributo cuenta de un producto.
modifDual(\$idmovimiento,\$idmoneda,\$precio,\$cantidad, \$iddualidadmonetaria,\$importealternativo, \$iddescuentorecarga)	Responsabilidad encargada de modificar los datos de la dualidad monetaria de un informe.
eliminarDM(\$iddualidadmonetaria)	Responsabilidad encargada de eliminar la dualidad monetaria de un informe.

**Tabla 6 Clase utilizada en la solución (GestproductoModel)**

### 2.6. Descripción de los algoritmos no triviales a implementar

Una de las funcionalidades fundamentales que presenta en proceso de recepción de productos en un almacén es la contabilización del documento que avala la entrada de los mismos, contabilizar(\$idestructuracomun,\$alias,\$iddocumento) es un método del módulo que surgió de la necesidad de dar cumplimiento a dicha funcionalidad.



### 2.6.1. Análisis de complejidad de los mismos

Para conocer la complejidad del algoritmo es necesario calcular la complejidad ciclomática, 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 en la figura 9 el código con sus instrucciones enmarcadas.

```

public function contabilizar($idestructuracomun,$alias,$iddocumento)
{
    $val = ''; 1
    $version = 1; 1
    $mov = $this->pIntegrator->movimientos->Existencia($iddocumento); 1
    if($mov[0]){ 2
        $sint = 0; 3
        foreach($movimientos as $k => $prod) 3
        {
            $producto = $this->pIntegrator->productos->obtenerprodID($prod->idproducto); 4
            if(($prod->cantidad == 0 || $prod->preciocontable == 0 || $prod->cuenta == 0 || $producto->idcategoriaproducto == 0)){5-6-7-8
                $array[$sint] = $prod->idmovimiento; 9
                $sint++; 9
            }
        } 10
        if(!$array){ 11
            $dat = $this->pIntegrator->movimientos->Movcondiferencia($iddocumento); 12
            if($dat){ 13
                $dif['idinformerecepcion'] = $iddocumento; 14
                $dif['idestructuracomun'] = $idestructuracomun; 14
                $dif['alias'] = $alias; 14
                $dif['observaciones'] = ''; 14
                $datos = $this->pIntegrator->documentos->AdicionarinfDif($dif); 14
                if($datos) 15
                    $val = 5; 16
                else
                    $val = 6; 17
            }
            foreach($mov as $k => $prod){ 18
                $result = $this->pIntegrator->productos->obtenerProducto()->obtenerProducto($prod->idproducto); 19
                $movexist = $this->pIntegrator->movimientos->BuscarMov($iddocumento,$prod->idproducto); 19
                $idmovimiento = $movexist->idmovimiento; 19
                $importetotalprod = $result[0]['importetotal']; 19
                $preciopromedio = $result[0]['preciopromedio']; 19
                $idproducto = $prod->idproducto; 19
                $cantidad = $prod->cantidad; 19
                $existencia = $result[0]['existencia']; 19
                $signo = '+'; 19
                $arr['idproducto'] = $idproducto; 19
                $arr['cantidad'] = 0; 19
                $this->pIntegrator->productos->actualizarcantdisp($cantidad,$arr,'+'); 19
                $cronologia = $this->pIntegrator->documentos->DevolverCronologia(); 19
                $datos = $this->pIntegrator->movimientos->UpdateMovContabilizar($idmovimiento,$cantidad,$signo,
                    $importetotalprod,$preciopromedio,$existencia,$cronologia); 19
                $productos[$k] = array ('idproducto' => $prod->idproducto,'importetotal' =>$datos->importetotal,
                    'preciopromedio' =>$datos->preciopromedio, 'cantidad' => $datos->cantidad); 19
            } 20
            $this->pIntegrator->productos->contabilizarProd($productos, '+'); 21
            $this->pIntegrator->documentos->ContabilizarRecepcion($iddocumento,$version); 21
        }
        else {
            $datt = json_encode($res); 22
            return $datt ; 22
        }
    }
    else{
        return 1; 23
    }
}
if($val !=5 && $val !=6) 24-25
return 2; 26
else
return $val; 27
} 28

```

Figura 9 Algoritmo no trivial a implementar contabilizar (\$idestructuracomun, \$alias, \$iddocumento).

Después de este paso, es necesario representar el grafo de flujo asociado, que para realizar su construcción se debe tener en cuenta la siguiente notación para cada una de las instrucciones de código tal y como se muestra en las figuras 10 y 11.

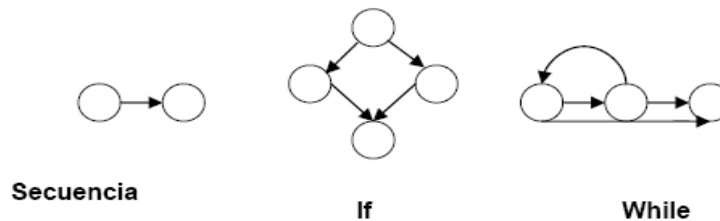


Figura 10 Notación de grafos de flujo para las instrucciones: Secuenciales, If, While.

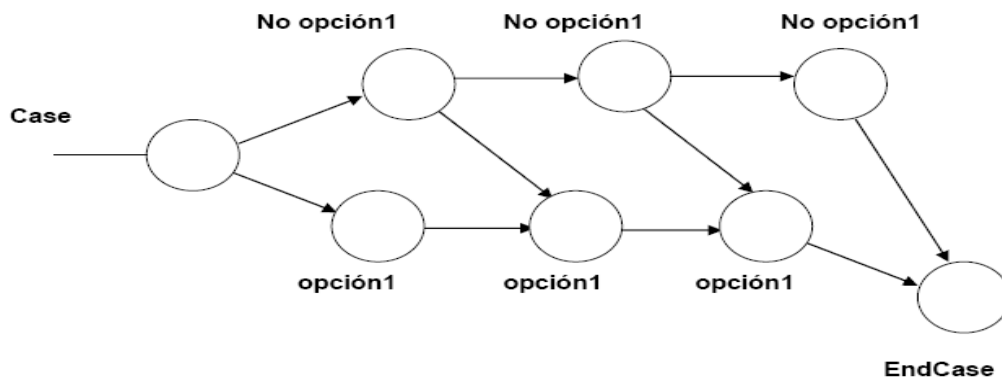


Figura 11 Notación de grafos de flujo para la instrucción Case.

En un grafo de flujo se representan distintos componentes como es el caso de:

**Nodos:** son los círculos representados en el grafo de flujo, el cual simboliza 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.

Teniendo en cuenta todo esto el grafo de flujo asociado al algoritmo no trivial quedaría de la forma en que se muestra en la figura 12:

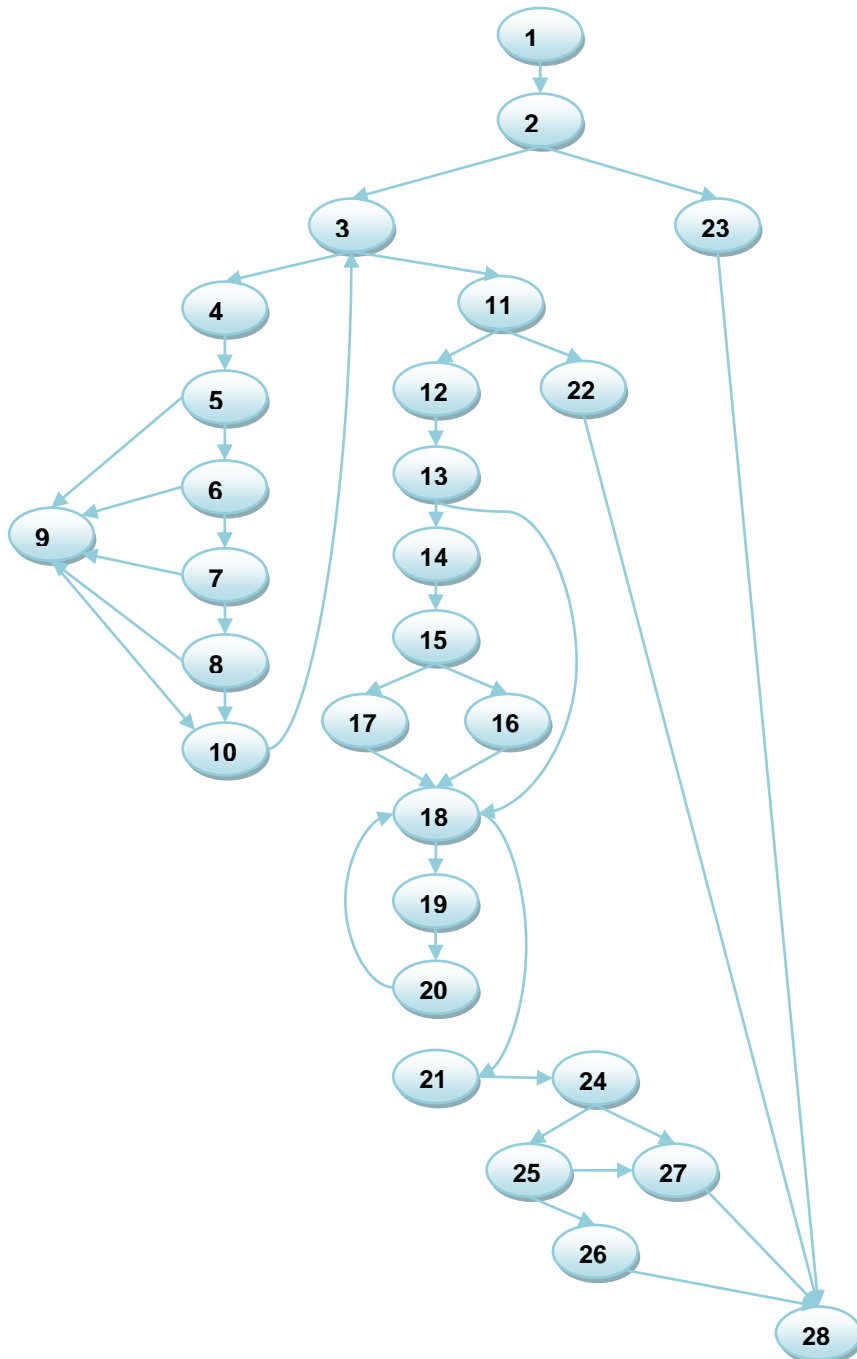


Figura 12 Grafo de flujo asociado al algoritmo contabilizar ().

Seguidamente a la construcción del grafo de flujo se procede a efectuar el cálculo de la complejidad ciclomática del código, el cálculo es necesario efectuarlo mediante tres vías o fórmulas que para concluir que el cálculo fue correcto es necesario que por las tres vías el resultado sea el mismo, las fórmulas para calcular son las siguientes:

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

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

$$V(G) = (39-28) + 2$$

$$V(G) = 13$$

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

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

$$V(G) = 12 + 1$$

$$V(G) = 13$$

3.  $V(G) = R$

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

$$V(G) = 13$$

Realizado el cálculo por las 3 vías necesarias se llega a la conclusión que el algoritmo presentado anteriormente tiene un complejidad ciclomática de 13 que da la visión de que existen a lo sumo once caminos lógicos por donde recorrer el algoritmo.

### 2.7. Conclusiones parciales

Con el desarrollo de este capítulo se conocieron los procesos objeto de automatización, lo que proporcionó una temprana idea de la complejidad de la solución. Se explicaron los componentes que fueron reutilizados para el desarrollo de la aplicación, teniendo en cuenta la arquitectura del sistema, para un mejor entendimiento de la solución propuesta al mostrarse la integración entre los mismos. La descripción de clases importantes permitió tener una visión de la implementación realizada. De forma general se logró implementar los requerimientos definidos por los analistas para los procesos de recepción, pertenecientes al subsistema de Inventario del Sistema Integral de Gestión CedruX.

# Capítulo 3

## Validación de la solución propuesta

### 3.1. Introducción

El desarrollo del software implica una serie de actividades de producción en las que las posibilidades de que aparezcan fallos humanos son muy grandes. Los errores pueden presentarse debido a especificaciones erróneas e imperfectas de los requisitos, uso indebido de las estructuras de datos, errores al integrar módulos, entre otras causas. Dado a la imposibilidad humana de trabajar y comunicarse de forma perfecta, el desarrollo del software ha de ir acompañado de una actividad que garantice la calidad.

Las pruebas son una actividad en la cual un sistema o componente es ejecutado bajo condiciones o requerimientos especificados, los resultados son observados y registrados, y una evaluación es hecha de algún aspecto del sistema o componente. En este capítulo se realiza la validación a la solución propuesta, con el objetivo de 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

Las pruebas de software son el conjunto de técnicas que permiten determinar la calidad de un producto. Estas se integran dentro de las diferentes fases del Ciclo del Vida del software dentro de la Ingeniería. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir que errores presenta. La calidad de un sistema informático es algo subjetivo que depende del contexto y del objeto que se pretenda conseguir. Para determinar dicho nivel de calidad se deben efectuar medidas o pruebas que permitan comprobar el grado de cumplimiento con respecto a las especificaciones iniciales del sistema.

### 3.2.1. Objetivos de las Pruebas

El objetivo de las pruebas de un programa es el de detectar todo posible mal funcionamiento, un error puede ser costoso de reparar mientras más se avanza en las etapas del ciclo de vida del software. Para minimizar estos riesgos se crean baterías de pruebas que serán de mayor calidad en la medida de cuantos menos errores queden por descubrir tras haberla pasado y viceversa, si un programa aún tiene muchos fallos tras haberla superado, se dirá que esta es de poca calidad. Si se pudiera probar un programa con todos los posibles datos de entrada, se tendría una batería de pruebas perfecta, pues no hay lugar para las sorpresas. Lamentablemente, casi nunca es posible probar con todos los casos. En consecuencia se necesita un criterio para elegir qué casos se prueban.

### 3.2.2. Pruebas de unidad

Las pruebas de unidades o “test de unidades” son pruebas llevadas a cabo por los implementadores sobre las unidades mínimas desarrolladas por ellos, estas unidades pueden ser clases, métodos, propiedades, componentes, etc. Se prueban separadas unas de otras y básicamente se hacen durante la implementación del software.

Para realizar estas pruebas es necesario establecer una serie de reglas que sirven como objetivos.

- La prueba es un proceso de ejecución de un programa con la intención de descubrir errores.
- Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Las pruebas de unidades son orientadas casi siempre a las pruebas de caja blanca aunque para realizarlas es necesario probar el flujo de datos desde la interfaz del componente. Si los datos no se comportan correctamente al ser introducidos en la aplicación, todas las demás pruebas no tienen sentido. Estas pruebas son las denominadas pruebas de caja blanca y son aplicadas a componentes representados en el modelo de implementación para verificar que los flujos de control y de datos están cubiertos, y que estos funcionen como se espera.

### 3.2.3. Técnicas de Diseño de Pruebas

Existen tres enfoques principales para el diseño de casos:

1. El enfoque estructural o de caja blanca: que se basan en un minucioso examen de los detalles procedimentales del código a evaluar, por lo que es necesario conocer la lógica del programa.
2. El enfoque funcional o de caja negra: que 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.
3. El enfoque aleatorio: consiste en utilizar modelos (en muchas ocasiones estadísticos) que representen las posibles entradas al programa para crear a partir de ellos los casos de prueba caja negra.

La Figura 13 representa gráficamente la filosofía de las pruebas de caja blanca y caja negra. En este caso profundizaremos en las pruebas de caja blanca teniendo en cuenta que fue seleccionada esta técnica para validar la implementación del módulo.

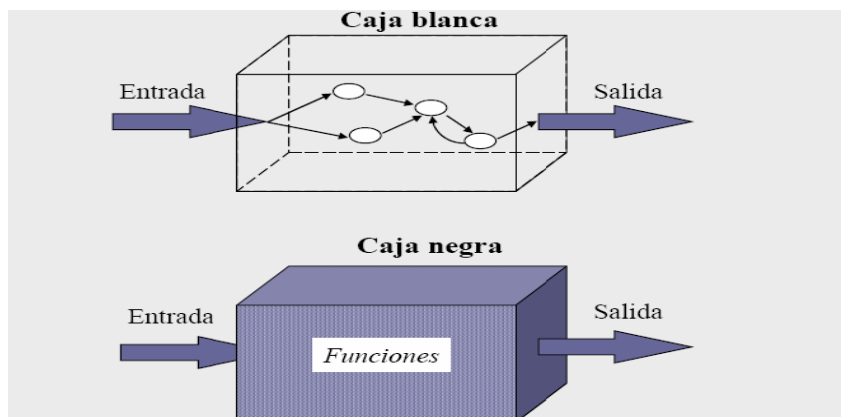


Figura 13 Representación de pruebas de Caja Blanca y Caja Negra.

### 3.2.4. Prueba de Caja Blanca o Estructurales

A este tipo de técnicas se le conoce también como Técnicas de Caja Transparente o de Cristal. Este método se centra en cómo diseñar los casos de prueba atendiendo al comportamiento interno y la estructura del programa examinando su lógica interna sin considerar los aspectos de rendimiento. El



objetivo de la técnica es diseñar casos de prueba para que se ejecuten, al menos una vez, todas las sentencias del programa, y todas las condiciones tanto en su vertiente verdadera como falsa.

En resumen, mediante la prueba de la caja blanca se puede obtener casos de prueba que:

- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

A continuación se citan algunas de las técnicas de prueba de Caja Blanca:

- Prueba de Condición
- Prueba de Flujo de Datos
- Prueba de Bucles
- Prueba del Camino Básico

Esta última 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 se construye el Grafo de Flujo asociado y se calcula su complejidad ciclomática.

Los pasos que se siguen para aplicar esta técnica son:

1. A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.
2. Se calcula la complejidad ciclomática del grafo.
3. Se determina un conjunto básico de caminos independientes.
4. Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Los casos de prueba derivados del conjunto básico garantizan que durante la prueba se ejecuta por lo menos una vez cada sentencia del programa.

### 3.2.5. Aplicación de pruebas de caja blanca

Para realizar las pruebas de caja blanca es necesario calcular antes la complejidad ciclomática del algoritmo o fragmento de código a analizar de la misma manera que se hizo en el apartado correspondiente al capítulo 2 epígrafe 2.6.1. A continuación en la figura 14 se enumeran las sentencias de código del procedimiento realizado sobre el método modifDual() el cual se encarga de modificar atributos referentes al trabajo con precios en los productos.

```

public function modifDual($idmovimiento,$idmoneda,$precio,$cantidad,$iddualidadmonetaria,$importealternativo,$iddescuentorecarga)
{
    $est=$this->global->Estructura; 1
    $identidad=$est->idestructura; 1
    $trabajadual=$this->integrator->parametros->ExisteMonedaAlertnativa($identidad); 1

    if($iddualidadmonetaria){ 2
        $id = $this->pIntegrator->movimientos->Buscar_MDM($iddualidadmonetaria); 3

        if ($id){ 4
            if($trabajadual) 5
                $dualidadmonetaria=1; 6
            else
                $dualidadmonetaria=0; 7
            $this->pIntegrator->movimientos->ModificarAlternativo_MDM($idmovimiento,$iddualidadmonetaria,
                $idmoneda,$precio,$importealternativo,$dualidadmonetaria,$cantidad,$iddescuentorecarga); 8
            return 1; 8
        }
        return 2; 9
    }
    else {
        $this->pIntegrator->movimientos->Insertar_MDM($idmovimiento,$cantidad,0,$idmoneda);10
        return 1; 10
    }
} 11
    
```

Figura 14 Representación del algoritmo modifDual ().

Seguidamente se construye el grafo de flujo asociado al código anterior quedando como se muestra en la figura 15.

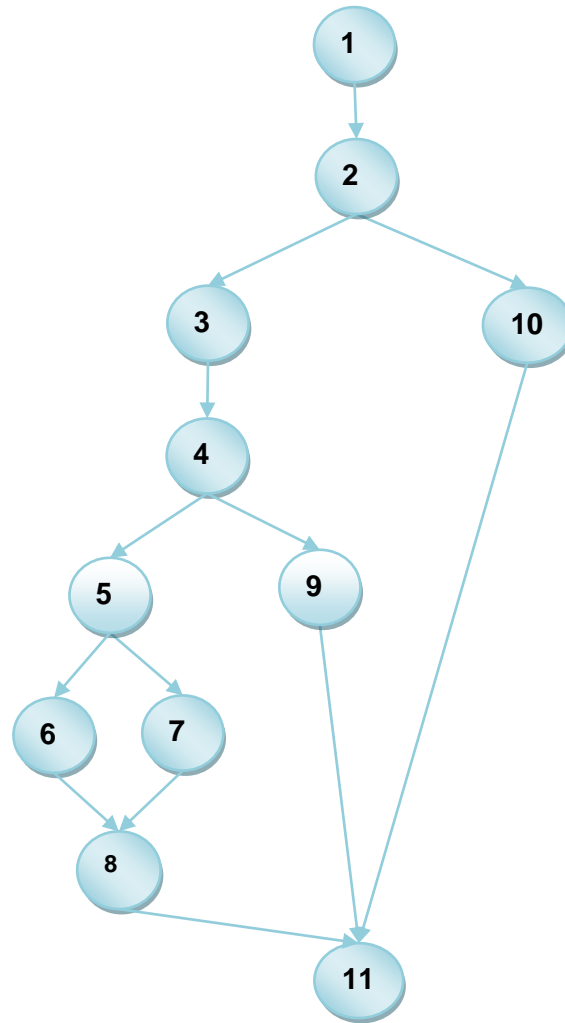


Figura 15 Grafo de flujo asociado al algoritmo modifDual ().

Luego de haber construido el grafo, se realiza el cálculo de la complejidad ciclomática, mediante las tres fórmulas descritas en el capítulo 2, epígrafe 2.6.1 “Análisis de complejidad del algoritmo”, las cuales tienen que arrojar el mismo resultado para asegurar que el cálculo de la complejidad es correcto.

**Fórmulas para calcular complejidad ciclomática.**

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

Aplicando estas fórmulas al grafo de flujo de la figura 15 se obtienen los siguientes resultados:

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

$$V(G) = 3 + 1$$

$$V(G) = 4$$

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

$$V(G) = 13 - 11 + 2$$

$$V(G) = 4$$

$$V(G) = R$$

$$V(G) = 4$$

El cálculo efectuado mediante las tres fórmulas ha dado el mismo valor, por lo que se puede plantear que la complejidad ciclomática del código es de 4, lo que significa que existen 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. En estas representaciones se subrayan los elementos de cada camino que los hacen independientes a los demás.

**Camino 1:** 1-2-10-11.

**Camino 2:** 1-2-3-4-9-11.

**Camino 3:** 1-2-3-4-5-6-8-11.

**Camino 4:** 1-2-3-4-5-7-8-11.

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 realizarlos 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 resultado que se espera que devuelva el procedimiento.

### Caso de prueba para el camino básico 1

#### Camino 1: 1-2-10-11.

##### Descripción:

- Los datos de entrada cumplirán con los siguientes requisitos: El parámetro iddualidadmonetaria será nulo.

##### Condición de ejecución:

- El idmovimiento será 1,00000000000001E15
- El idmoneda será 10001
- La cantidad será 30
- El iddualidadmonetaria tendrá valor nulo

##### Entrada:

- \$idmovimiento = 1,00000000000001E15
- \$idmoneda = 90000000008
- \$precio = nulo
- \$cantidad = 30
- \$iddualidadmonetaria = nulo
- \$importealternativo = nulo
- \$iddescuentorecarga = nulo

##### Resultados esperados:

- Se espera que al movimiento se le asocie una nueva moneda satisfactoriamente

### Caso de prueba para el camino básico 2

#### Camino 2: 1-2-3-4-9-11.

##### Descripción:

- Los datos de entrada cumplirán con los siguientes requisitos: El iddualidadmonetaria introducido no existirá en la base de datos

##### Condición de ejecución:

- El idmovimiento será 1,00000000000001E15
- El idmoneda será 90000000008
- El precio será 10.52
- La cantidad será 30
- El iddualidadmonetaria tendrá valor 100
- El importealternativo tendrá un valor de 5.63
- El iddescuentorecarga será 12.88

##### Entrada:

- \$idmovimiento = 1,00000000000001E15
- \$idmoneda = 90000000008
- \$precio = 10.52
- \$cantidad = 30
- \$iddualidadmonetaria = 100
- \$importealternativo = 10.55
- \$iddescuentorecarga = 45.50

##### Resultados esperados:

- Se espera que se lance un mensaje con el siguiente texto: “No se pudo realizar la modificación. El movimiento asociado a la moneda no existe.”

### Caso de prueba para el camino básico 3

#### Camino 3: 1-2-3-4-5-6-8-11.

##### Descripción:

- Los datos de entrada cumplirán con los siguientes requisitos: El iddualidadmonetaria introducido existirá en la base de datos y la entidad trabajará con más de una moneda.

Condición de ejecución:

- El idmovimiento será 1,00000000000001E15
- El idmoneda será 90000000032
- El precio será 60.99
- La cantidad será 30
- El iddualidadmonetaria tendrá valor 100000000057
- El importealternativo tendrá un valor de 25.05
- El iddescuentorecarga será 54.99

Entrada:

- \$idmovimiento = 1,00000000000001E15
- \$idmoneda = 90000000032
- \$precio = 60.99
- \$cantidad = 30
- \$iddualidadmonetaria = 100000000057
- \$importealternativo = 25.05
- \$iddescuentorecarga = 54.99

Resultados esperados:

- Se realizará la modificación sin problema alguno, con los valores de los parámetros.

#### **Caso de prueba para el camino básico 4**

**Camino 4:** 1-2-3-4-5-7-8-11.

Descripción:

- Los datos de entrada cumplirán con los siguientes requisitos: El iddualidadmonetaria introducido existirá en la base de datos y la entidad no trabajará con más de una moneda.

Condición de ejecución:

- El idmovimiento será 1,00000000000001E15
- El idmoneda será 90000000008
- El precio será \$10.52
- La cantidad será 30
- El iddualidadmonetaria tendrá valor 100000000058

- El importe alternativo tendrá un valor de 5.63
- El id descuento recargo será 18.6

Entrada:

- \$idmovimiento = 1,00000000000001E15
- \$idmoneda = 90000000008
- \$precio = 10.55
- \$cantidad = 35
- \$id dualidad monetaria = 100000000058
- \$importe alternativo = 20.80
- \$id descuento recargo = 66.56

Resultados esperados:

- Se realizará la modificación sin problema alguno, con los valores de los parámetros.

Luego de aplicar los distintos casos de pruebas, se pudo comprobar que el flujo de trabajo de la función está correcto ya que cumple con las condiciones necesarias que se habían planteado.

### 3.3. Evaluación de la solución obtenida mediante métricas

Numerosos son los puntos de vista relacionados con la calidad de un software. Desde las disímiles metodologías existentes hasta las normas de calidad vigentes en la actualidad. En este apartado no se realizará un análisis de los distintos temas de calidad existentes sino que se desarrollará una evaluación de la solución obtenida al implementar los artefactos entregados por los analistas correspondientes al módulo de recepción de productos en un almacén.

Primeramente se realizó un mapeo entre las características funcionales asociadas al proceso de recepción y las clases implementadas para dar solución a los requisitos funcionales (tabla 7). Garantizándose de esta manera el chequeo de cumplimiento del atributo interno de calidad que plantea la norma ISO 9126 (37) la Funcionalidad, que consiste en la capacidad del software de proveer las funciones que cumplen con las necesidades implícitas y explícitas cuando el mismo es utilizado bajo ciertas condiciones (37) Demostrándose que las características funcionales más importantes definidas para ambos módulos han sido cubiertas en la solución obtenida.



<b>Proceso o funcionalidad</b>	<b>Requerimiento funcional asociado.</b>	<b>Clase de la implementación que lo cubre.</b>	<b>Método de la clase que lo cubre.</b>
Gestionar informe de recepción	Crear informe de recepción	<b>GestinfrecepcionController</b>	<b>guardarinfrecepAction()</b>
	Buscar informes de recepción	<b>GestinfrecepcionController</b>	<b>cargardocumentosrepcionAction()</b>
	Búsqueda avanzada de informe de recepción	<b>GestinfrecepcionController</b>	<b>cargardocumentosrepcionAction()</b>
	Modificar informe de recepción	<b>GestinfrecepcionController</b>	<b>modificardocumentoAction()</b>
	Eliminar informe de recepción	<b>GestinfrecepcionController</b>	<b>eliminarinfrecepAction()</b>
	Cancelar informe de recepción	<b>GestinfrecepcionController</b>	<b>cancelarrepcionAction()</b> <b>cancelarrepcionmovAction()</b>
	Contabilizar informe de recepción	<b>GestinfrecepcionController</b>	<b>contabilizarAction()</b>
Gestionar productos al informe de recepción	Adicionar productos al informe de recepción	<b>GestinfrecepcionController</b>	<b>adicionardatprodAction()</b> <b>modifdualAction()</b> <b>modcategoriaprodrecepAction()</b> <b>nsertarcantidadAction()</b> <b>eliminaradmAction()</b> <b>adicionarcuentaAction()</b>
	Buscar productos del informe de recepción	<b>GestinfrecepcionController</b>	<b>buscarproductosdocAction()</b>
	Modificar productos del informe de recepción	<b>GestinfrecepcionController</b>	<b>modificarprodrepcionAction()</b>
	Eliminar productos del	<b>GestinfrecepcionController</b>	<b>eliminarprodAction()</b>

	informe de recepción		
	Registrar diferencias a los productos en el informe de recepción	<b>GestindiferenciaController</b>	<b>productodiferenciaAction()</b>
	Modificar diferencias a los productos en el informe de recepción	<b>GestindiferenciaController</b>	<b>productodiferenciaAction()</b>
	Eliminar diferencias a los productos en el informe de recepción	<b>GestindiferenciaController</b>	<b>elimdiferenciaAction()</b>
	Crear tarjeta de estiba		<b>adicionardenomprodAction()</b>
Gestionar informes de diferencias	Crear informe de diferencias	<b>GestindiferenciaController</b>	<b>guardarindiferenciaAction()</b>
	Eliminar informe de diferencias	<b>GestindiferenciaController</b>	<b>eliminarindiferenciaAction()</b>
	Modificar informe de diferencias	<b>GestindiferenciaController</b>	<b>modificarindiferenciaAction()</b>
	Buscar un informe de diferencia	<b>GestindiferenciaController</b>	<b>cargarindiferenciaAction()</b>
	Búsqueda avanzada de un informe de diferencias	<b>GestindiferenciaController</b>	<b>cargarindiferenciaAction()</b>
	Aprobar informe de diferencias	<b>GestindiferenciaController</b>	<b>aprobarAction()</b>
Gestionar productos al informe de diferencias	Adicionar productos al informe de diferencias	<b>GestindiferenciaController</b>	<b>adicionarprodderecepAction()</b> <b>adicionardatprodAction()</b>

	Buscar productos del informe de diferencias	<b>GestindiferenciaController</b>	<b>cargarprodrepcionAction()</b>
	Registrar diferencias a los productos del informe de diferencias	<b>GestindiferenciaController</b>	<b>precioimportediferenciaAction()</b>
	Modificar diferencias a los productos del informe de diferencias	<b>GestindiferenciaController</b>	<b>precioimportediferenciaAction()</b>
	Modificar productos del informe de diferencias	<b>GestindiferenciaController</b>	<b>modifdualAction()</b>
	Eliminar productos del informe de diferencias	<b>GestindiferenciaController</b>	<b>eliminarproductoAction()</b>
Gestionar lotes de los productos	Registrar lotes del producto	<b>LoteController</b>	<b>adicionarloteAction()</b>
	Eliminar lotes del producto	<b>LoteController</b>	<b>eliminarloteAction()</b>
	Modificar lotes del producto	<b>LoteController</b>	<b>adicionarloteAction()</b>
Gestionar números de serie de los equipos	Adicionar número de serie del equipo	<b>GestnrosSerieController</b>	<b>addnrosSerieAction()</b>
	Modificar número de serie del equipo	<b>GestnrosSerieController</b>	<b>modnrosSerieAction()</b>
	Eliminar número de serie del equipo	<b>GestnrosSerieController</b>	<b>eliminarnroSerieAction()</b>

Ubicar productos en las áreas de almacenamiento	Asignar ubicación al producto	<b>GestestructubicacionController</b>	<b>adicionarAreaAction()</b>
	Modificar ubicación	<b>GestestructubicacionController</b>	<b>modificarAreaAction()</b>
	Eliminar ubicación	<b>GestestructubicacionController</b>	<b>eliminarAreaAction()</b>

**Tabla 7 Requisitos asociados al módulo de recepción.**

Inspirados en los estudios que realizara Pressman sobre la calidad del diseño orientado a objeto se crearon un grupo de métricas básicas, referenciadas en (37). Las mismas son otro aspecto muy importante a la hora de realizar la evaluación de la solución obtenida teniendo en cuenta que estas cubren los principales atributos de calidad que un software debe requerir.

Atributos de calidad que se abarcan:

**1. Responsabilidad:**

Su esencia radica fundamentalmente en la responsabilidad asignada a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.

**2. Complejidad de implementación:**

Su esencia radica fundamentalmente en el grado de dificultad que tiene implementar un diseño de clases determinado.

**3. Reutilización:**

Su esencia radica fundamentalmente en el grado de reutilización presente en una clase o estructura de clase, dentro de un diseño de software.

**4. Acoplamiento:**

Su esencia radica fundamentalmente en el grado de dependencia o interconexión de una clase o estructura de clase, con otras, está muy ligada a la característica de Reutilización.

**5. Complejidad del mantenimiento:**

Su esencia radica fundamentalmente 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.

**6. Cantidad de pruebas:**

Su esencia radica fundamentalmente en el número o el grado de esfuerzo para realizar las pruebas de calidad (Unidad) del producto (Componente, módulo, clase, conjunto de clases, etc.) implementado.

**7. Nivel de Cohesión:**

Su esencia radica fundamentalmente en el grado de especialización de las clases concebidas para modelar un dominio o concepto específico.

**8. Abstracción del diseño:**

Su esencia radica fundamentalmente en capacidad de modelar lo más cercano posible a la realidad un concepto o dominio determinado.

Luego de concebir estas métricas, las cuales se muestran a continuación y que servirán de instrumento para realizar una evaluación crítica de lo implementado se realizó un análisis de las tablas de resultados obtenidos en la evaluación de los instrumentos de medición.

**3.3.1. Métricas aplicadas**

**1- Tamaño operacional de clase (TOC):** Está dado por el número de métodos asignados a una clase. La cual afecta un grupo de atributos como se muestra en la tabla 8.

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.

**Tabla 8 Tamaño operacional de clase (TOC).**

**2- Relaciones entre clases (RC):** Está dado por el número de relaciones de uso de una clase con otras. La cual afecta un grupo de atributos de calidad como se muestra en la tabla 9.

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.

**Tabla 9 Relaciones entre clases (RC).**

**3- Profundidad de Herencia (PH):** Está dado por la profundidad en la herencia de las clases heredades de un nodo padre. La cual afecta un grupo de atributos de calidad como se muestra en la tabla 10.

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

**Tabla 10 Profundidad de Herencia (PH).**

**4- Número de Operaciones Redefinidas para una clase hija (NOR):** Está dado por la cantidad de operaciones redefinidas en cada clase hija. La cual afecta un grupo de atributos de calidad como se muestra en la tabla 11.

Atributo que afecta	Modo en que lo afecta
Abstracción del diseño	Un aumento del NOR implica que se ha logrado una buena definición

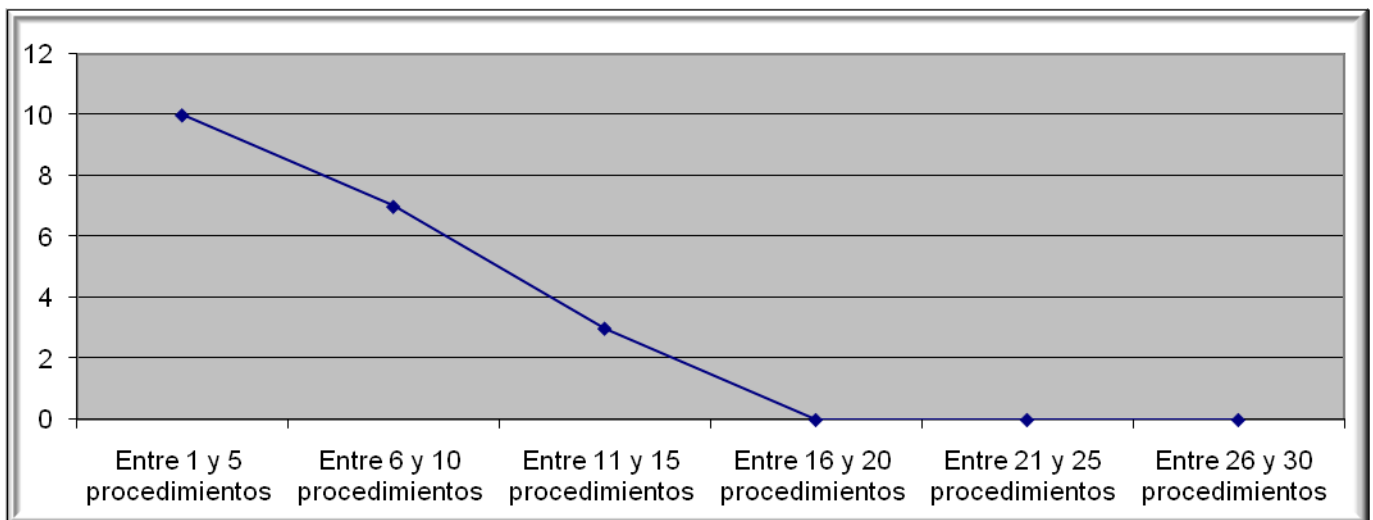
	de la abstracción del diseño de clases.
Cantidad de pruebas	Un aumento del NOR implica un aumento de la cantidad de pruebas de unidad necesarias para probar una clase.
Complejidad del mantenimiento	Un aumento del NOR implica un aumento de la complejidad del mantenimiento de la clase.

**Tabla 11 Número de Operaciones Redefinidas para una clase hija (NOR).**

### 3.3.2. Resultados de los instrumentos de evaluación de las métricas

#### 3.3.2.1. Resultados del instrumento de evaluación de la métrica Tamaño operacional de clase (TOC)

En las figuras 16 y 17 se muestran los resultados obtenidos por el instrumento de medición agrupados en intervalos definidos. También se mostraran en las figuras 18, 19 y 20, todos los resultados obtenidos en el instrumento de medición TOC para los atributos de calidad responsabilidad, complejidad y reutilización. Para más información ver instrumentos y tabla de resultados en ([Anexo 11 Instrumento de medición de la métrica Tamaño operacional de clase \(TOC\)](#)).



**Figura 16 Representación de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.**

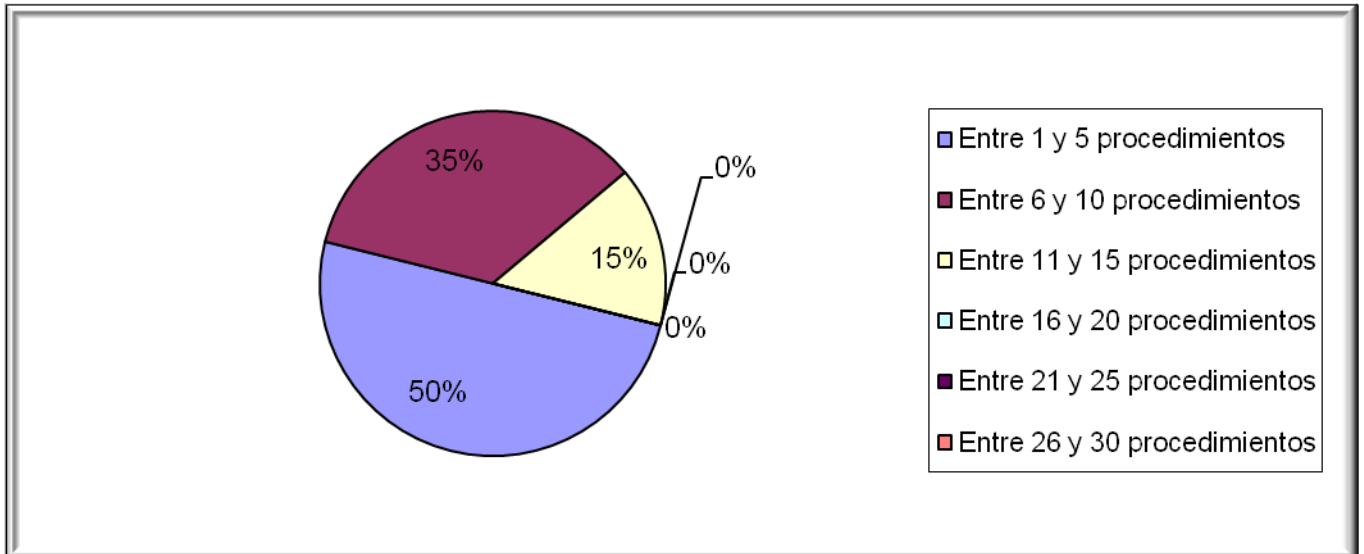


Figura 17 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.

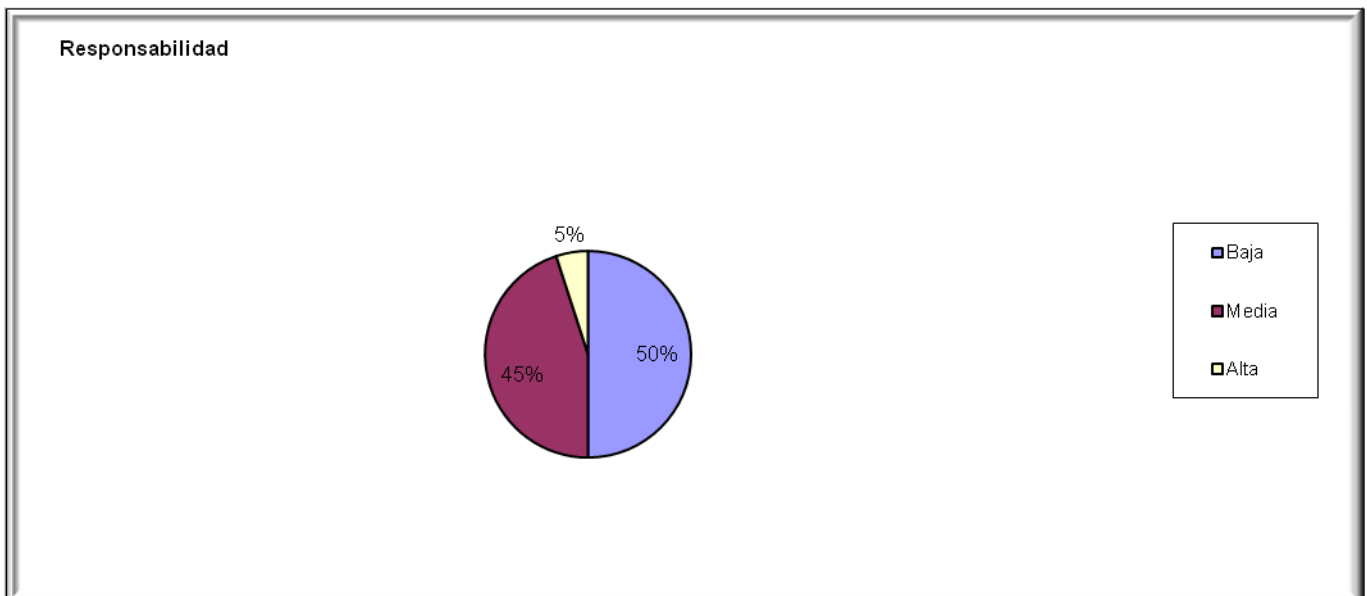
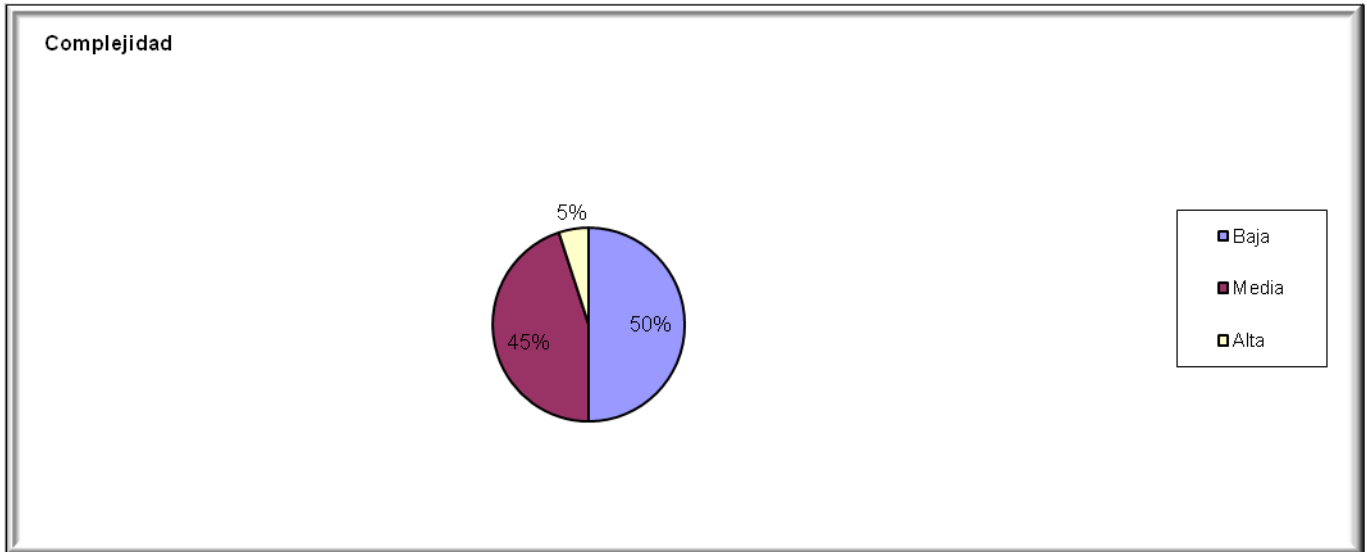
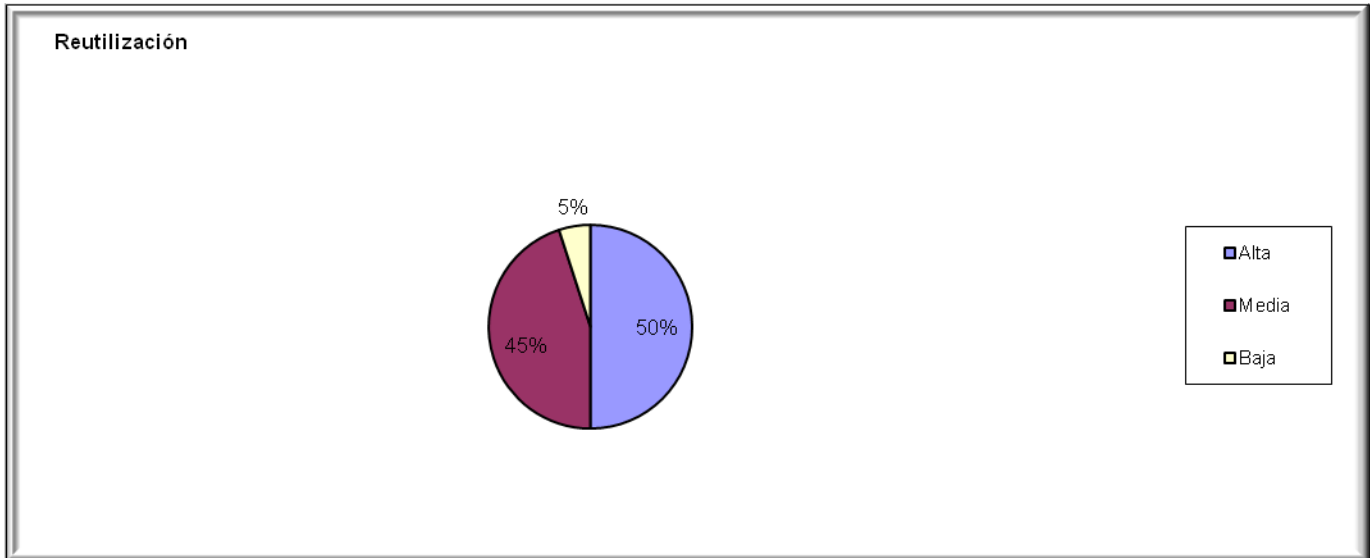


Figura 18 Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo Responsabilidad.





**Figura 19** Representación de la incidencia de los resultados de la evaluación de la métrica TOC en el atributo **Complejidad de Implementación.**



**Figura 20** 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 pudo observar que la implementación de las clases necesarias para dar solución a la problemática tiene una calidad aceptable, teniendo en cuenta que el 55 % de las clases utilizadas, incluidas en la solución posee menos cantidad de operaciones que la mitad del valor máximo registrado en las mediciones. Además el 100% de las clases poseen evaluaciones positivas en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización).

### 3.3.2.2. Resultados del instrumento de evaluación de la métrica Relaciones entre Clases (RC)

En las figuras 21 y 22 se muestran los resultados obtenidos por el instrumento de medición agrupados por la tendencia de los valores. También se mostraran en las figuras 23, 24, 25 y 26, todos los resultados obtenidos en el instrumento de medición RC para los atributos de calidad acoplamiento, complejidad cantidad de pruebas y reutilización.

Para más información ver instrumentos y tabla de resultados en ([Anexo 12 Instrumento de medición de la métrica Relaciones entre clases \(RC\)](#)).

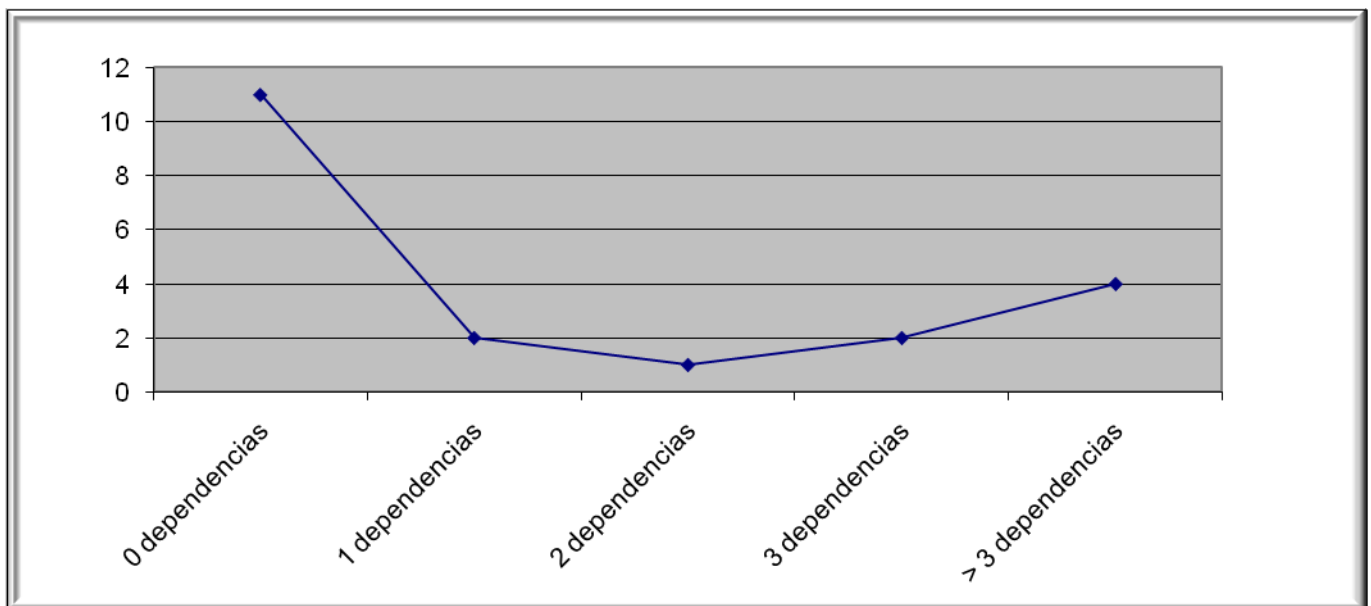


Figura 21 Representación de los resultados de la evaluación de la métrica RC agrupados por la tendencia de los valores.

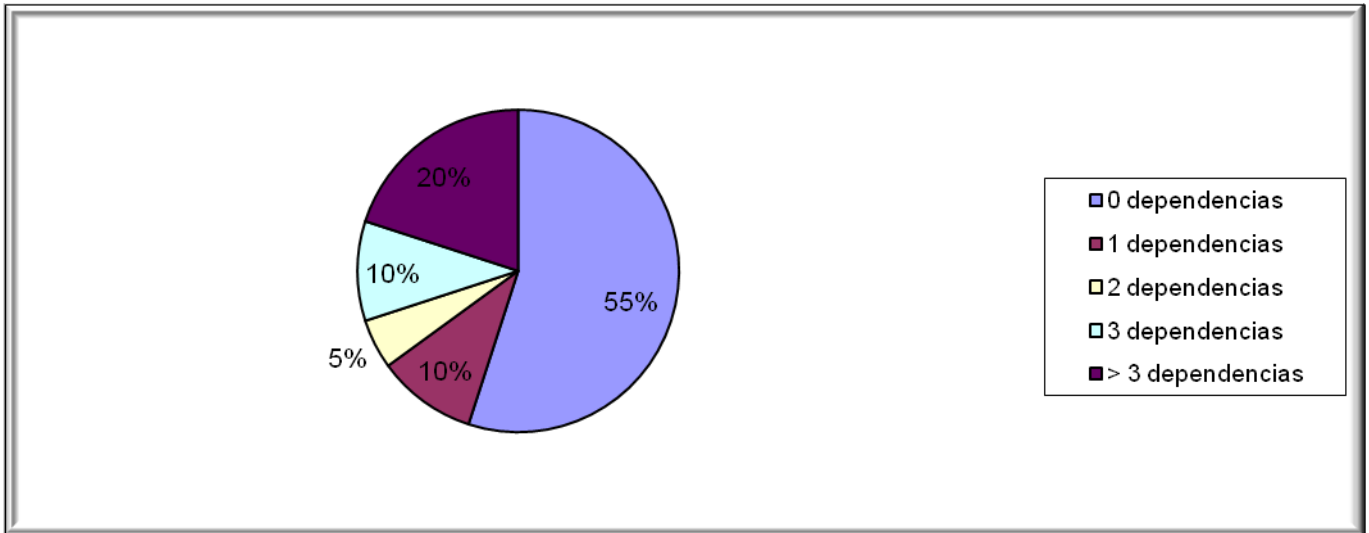


Figura 22 Representación en % de los resultados obtenidos en el instrumento agrupados en los intervalos definidos.

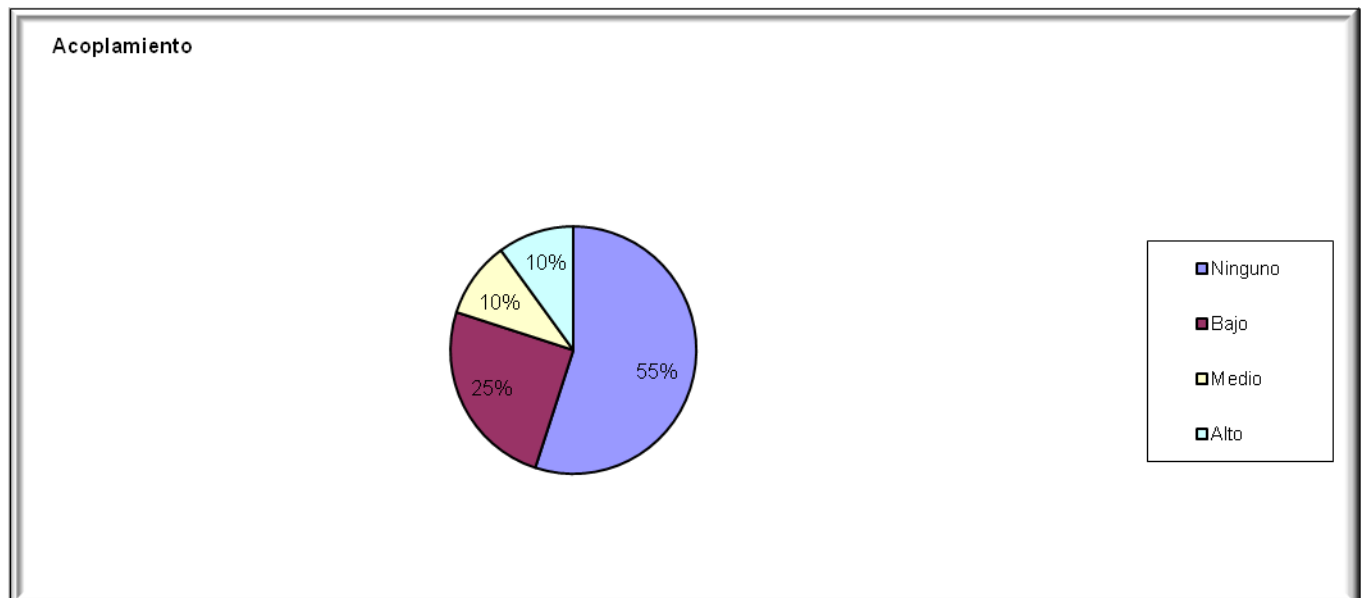


Figura 23 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Acoplamiento.

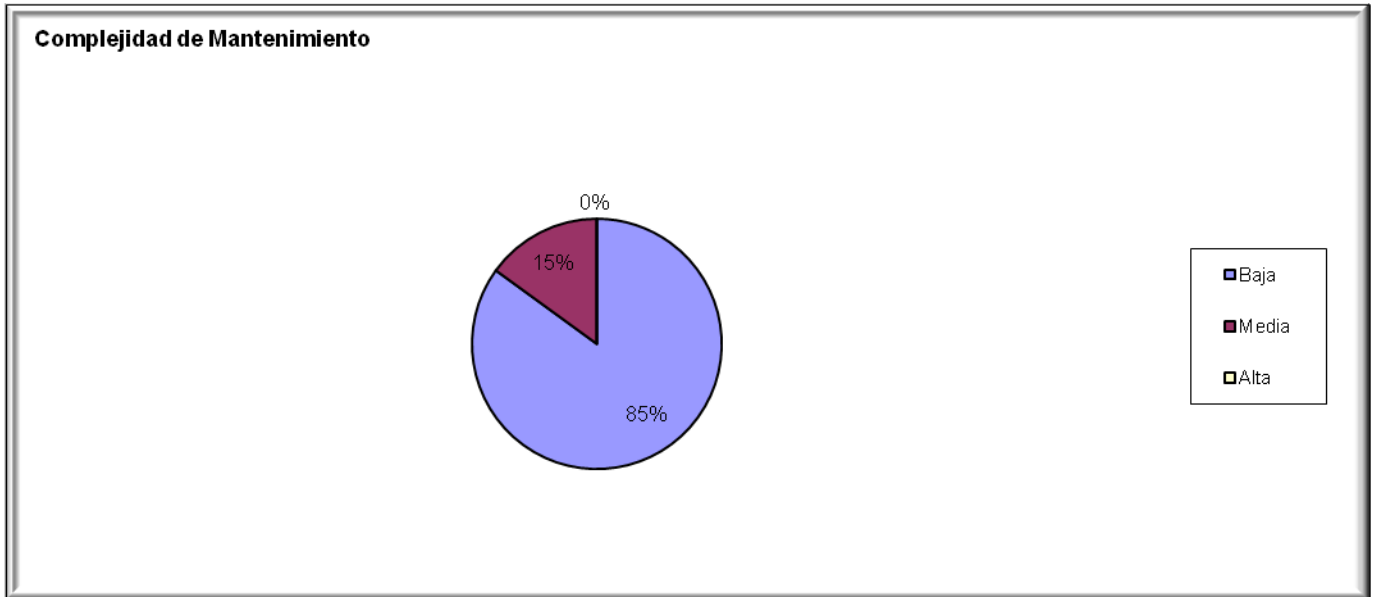


Figura 24 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Complejidad de Mantenimiento.

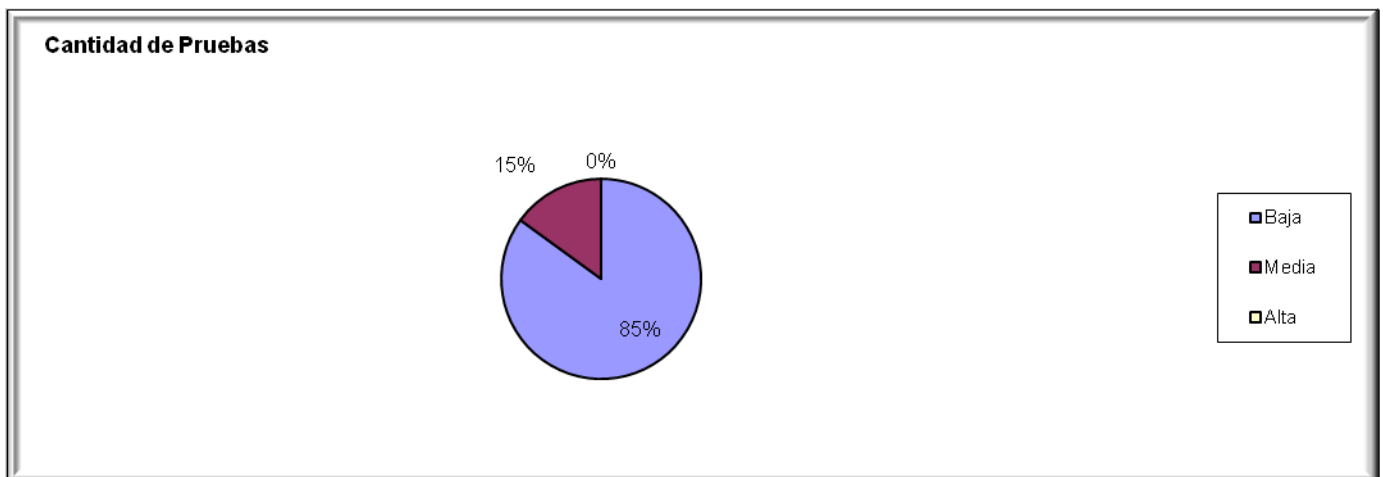
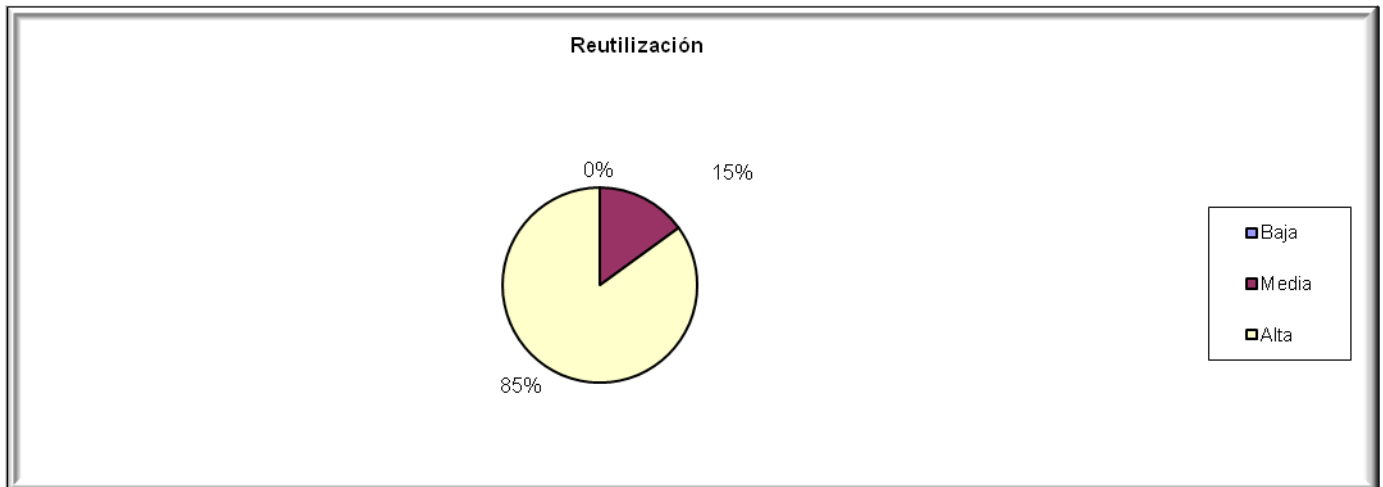


Figura 25 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Cantidad de Pruebas.



**Figura 26 Representación de la incidencia de los resultados de la evaluación de la métrica RC en el atributo Reutilización.**

Al hacer un análisis de los resultados obtenidos al evaluar la métrica Relaciones entre clases (RC) con el instrumento de evaluación, se pudo observar que la implementación de las clases necesarias para dar solución a la problemática tiene una calidad aceptable teniendo en cuenta que de las clases implementadas el 70 % de las clases desarrolladas posee 3 o menos dependencias de otras clases. Además de que el 55 % de las clases no posee acoplamiento alguno y el 70% posee índices de calidad aceptables respecto al mismo atributo de calidad. A la vez que los atributos de calidad Complejidad de Mantenimiento, Cantidad de Pruebas y Reutilización se comportan satisfactoriamente en un 85 % de las clases.

### **3.3.2.3. Resultados del instrumento de evaluación de la métrica Profundidad de Herencia (PH)**

En la figura 27 se representan en por ciento el resultado obtenido por el instrumento de medición agrupados por nivel. También se mostraran en las figuras 28, 29 y 30 todos los resultados obtenidos en el instrumento de medición PH para los atributos de calidad complejidad de mantenimiento, de implementación y reutilización.

Para más información ver instrumentos y tabla de resultados en ([Anexo 13 Instrumento de medición de la métrica Profundidad de Herencia \(PH\)](#)).

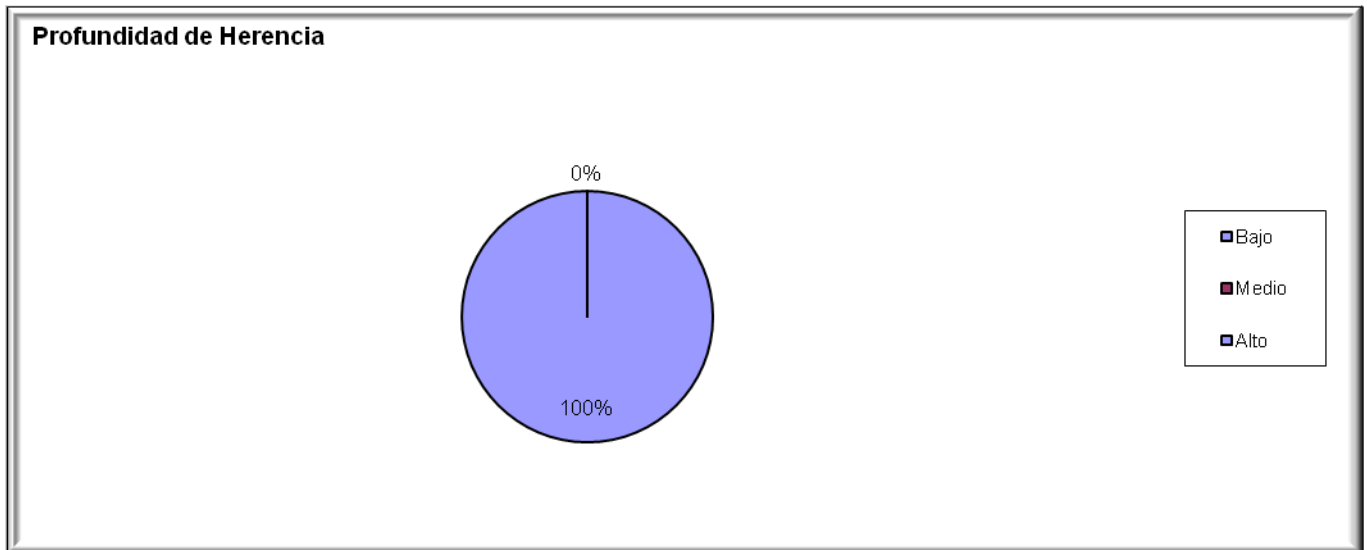


Figura 27 Representación en % de los resultados obtenidos en el instrumento agrupados por nivel.

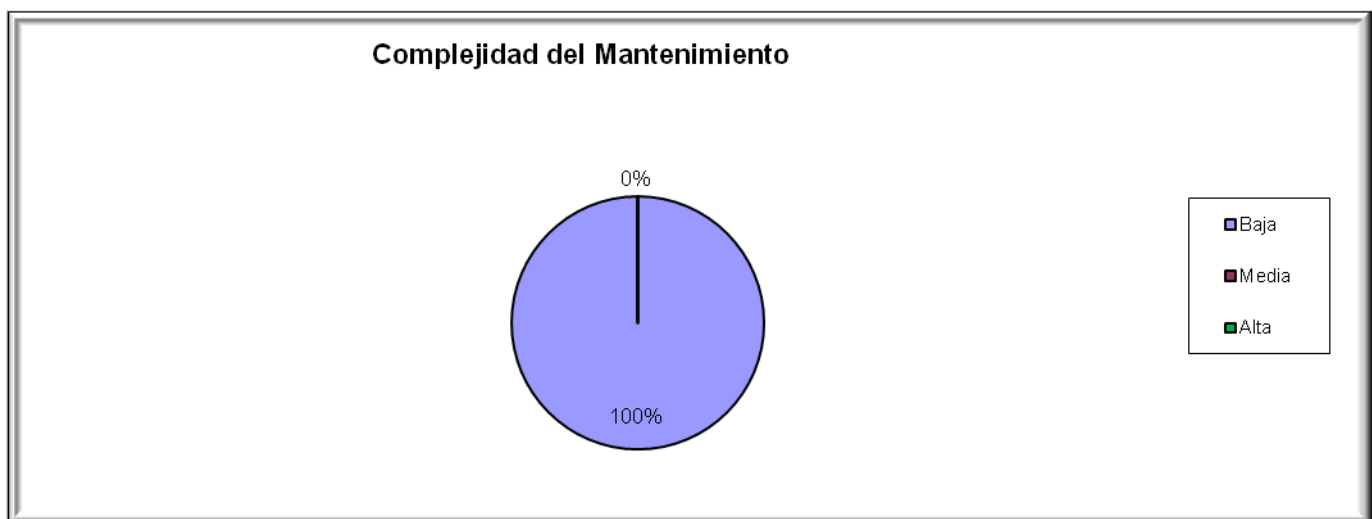
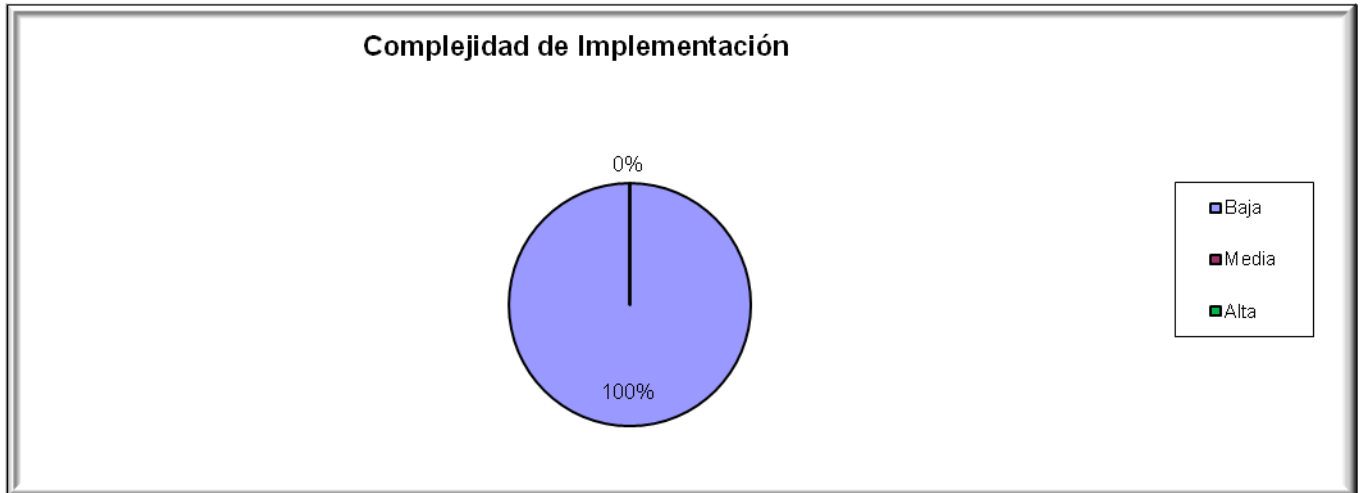
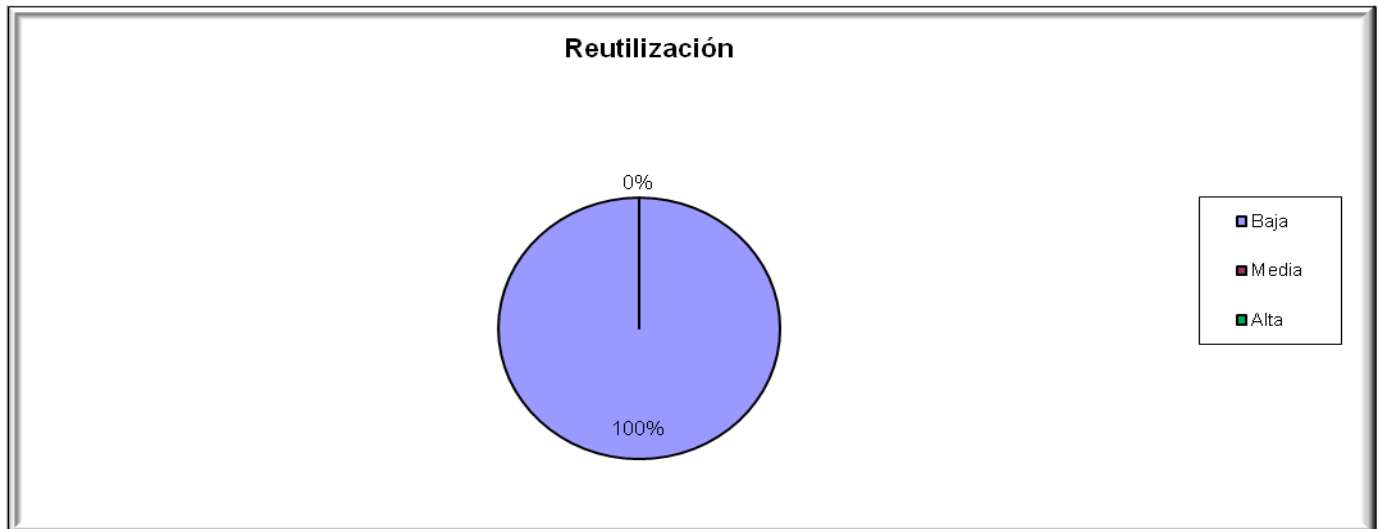


Figura 28 Representación de la incidencia de los resultados de la evaluación de la métrica PH en el atributo Complejidad de Mantenimiento.



**Figura 29** Representación de la incidencia de los resultados de la evaluación de la métrica PH en el atributo **Complejidad de Implementación.**



**Figura 30** Representación de la incidencia de los resultados de la evaluación de la métrica PH en el atributo **Reutilización.**

Al hacer un análisis de los resultados obtenidos al evaluar la métrica Profundidad de Herencia (PH) con el instrumento de evaluación, se pudo observar que la implementación de las clases necesarias para dar solución a la problemática tiene una calidad aceptable teniendo en cuenta que la profundidad de herencia

presentes en los subsistemas desarrollados es siempre de 1, un nivel menos que el valor de nivel mínimo posible.

Analizando los atributos de calidad Complejidad de Mantenimiento, se puede observar que los mismos poseen buenos valores producto de que en la implementación se cuenta el nivel dos, como máximo nivel de profundidad en la herencia. No obstante el atributo reutilización no posee indicadores alentadores debido al nivel tan bajo de profundidad que posee la herencia.

### 3.3.2.4. Resultados del instrumento de evaluación de la métrica Número de Operaciones Redefinidas (NOR)

En la figura 31 se representan en porcentaje el resultado obtenido por el instrumento de medición para la redefinición de métodos agrupados en los valores existentes. También se mostraran en las figuras 32, 33 y 34 todos los resultados obtenidos en el instrumento de medición PH para los atributos de calidad complejidad de mantenimiento, cantidad de pruebas y violación de la abstracción representada por la superclase.

Para más información ver instrumentos y tabla de resultados en ([Anexo 15 Instrumento de medición de la métrica Número de Operaciones Redefinidas \(NOR\)](#)).

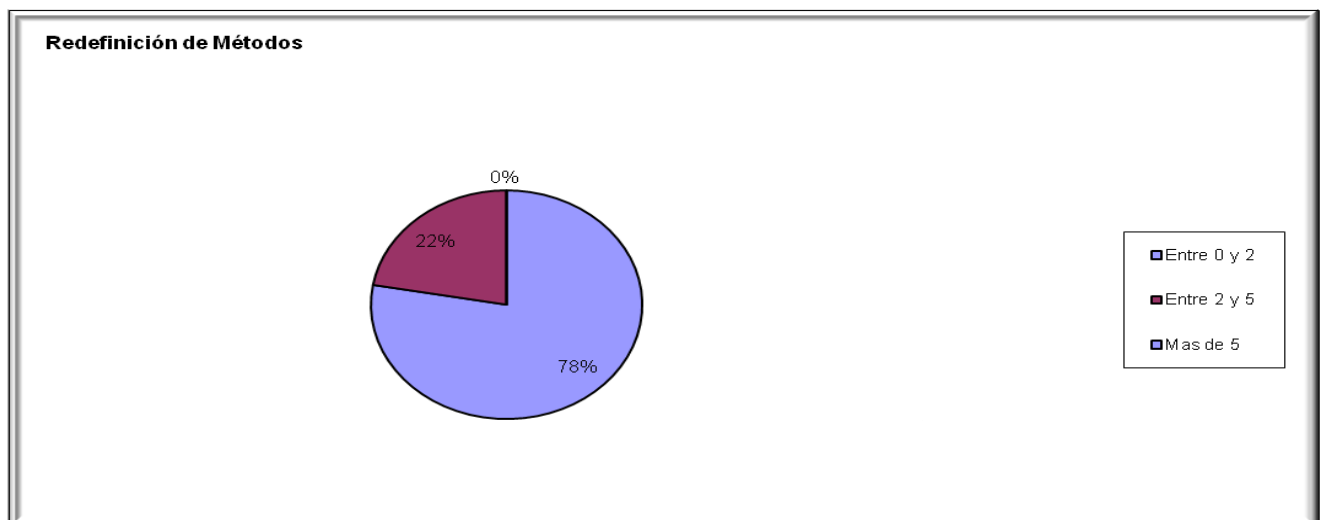


Figura 31 Representación en % de los resultados obtenidos en el instrumento agrupados en los valores existentes.



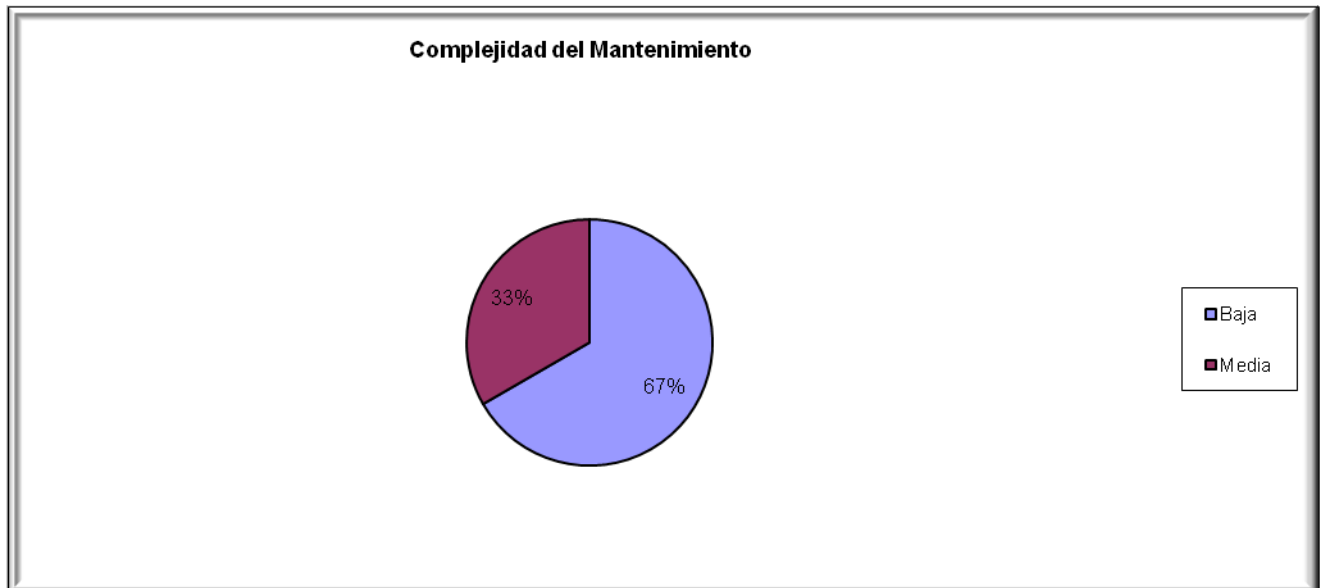


Figura 32 Representación de la incidencia de los resultados de la evaluación de la métrica NOR en el atributo Complejidad del Mantenimiento.

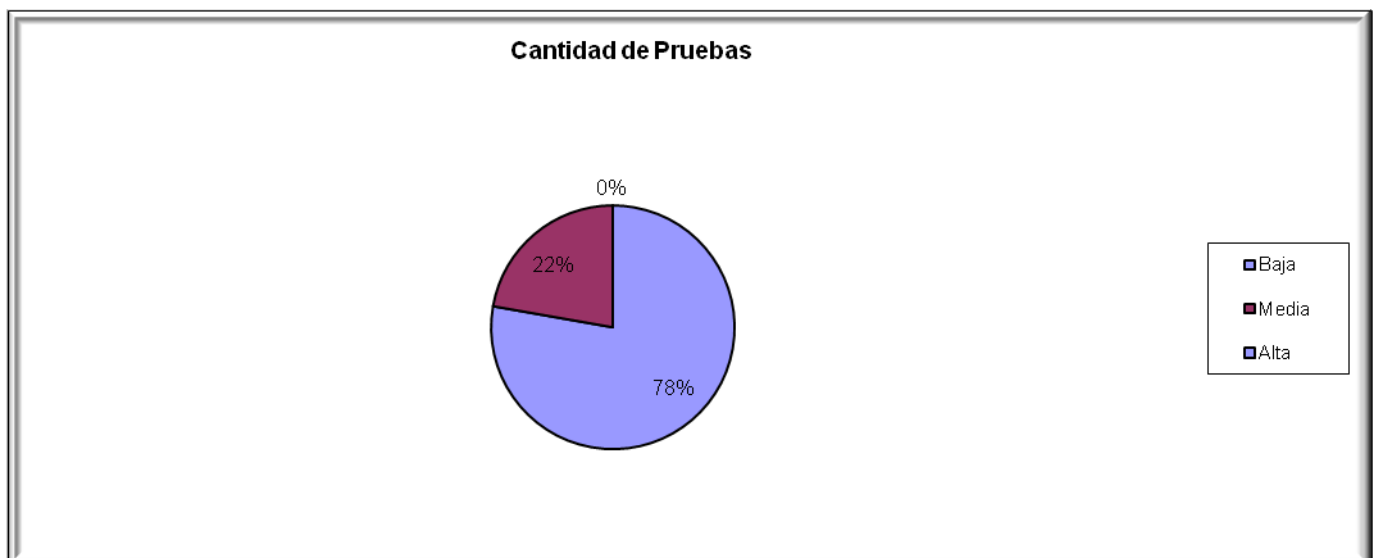


Figura 33 Representación de la incidencia de los resultados de la evaluación de la métrica NOR en el atributo Cantidad de Pruebas.



**Figura 34 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.**

Haciendo un análisis de los resultados obtenidos en la evaluación del instrumento de medición de la métrica NOR, se puede concluir que la implementación realizada tiene una calidad aceptable teniendo en cuenta que solo el 33 % de las clases incluidas en el análisis (solo relacionadas con la herencia) redefinen algún método heredado. 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.

### **3.3.3. Matriz de cubrimiento de los parámetros de calidad evaluados con las métricas propuestas.**

La matriz de cubrimiento o matriz de inferencia de los indicadores de calidad es un resumen de los resultados conseguidos al aplicar las métricas referidas en el epígrafe anterior. Esta matriz es una representación estructurada de los atributos de calidad y métricas utilizadas para evaluar la calidad de la solución obtenida como se muestra en la tabla 14. La misma permite conocer si el resultado obtenido de la relación atributo/métricas es positivo o negativo. Al llevar estos resultados a una escala numérica se le

asignará el valor 1 si los resultados son positivos, en caso de sean negativos el valor 0 y si no existe relación alguna tomará valor -1, tal y como se muestra en la tabla 12.

Calificativo	Valor
Positivo	1
Negativo	0
Nulo	-1

**Tabla 12 Clasificación de los parámetros de calidad según impacto en el diseño propuesto**

Una vez completado los datos de dicha relación se determina el promedio de los valores obtenidos de la relación atributo/métrica (solo se toman en consideración las que arrojan un resultado distinto de -1). Este valor representa el impacto que tiene cada atributo en la implementación de la solución determinando si su impacto fue bueno, regular o malo atendiendo al rango en que se encuentre dicho valor, tal y como se muestra en la tabla 13. Al desarrollar este procedimiento con los resultados que se obtuvieron una vez aplicadas las métricas en la implementación realizada se obtuvo la matriz de cubrimiento que se expone a continuación en la tabla 14.

Calificativo	Rango
Malo	$\leq 0.4$
Regular	$>0.4$ y $\leq 0.7$
Bueno	$>0.7$

**Tabla 13 Rangos para evaluar el impacto de los parámetros de calidad en la solución.**

Atributos de calidad evaluados	Métricas aplicadas al diseño de la solución propuesta				
	TOC	CR	PH	NOR	PROMEDIO
Complejidad Implementación	1	-1	1	-1	1
Reutilización	1	1	0	-1	0,67
Acoplamiento	-1	1	-1	-1	1

<b>Complejidad Mantenimiento</b>	1	1	1	1	1
<b>Cantidad Pruebas</b>	-1	1	-1	1	1
<b>Abstracción</b>	-1	-1	-1	1	1
<b>Calidad Diseño</b>	1	-1	-1	1	1

**Tabla 14 Matriz de cubrimiento para los parámetros de calidad evaluados con las métricas aplicadas a la solución.**

Como se evidencia en los datos visualizados en la matriz se obtuvieron resultados positivos en todos los atributos de calidad evaluados, exceptuando el atributo de la Reutilización, el cual al aplicar la métrica de PH se ve afectado por las características propias del sistema, donde las clases hijas desarrolladas poseen un nivel muy bajo de herencia, provocando a una pobre reutilización, pero propiciando a su vez excelentes valores en cuanto Complejidad de Mantenimiento y Complejidad de Implementación. Todo lo anteriormente expuesto permitió demostrar que las características funcionales más importantes definidas fueron cubiertas en la solución obtenida

### 3.4. Conclusiones parciales

En el desarrollo de este capítulo se analizaron diferentes aspectos como el significado de las pruebas de software y sus objetivos. Se realizó una descripción de las pruebas de unidad, dentro de los cuales se analizó y aplicó el tipo de prueba Caja Blanca, verificando sentencias del programa, y todas las condiciones tanto en su vertiente verdadera como falsa, obteniéndose resultados satisfactorios, que permitieron validar el correcto funcionamiento de las funciones implementadas. Además se elaboraron instrumentos inspirados en métricas para calidad del diseño que permitieron evaluar distintos criterios de calidad arrojando como resultado, valores positivos de su comportamiento, ratificando que la implementación realizada puede catalogarse de aceptable.

## Conclusiones generales

El presente trabajo recoge todo el proceso de implementación del Módulo de Recepción del Subsistema de Inventario del Sistema Cedrux quedando como colofón del mismo:

El análisis realizado de los procesos de Recepción de productos de un almacén y los sistemas que existen actualmente para su control así como la especificación de las tecnologías y herramientas que se utilizaron para el desarrollo de la solución.

La descripción de los aspectos más significativos de la solución como son, la reutilización de componentes, el cálculo de la complejidad ciclomática de algoritmos no triviales así como las principales clases y estándares de codificación utilizados en la implementación de los requerimientos especificados para los procesos de Recepción.

La validación de la solución propuesta mediante el diseño y aplicación de las pruebas de caja blanca, permitiendo validar el correcto funcionamiento de las funciones implementadas.

La evaluación de la implementación a partir de la aplicación de métricas que permitieron analizar el comportamiento de los atributos de calidad de reutilización, facilidad de mantenimiento, complejidad del diseño, complejidad de implementación, cohesión, acoplamiento y cantidad de pruebas; lo cual arrojó resultados positivos que permiten catalogar de aceptable la implementación realizada.

Por todo lo anteriormente expresado se puede concluir que se cumplió con el objetivo general propuesto: Implementar los procesos de recepción de productos en almacenamiento para las entidades empresariales y unidades presupuestadas del país.

## Recomendaciones

Existen una serie de factores que permitirán la mejora de los procesos de Recepción y que deben tenerse en consideración para versiones posteriores de este producto, entre ellos están:

- Corregir las no conformidades detectadas en las pruebas pilotos con el objetivo de refinar la solución.
- Ampliar las funcionalidades del módulo con los nuevos requerimientos que surjan por necesidades del cliente.
- Incorporar la posibilidad de recibir documentos digitales (factura, conduce, transferencia, vale de devolución, modelo de movimiento de activo fijo) que originan el informe de recepción, para posibilitar la creación automática de la entrada de los medios materiales soportados por los documentos de origen.

## Referencias bibliográficas

1. **Torres Saquipova, Dina Yaksilik y de la Rosa Hernández, Carlos.** *Análisis y Diseño de los módulos Inventario y Administración del proyecto ERP Cubano.* [Digital] Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2008.
2. **Ríos, José Carlos del Toro y Brito, Henry Raúl González.** *Documento Visión. Proyecto ERP Cuba.* [Digital] La Habana : Universidad de las Ciencias Informáticas, 2008.
3. **SAP España.** SAP: número uno en software ERP. [En línea] 2007. [Citado el: 17 de Mayo de 2009.] <http://www.sap.com/spain/solutions/business-suite/erp/index.epx..>
4. **software, Exact.** Software de gestión empresarial Exact software. [En línea] Exact software. [http://www.exactsoftware.es/index.igw?item=1635&lang=es\\_ES&site=1](http://www.exactsoftware.es/index.igw?item=1635&lang=es_ES&site=1).
5. Características OpenBravo. [En línea] 2009. [Citado el: 15 de Abril de 2009.] <http://www.openbravo.com/es/product/erp/features/..>
6. Rodas XXI Sistema Integral Económico- Administrativo . [En línea] CITMATEL , 2002. [Citado el: 9 de Enero de 2009.] <http://www.rodasxxi.cu/inventario.php>.
7. Manual del sistema SISCONT5. Módulo Inventario. [En línea] Tecnomática (MINBAS), 1985. [Citado el: 23 de Marzo de 2009.] <http://siscont.tm.minbas.cu/Docs/Documentos/Manuales/Inventario%20SISCONT5.pdf>.
8. **Autores, Colectivo de.** *Introducción al proceso de desarrollo de software.* [Digital] Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2007.
9. **Producción, Equipo de.** *Modelo de Desarrollo orientado a componentes del proyecto ERP -CUBA.* [Digital] Ciudad de la Habana : Centro de desarrollo de soluciones empresariales (CEDRUX), 2008.
10. **Rivas, Lornel A., y otros.** Herramientas de Desarrollo de Software: Hacia la construcción de una Ontología. 2007.
11. Visual Paradigm for UML. [En línea] [http://www.freedownloadmanager.org/es/downloads/Paradigma\\_Visual\\_para\\_UML\\_\(M%C3%8D\)\\_14720\\_p/](http://www.freedownloadmanager.org/es/downloads/Paradigma_Visual_para_UML_(M%C3%8D)_14720_p/).

12. Zend Studio v5.5.1.281 Enterprise Edition. [En línea] [Citado el: 6 de Abril de 2008.] <http://utilzone.net/index.php?s=5144f7c47c6584c98cd6857128950893&showtopic=1655&pid=39239&st=0&#entry39239>.
13. About Notepad++. [En línea] [Citado el: 5 de Abril de 2009.] <http://notepad-plus.sourceforge.net/es/site.htm>.
14. PgAdmin III. [En línea] [Citado el: 6 de Abril de 2008.] [http://www.guia-ubuntu.org/index.php?title=PgAdmin\\_III](http://www.guia-ubuntu.org/index.php?title=PgAdmin_III).
15. **SQLManager**. [En línea] 2006. [Citado el: 8 de marzo de 2009.] <http://sqlmanager.net/>.
16. Guia Ubuntu. [En línea] [Citado el: 12 de abril de 2009.] <http://www.guia-ubuntu.org/index.php?title=Subversion>.
17. **Küng, Stefan, Onken, Lübbe y Large, Simon**. TortoiseSVN. Un cliente de Subversion para Windows. [En línea] 9 de mayo de 2009. [Citado el: 6 de junio de 2009.] <http://hivelocity.dl.sourceforge.net/sourceforge/tortoisesvn/TortoiseSVN-1.6.2-es.pdf>.
18. **Orallo., Enrique Hernández**. El Lenguaje Unificado de Modelado (UML). [En línea] [http://www.acta.es/articulos\\_mf/26067.pdf](http://www.acta.es/articulos_mf/26067.pdf).
19. **Fernández, Víctor Fresno**. Representación Autocontenida de Documentos HTML: una propuesta basada en Combinaciones Heurísticas de Criterios. [En línea] [http://www.escet.urjc.es/~vfresno/phd\\_sp.html](http://www.escet.urjc.es/~vfresno/phd_sp.html).
20. ¿Que es Javascript? [En línea] [Citado el: 4 de Abril de 2009.] <http://www.desarrolloweb.com/articulos/25.php>.
21. Manual de PHP. [En línea] [Citado el: 15 de enero de 2009.] <http://www.php-es.com/preface.html>.
22. ¿Qué es PHP? [En línea] <http://www.desarrolloweb.com/articulos/392.php>.
23. ¿Qué es Postgres? [En línea] [Citado el: 19 de Marzo de 2008.] <http://www.postgresql.com.ar/html/informacion.php?opcion=queespostgres> (19/3/2008)..
24. **Perera Morales, José Raúl**. *Arquitectura de Software para Sistema Gestion de Inventario*. . [Digital] Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2007.
25. Licencia de Ext JS and Ext GWT. [En línea] <http://extjs.com/products/license.php>.
26. **Pressman, Roger S**. *Ingeniería del Software. Un enfoque práctico*. .
27. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James**. *El Proceso Unificado de Desarrollo de Software*. . s.l. : Pearson Educación S.A.



28. Herramientas Web. El modelo cliente – servidor. [En línea] [Citado el: 5 de Abril de 2009.] <http://neo.lcc.uma.es/evirtual/cdd/tutorial/aplicacion/cliente-servidor.html>.
29. El servidor web. [En línea] <http://www.infor.uva.es/~jvegas/cursos/buendia/pordocente/node20.html>.
30. Configuración de apache. [En línea] <http://www.mailxmail.com/curso-configuracion-apache/que-es-apache-que-alternativas-hay-mercado>.
31. Una Introducción a APACHE. [En línea] [http://linux.ciberaula.com/articulo/linux\\_apache\\_intro/](http://linux.ciberaula.com/articulo/linux_apache_intro/).
32. Mozilla Firefox. [En línea] [Citado el: 6 de Abril de 2009.] <http://www.javinet.com.ar/firefox.htm>.
33. ¿Qué es AJAX? [En línea] <http://www.programacionweb.net/articulos/articulo/?num=317>.
34. ¿Qué es AJAX? [En línea] <http://www.elguruprogramador.com.ar/articulos/que-es-ajax.htm>.
35. Qué es XML. [En línea] <http://www.desarrolloweb.com/articulos/449.php>.
36. **ERP, Línea de Arquitectura.** *Normas y estándares de Codificación del ERP.* [Digital] Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2008.
37. **Yzquierdo Herrera, Raykenler y Lazo Ochoa, René.** *modelo de diseño del sistema HyperWeb.* [Digital] Ciudad de la Habana : Universidad de las Ciencias Informáticas, 2007.
38. **Precios, Ministra de Finanzas y.** *RESOLUCION No. 11-2007.* [Digital] Ciudad de la Habana : MINISTERIO DE FINANZAS Y PRECIOS, 2007.

## Glosario de términos

➤ **Almacén:**

Lugar en el que llevan a cabo los movimientos de productos; entradas, salidas e inventario de los mismos.

➤ **Factura:**

Es el documento que formaliza las ventas de productos y prestaciones de servicios que se efectúen, así como las entregas de productos en consignación o en depósito, que no son sujetas de la Factura Comercial. Se utiliza además, para formalizar las ventas de activos fijos tangibles y de productos para efectuar devoluciones. (38)

➤ **Informe de Recepción:**

Formaliza la recepción en el Almacén de los productos adquiridos de otras entidades. (38)

➤ **Informe de Reclamación:**

Es el documento que notifica al suministrador o al transportador las reclamaciones originadas por errores en precios y cálculos al primero y por averías, roturas o faltantes en ambos. (38)

➤ **Producto:**

Es cualquier objeto que puede ser ofrecido a un mercado que pueda satisfacer un deseo o una necesidad. Sin embargo, es mucho más que un objeto físico. Es un completo conjunto de beneficios o satisfacciones que los consumidores perciben cuando compran; es la suma de los atributos físicos, psicológicos, simbólicos y de servicio.(38)

➤ **Lotes:**

Los lotes son las fracciones de una producción caracterizada por la homogeneidad en sus propiedades y en sus condiciones de fabricación. (38)



**Anexo 2 Factura – Recepción de operaciones comerciales con productores individuales agropecuarios.**

RECEPTOR	Entidad	Código	FACTURA – RECEPCION DE OPERACIONES COMERCIALES CON PRODUCTORES INDIVIDUALES AGROPECUARIOS				D	M	A
	Unidad	Código							
	Almacén	Código							
Nombre productor individual			Contrato No.	No. Identidad Permanente					
Transportador			Chapa No.	Dirección Particular:					
CODIO	DESCRIPCION	Unidad de Medida	Cantidad	Precio Unitario	Importe	Saldo en Exist.			
IMPORTE TOTAL					\$	No.			
Nombre del que recibe	Firma del que recibe	Recepcionado por:	Anotado control Inventario por	Contabilizado por:					
Nombre del que entrega	Firma del que entrega								

**Figura 36 Factura – Recepción de operaciones comerciales con productores individuales agropecuarios.**

**Anexo 3 Informe de recepción.**

INFORME DE RECEPCION								
ENTIDAD		CODIGO.	ALMACEN		COD.	<u>D</u>	<u>M</u>	<u>A</u>
SUMINISTRADOR		CODIGO	CONTRATO No.	FACTURA No	CONDUCE No.			
TRANSPORTADOR	C. IDENTIDAD		CHAPA No.		B U L T O S			
					CANTIDAD:	TIPO:		
CODIGO	DESCRIPCION	U. M.	CANT.	PRECIO	IMPORTE	SALDO		
Los productos recibidos SI_ NO_ Corresponden a la calidad, especificaciones, estado de conservación y cantidades que muestren los documentos del suministrador.		Reclamación No.	Recepcionado por:		Anotado S.Mayor Inventario por:			
Jefe Almacén		Firma	Contabilizado por:			No.		

**Figura 37 Informe de recepción.**















**Anexo 10 Documento de Recepción Model.**

<b>Nombre: DocRecepcionModel.</b>	
<b>Tipo de clase: Modelo</b>	
<b>Atributo</b>	<b>Tipo</b>
\$nrorec	numeric
\$idsum	numeric
\$idtransp	numeric
\$nombre_transp	varchar
\$ci_transp	varchar
\$cartaporte	numeric
\$idgrado_transp	numeric
\$bcant	numeric
\$chapa	varchar
\$emb	varchar
\$manif	numeric
\$partida	varchar
\$contrato	varchar
\$casilla	entero
\$btipo	varchar
\$observaciones	varchar
\$infodif	entero
\$idgradosuministrador	numeric
\$nombresuministrador	varchar
\$cisuministrador	entero
\$idtransportador	entero
\$noconduce	entero
\$fechaconduce	date
\$nrofactura	entero

\$fdoc	date
\$idtipoderepcion	entero
\$nrotransferencia	entero
\$fechatransferencia	date
<b>Para cada responsabilidad</b>	
<b>Nombre.</b>	<b>Descripción.</b>
DocRecepcionModel(\$idestructuracomun = ", \$idusuario = ", \$observaciones=", \$nrorec=", \$idsun = ", \$idtransp = ", \$nombre_transp=", \$ci_transp = ", \$cartaporte=", \$idgrado_transp=", \$fdoc=", \$bcant=", \$chapa=", \$emb = ", \$manif=", \$partida=", \$contrato=", \$casilla=", \$btipo=", \$infodif=", \$idcomprador=", \$idgradosuministrador=", \$nombresuministrador=", \$cisuministrador=", \$nroconduce=", \$fechaconduce=", \$nrofactura=", \$fechacontable=", \$idtransportador=", \$idtipoderepcion=", \$nrotransferencia=", \$fechatransferencia=")	Constructor de la clase.
devolverRecepcion(\$iddocumento)	Responsabilidad encargada de devolver los datos de un Informe de Recepción.
CargarDatosEncabezadoModificar(\$iddocumento)	Responsabilidad encargada de devolver los datos del encabezado y de un Informe de Recepción a modificar.
adicionarDoc(\$doc) modificar(\$iddocumento, campos)	Responsabilidades encargadas de la gestión de los Informes de Recepción.
cancelarDoc(\$iddocumento)	Responsabilidad encargada de cambiar a estado cancelado un Informe de Recepción previamente contabilizado.

Tabla 15 Documento de Recepción Model.

**Anexo 11 Documento de Diferencia Model.**

<b>Nombre: DocDiferenciaModel.</b>	
<b>Tipo de clase: Modelo</b>	
<b>Atributo</b>	<b>Tipo</b>
\$acceptada	numeric
\$idinformerecepcion	numeric
\$fechalegal	date
\$estadobulto	varchar
\$observacionsum	varchar
\$viatransportacion	numeric
<b>Para cada responsabilidad</b>	
<b>Nombre.</b>	<b>Descripción.</b>
DocDiferenciaModel(\$idestructuracomun = ", \$idusuario = ", \$observsum = ", \$idinformerecepcion = ", \$acceptada = ", \$viatransportacion = ", \$observaciones = ")	Constructor de la clase.
CargarDiferencia(\$idestructuracomun,\$iddocumomento)	Responsabilidades encargadas de cargar los datos del Informe de Diferencias a crear.
adicionarDoc(\$doc) eliminar(\$iddocumento,\$alias)	Responsabilidades encargadas de la gestión de los Informes de Diferencias.
aprobar(\$iddocumento)	Responsabilidad encargada de cambiar a estado aprobado un Informe de Diferencias.

**Tabla 16 Documento de Diferencia Model.**

**Anexo 12 Movimiento de productos del Informe de Recepción.**

<b>Nombre: movrepcionfacturaModel.</b>	
<b>Tipo de clase: Modelo</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad</b>	
<b>Nombre.</b>	<b>Descripción.</b>
movrepcionfacturaModel()	Constructor de la clase.
insertarmov(\$iddocumento, cantidad,\$idproducto, \$existenciaactual,\$preciopromedio,\$importetotalprod)	Responsabilidad encargada de insertar un movimiento.
modificarmov(\$idmovimiento,\$idmoneda, \$idoperacion,\$precio, \$cantidad,\$importetotalprod,\$concepto)	Responsabilidad encargada de modificar los datos de un movimiento.
buscarmov(\$iddocumento,\$idproducto)	Responsabilidades encargadas de buscar los datos de un movimiento.
importemov(\$iddocumento)	Responsabilidad encargada de calcular el importe de los productos de un documento.
getdatos(\$iddocumento)	Responsabilidad encargada de mostrar los datos de los movimientos de un documento.
ubicado(\$idmovimiento)	Responsabilidad encargada de verificar si un producto del documento está ubicado.
setubicacion(\$idmovimiento, \$ubicacion)	Responsabilidad encargada de modificar el dato ubicación de un movimiento.
updatecategoria(\$idmovimiento,\$idcategoria)	Responsabilidad encargada de actualizar el dato categoría de un producto del documento.
updateprecioventa(\$idmovimiento,\$precioventa)	Responsabilidad encargada de actualizar el dato precio venta de un producto del documento.

**Tabla 17 Movimiento de productos del Informe de Recepción.**

**Anexo 13 Movimiento de productos del Informe de Diferencia.**

<b>Nombre: movdiferenciaModel.</b>	
<b>Tipo de clase: Modelo</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad</b>	
<b>Nombre.</b>	<b>Descripción.</b>
movdiferenciaModel()	Constructor de la clase.
insertarmov(\$idmovimiento,\$cantidad,\$idcalidad,\$cantidadcalidad,\$motivo)	Responsabilidad encargada de insertar un movimiento de diferencias.
eliminararmov(\$idmovimiento,\$idcalidad)	Responsabilidad encargada de eliminar un movimiento de diferencias.
modificarmov(\$iddiferencia,\$idcalidad,\$cantidadcalidad,\$cantidad,\$motivo)	Responsabilidad encargada de modificar los datos calidad, cantidad, y motivo de un movimiento de diferencia.
updateprecio(\$idmovdiferencia,\$idunidadmonetaria,\$idmoneda,\$preciodiferencia,\$importediferencia)	Responsabilidad encargada de modificar los datos precio e importe de un movimiento de diferencias.
buscarmov(\$idmovimiento) buscarmovcal(\$idmovimiento,\$idcalidad)	Responsabilidad encargada de buscar los datos de un movimiento de diferencias.
updatecondif(\$iddocumento,\$idproducto)	Responsabilidad encargada de modificar el atributo condif de un movimiento.
ubicado(\$idmovimiento)	Responsabilidad encargada verificar si un producto del documento está ubicado.
movcondiferencia(\$iddocumento)	Responsabilidad encargada verificar si existe algún producto de un documento con diferencias.

**Tabla 18 Movimiento de productos del Informe de Diferencia.**



**Anexo 14 Instrumento de medición de la métrica Tamaño operacional de clase (TOC).**

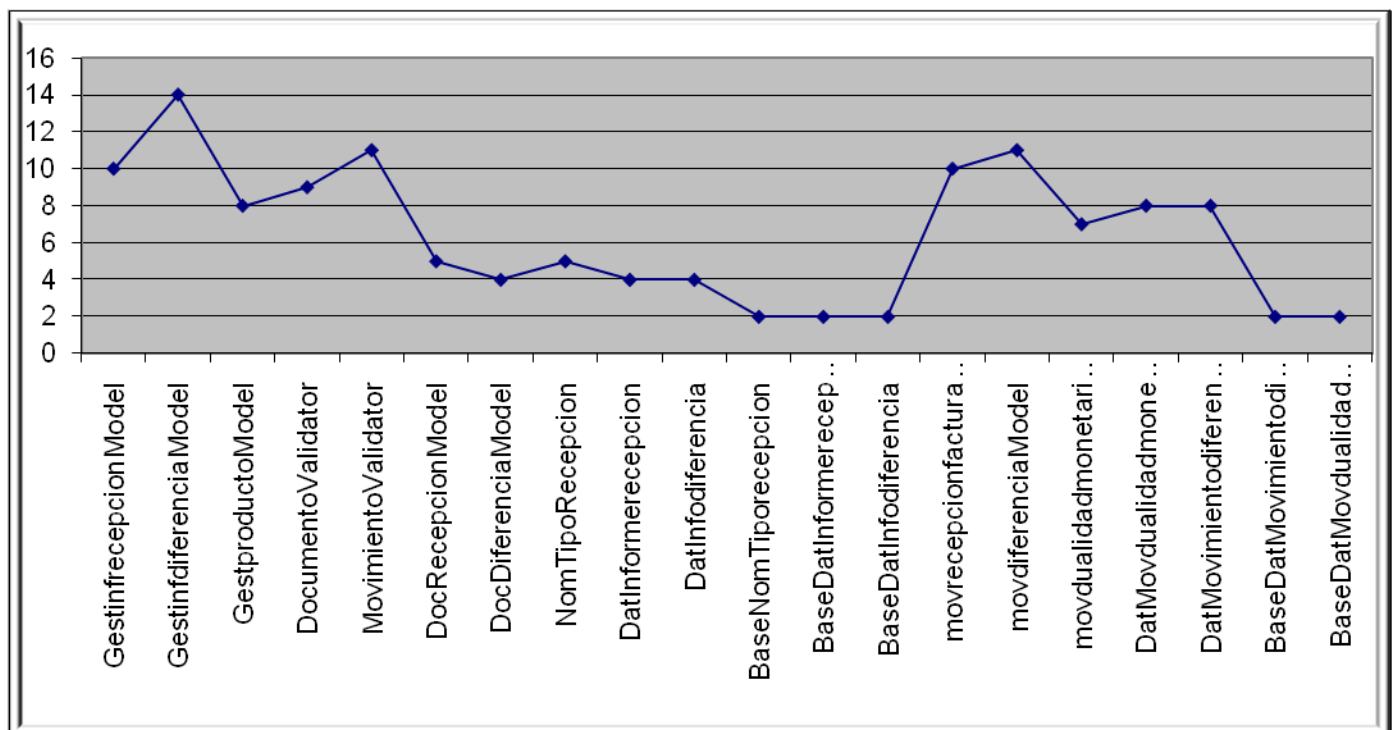
	CATEGORIA	CRITERIO
<b>RESPONSABILIDAD</b>	Baja	Entre 0 y 5
	Media	Entre 6 y 12
	Alta	Más de 13
	CATEGORIA	CRITERIO
<b>COMPLEJIDAD IMPLEMENTACIÓN</b>	Baja	Entre 0 y 7
	Media	Entre 8 y 14
	Alta	Más de 15
	CATEGORIA	CRITERIO
<b>REUTILIZACIÓN</b>	Baja	Más de 11
	Media	Entre 10 y 6
	Alta	Entre 0 y 5

**Tabla 19 Rango de valores de para la evaluación técnica de los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización) relacionados con la métrica TOC.**

No	Subsistema	Clase	Cantidad de Procedimientos	Responsabilidad	Complejidad	Reutilización
1	Recepcion	GestinfrecepcionModel	10	Media	Media	Media
2	Recepcion	GestinfdiferenciaModel	14	Alta	Media	Baja
3	Recepcion	GestproductoModel	8	Media	Media	Media
4	Recepcion	DocumentoValidator	9	Media	Media	Media
5	Recepcion	MovimientoValidator	11	Media	Media	Media
6	Documento	DocRecepcionModel	5	Baja	Baja	Alta
7	Documento	DocDiferenciaModel	4	Baja	Baja	Alta
8	Documento	NomTipoRecepcion	5	Baja	Baja	Alta
9	Documento	DatInformerecepcion	4	Baja	Baja	Alta
10	Documento	DatInfodiferencia	4	Baja	Baja	Alta
11	Documento	BaseNomTiporecepcion	2	Baja	Baja	Alta

12	Documento	BaseDatInformerecepcion	2	Baja	Baja	Alta
13	Documento	BaseDatInfodiferencia	2	Baja	Baja	Alta
14	Movimiento	movrecepcionfacturaModel	10	Media	Media	Media
15	Movimiento	movdiferenciaModel	11	Media	Media	Media
16	Movimiento	movdualidadmonetariaModel	7	Media	Baja	Media
17	Movimiento	DatMovdualidadmonetaria	8	Media	Media	Media
18	Movimiento	DatMovimientodiferencia	8	Media	Media	Media
19	Movimiento	BaseDatMovimientodiferencia	2	Baja	Baja	Alta
20	Movimiento	BaseDatMovdualidadmonetaria	2	Baja	Baja	Alta

**Tabla 20 Resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización).**



**Figura 44 Gráfica de los resultados de la evaluación de la métrica TOC y su influencia en los atributos de calidad (Responsabilidad, Complejidad de Implementación y Reutilización).**

**Anexo 15 Instrumento de medición de la métrica Relaciones entre clases (RC).**

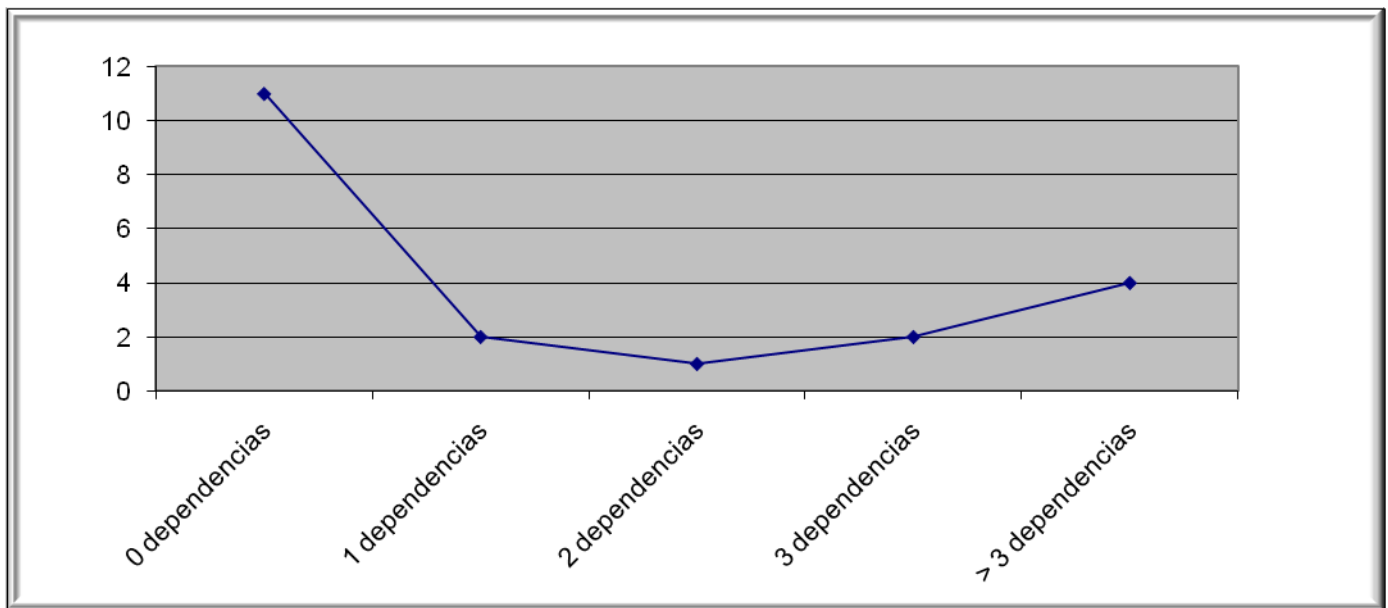
	CATEGORIA	CRITERIO
<b>ACOPLAMIENTO</b>	Ninguno	0
	Bajo	Entre 1 y 3
	Medio	Entre 4 y 8
	Alto	Más de 8
	CATEGORIA	CRITERIO
<b>COMPLEJIDAD MANT.</b>	Baja	Menor 5
	Media	Entre 6 y 10
	Alta	Más de 10
	CATEGORIA	CRITERIO
<b>REUTILIZACIÓN</b>	Baja	Más de 12
	Media	Entre 6 y 12
	Alta	Entre 0 y 5

**Tabla 21 Rango de valores de para la evaluación técnica de los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas) relacionados con la métrica RC.**

No	Subsistema	Clase	Cantidad de Relaciones de Uso	Acoplamiento	Complejidad Mant.	Reutilización
1	Recepcion	GestinfrecepcionModel	10	Alto	Media	Media
2	Recepcion	GestinfdiferenciaModel	7	Medio	Media	Media
3	Recepcion	GestproductoModel	9	Alto	Media	Media
4	Recepcion	NomTipoRecepcion	0	Ninguno	Baja	Alta
5	Recepcion	DocumentoValidator	3	Bajo	Baja	Alta
6	Recepcion	MovimientoValidator	3	Bajo	Baja	Alta
7	Documento	DocRecepcionModel	1	Bajo	Baja	Alta
8	Documento	DocDiferenciaModel	0	Ninguno	Baja	Alta
9	Documento	DatInformerecepcion	0	Ninguno	Baja	Alta
10	Documento	DatInfodiferencia	0	Ninguno	Baja	Alta
11	Documento	BaseNomTiporecepcion	0	Ninguno	Baja	Alta

12	Documento	BaseDatInformerecepcion	0	Ninguno	Baja	Alta
13	Documento	BaseDatInfodiferencia	0	Ninguno	Baja	Alta
14	Movimiento	movrecepcionfacturaModel	1	Bajo	Baja	Alta
15	Movimiento	movdiferenciaModel	4	Medio	Baja	Alta
16	Movimiento	movdualidadmonetariaModel	2	Bajo	Baja	Alta
17	Movimiento	DatMovdualidadmonetaria	0	Ninguno	Baja	Alta
18	Movimiento	DatMovimientodiferencia	0	Ninguno	Baja	Alta
19	Movimiento	BaseDatMovimientodiferencia	0	Ninguno	Baja	Alta
20	Movimiento	BaseDatMovdualidadmonetaria	0	Ninguno	Baja	Alta

**Tabla 22 Resultados de la evaluación de la métrica RC y su influencia en los atributos de calidad (Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas)**

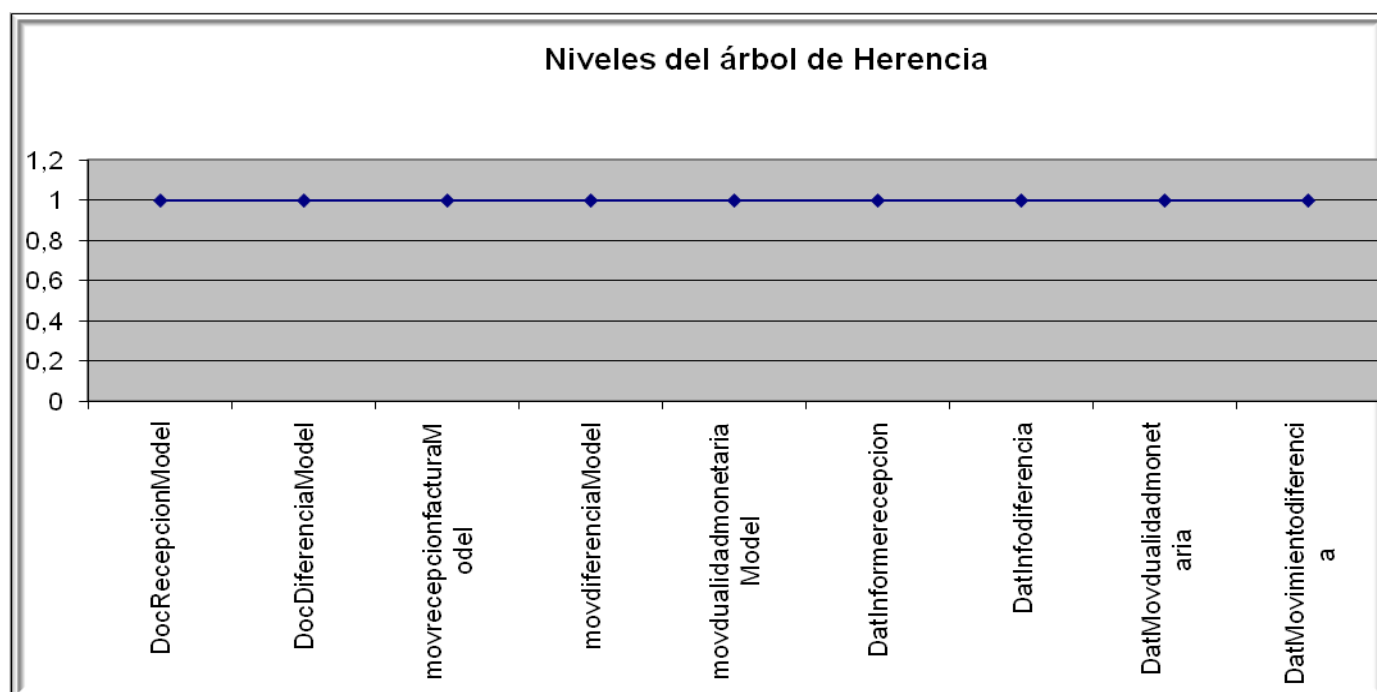


**Figura 45 Gráfica de los resultados de la evaluación de la métrica RC agrupados por la tendencia de los valores.**

**Anexo 16 Instrumento de medición de la métrica Profundidad de herencia.**

No	Subsistema	Clase	Clase padre	Niveles del árbol de herencia	Profundidad de herencia
1	Documento	DocRecepcionModel	DocumentoModel	1	Bajo
2	Documento	DocDiferenciaModel	DocumentoModel	1	Bajo
3	Movimiento	movrecepcionfacturaModel	movimientoModel	1	Bajo
4	Movimiento	movdiferenciaModel	movimientoModel	1	Bajo
5	Movimiento	movdualidadmonetariaModel	movimientoModel	1	Bajo
6	Documento	DatInformerecepcion	BaseDatInformerecepcion	1	Bajo
7	Documento	DatInfodiferencia	BaseDatInfodiferencia	1	Bajo
8	Movimiento	DatMovdualidadmonetaria	BaseDatMovdualidadmonetaria	1	Bajo
9	Movimiento	DatMovimientodiferencia	BaseDatMovimientodiferencia	1	Bajo

**Tabla 23 Resultados de la evaluación de la métrica PH.**



**Figura 46 Gráfica de los resultados de la evaluación de la métrica PH.**

**Anexo 17 Instrumento de medición de la métrica Número de Operaciones Redefinidas (NOR).**

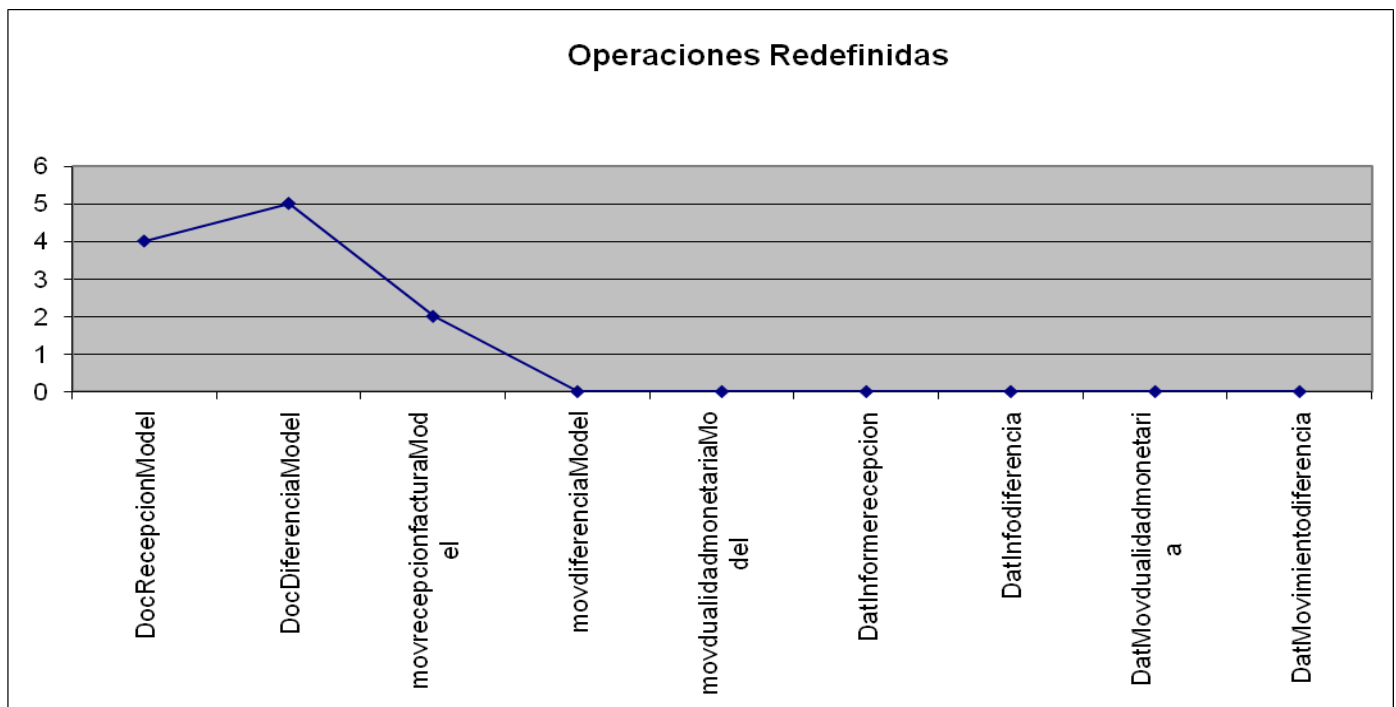
	CATEGORIA	CRITERIO
<b>COMPLEJIDAD DE MANTENIMIENTO</b>	Baja	Entre 0 y 2
	Media	Entre 3 y 5
	Alta	Más de 5
	CATEGORIA	CRITERIO
<b>CANTIDAD DE PRUEBAS</b>	Baja	Entre 0 y 2
	Media	Entre 3 y 5
	Alta	Más de 5
	CATEGORIA	CRITERIO
<b>VIOLACIÓN DE LA ABSTRACCIÓN REPRESENTADA POR LA SUPERCLASE</b>	Baja	Entre 0 y 2
	Media	entre 3 y 5
	Alta	Más de 5

Tabla 24 Rango de valores de para la evaluación técnica de los atributos de calidad (Abstracción del diseño, Cantidad de pruebas, Complejidad de Mantenimiento) 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	Documento	DocRecepcionModel	DocumentoModel	4	Aceptable	Media	Media	Media
2	Documento	DocDiferenciaModel	DocumentoModel	5	Aceptable	Media	Media	Media
3	Movimiento	movrecepcionfacturaModel	movimientoModel	2	Aceptable	Media	Baja	Bajo
4	Movimiento	movdiferenciaModel	movimientoModel	0	Buena	Baja	Baja	Bajo
5	Movimiento	movdualidadmonetariaModel	movimientoModel	0	Buena	Baja	Baja	Bajo
6	Documento	DatInformerece	BaseDatInfor	0	Buena	Baja	Baja	Bajo

		opcion	merecepcion					
7	Documento	DatInfodiferencia	BaseDatInfo diferencia	0	Buena	Baja	Baja	Bajo
8	Movimiento	DatMovdualidadmonetaria	BaseDatMov dualidadmonetaria	0	Buena	Baja	Baja	Bajo
9	Movimiento	DatMovimiento diferencia	BaseDatMovi mientodiferencia	0	Buena	Baja	Baja	Bajo

**Tabla 25 Resultados de la evaluación de la métrica NOR y su influencia en los atributos de calidad (Abstracción del diseño, Cantidad de pruebas, Complejidad de Mantenimiento).**



**Figura 47 Representación de los resultados obtenidos al aplicar los instrumentos que evalúan la métrica NOR.**