Universidad de las Ciencias Informáticas.

“Interoperabilidad del proceso inventario en el sistema Cedrux”.

Trabajo de Diploma para optar por el título de

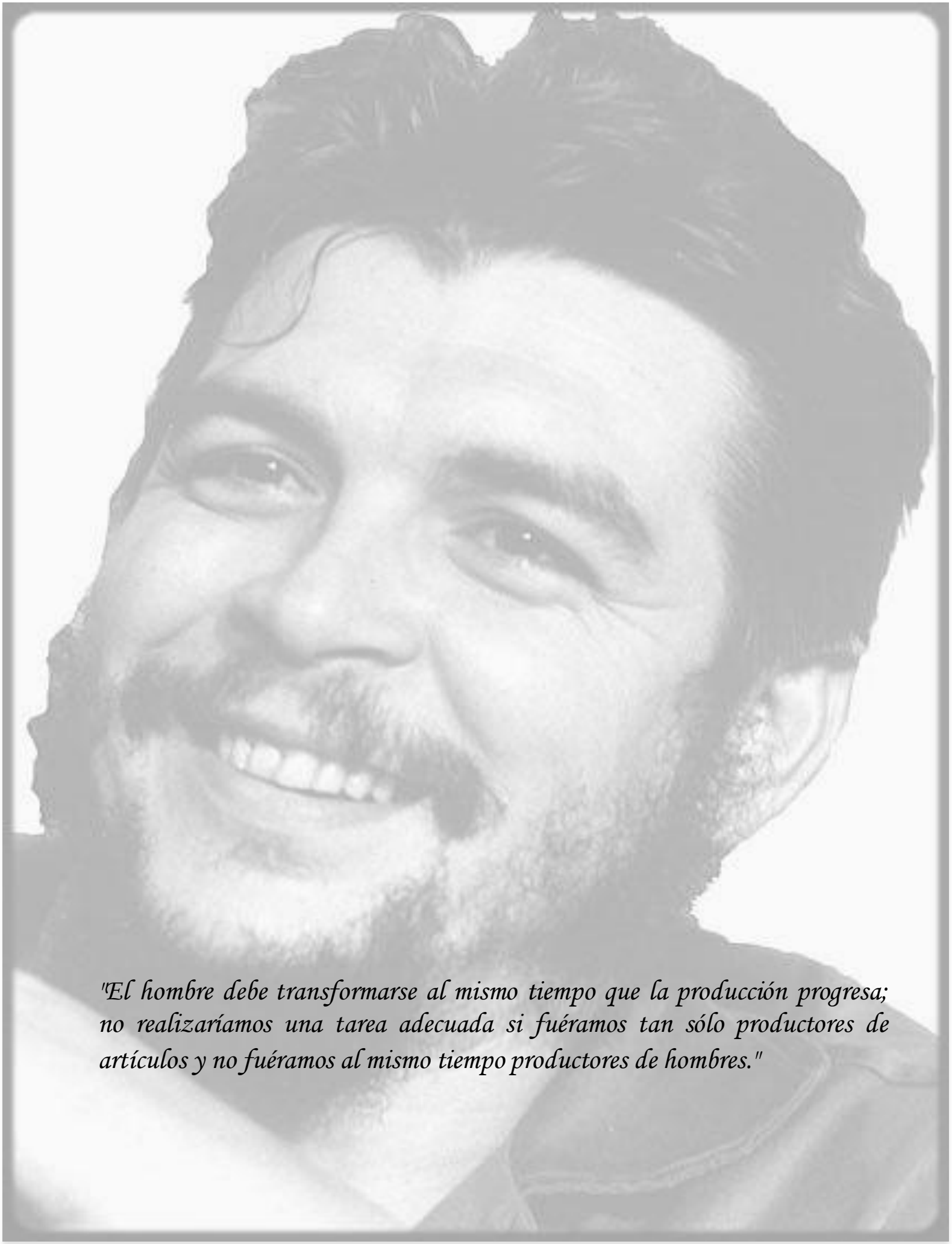
Ingeniero Informático.

Autor: Oscar Oramas González.

Tutor(es): Ing. Pedro Manuel Nogales Cobas.

Ing. Magdanis Galván Rey.

Junio de 2011



"El hombre debe transformarse al mismo tiempo que la producción progresa; no realizaríamos una tarea adecuada si fuéramos tan sólo productores de artículos y no fuéramos al mismo tiempo productores de hombres."

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al centro CEIGE de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

_____ _____

Oscar Oramas González Ing. Pedro Manuel Nogales Cobas

Autor **Tutor**

_____

Ing. Magdanis Galván Rey

Tutor

Datos de Contacto

Autor: Oscar Oramas González.....Correo: ooramas@estudiantes.uci.cu

Tutor: Ing. Pedro Manuel Nogales Cobas.....Correo: pmnogales@uci.cu

Tutor: Ing. Magdanis Galván Rey.....Correo: mgalvan@uci.cu

AGRADECIMIENTOS

Quiero agradecer en primer lugar a mi abuela mimi, por ser una madre ejemplar en todos los sentidos, te quiero muchísimo. Gracias por criarme, por ser mi guía, por todos estos años de sacrificio, por brindarme amor y comprensión.

A mi mamá por ser una madre ejemplar igualita a mimi, te agradezco mucho por brindarme todo el amor y cariño del mundo, me quedan cortas estas palabras, sinceramente.

A mi tía yeya por ser mi tercera mamá, siempre preocupándose por mí, soy privilegiado por tener tres mamá. Te quiero muchísimo.

A mi papá por ser mi guía, mi ejemplo a seguir. Siempre dije que cuando grande iba a ser igual que mi papá un ingeniero, parece que voy a lograr mis metas. Gracias por mostrarme el camino a seguir. Un abrazo enorme para ti.

A mi abuelo (papá chino) por ser un padre para mí, por brindarme sabios consejos, al igual te quiero mucho y mi tío Fernando por siempre estar preocupado por mis estudios se que estará muy orgulloso de mi.

A mis dos hermanas Anyelis y Anisley quiero que sigan mi ejemplo y sean profesionales en un futuro. Las quiero a las dos por igual.

A mi madrastra Salmita por ser la madrastra mas buena del mundo y mis dos hermanos Lioscar y Liosbanis los quiero mucho.

A mi abuelita nena y abuelito candito por ser unos abuelos ejemplares y cariñosos.

Quiero agradecer a mis amigos en la universidad por compartir estos 5 años de sacrificios. En especial a Rey, Grabiel, Danarys, Elton, Tito, Driggs, Mayte, Nely, Ariagna, leydi, Neosotis, Muñequito, Sandy en fin el aula antigua de 1ero y 2do año.

Agradecer a mis tutores Pedro y Magdanis por brindarme su ayuda. Gracias por sus consejos y observaciones.

Al tribunal y al oponente por sus consejos y recomendaciones.

Agradecer de todo corazón a todos mis amigos a los que mencioné y a los que no, nunca los olvidaré. Gracias por existir.

DEDICATORIA

Dedico este trabajo a mi abuela mimi por ser una de las personas más importante de mi vida.

RESUMEN

La interoperabilidad es una característica importante que deben cumplir los sistemas de planificación de recursos empresariales (ERP) que es la capacidad de intercambiar procesos o datos entre sistemas heterogéneos.

Con el desarrollo del sistema Cedrux llevado a cabo por el Centro de Informatización de la Gestión de Entidades, de la Universidad de las Ciencias Informáticas, se consideró que debe tener como una de sus premisas fundamentales, el ser capaz de interoperar con otros sistemas homólogos, cuyos beneficios estén orientados a la extensibilidad del producto en diferentes entornos, la homogeneidad y el orden, así como la capacidad de distribuir sus datos.

Una enorme dificultad para lograr la interoperabilidad entre sistemas heterogéneos, es que suelen ser desarrollados sin ningún requisito para interoperar; lográndose por lo general mediante la utilización de diferentes tecnologías como son: estándares de marcado, lenguajes de consulta y servicios web.

Este trabajo tiene como objetivo el desarrollo de un componente de interoperabilidad que permita el intercambio de datos de una forma transparente y seguro con los demás sistemas de gestión homólogos instalados en las diferentes entidades a nivel nacional. Además de detallar las decisiones tomadas en el desarrollo del mismo, describiendo las tecnologías a emplear y las nuevas funcionalidades implementadas, con el objetivo de solucionar los problemas de intercambio de datos.

Palabras claves: interoperabilidad, intercambio de datos, tecnologías.

TABLA DE CONTENIDO

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

En la actualidad el avance en las tecnologías de la información ha incidido satisfactoriamente en todas las esferas de la sociedad, provocando que empresas e instituciones tengan la necesidad de nuevas tecnologías informáticas, capaces de gestionar los procesos que en ellas se llevan a cabo en aras de alcanzar la eficiencia económica, es por ello que surgen los Sistemas de Planificación de Recursos Empresariales por sus siglas en inglés (ERP), su gran adaptabilidad a la realidad de cada empresa y su integración, permiten minimizar el tiempo para realizar y analizar informes mejorando de este modo la productividad.

Cuba no se encuentra alejada de dichas tecnologías, y por tanto se ha visto en la mera necesidad de adoptar tanto soluciones nacionales como internacionales, para no verse excluida de dichos avances que pueden mejorar el desarrollo del país. Muchos de estos sistemas tienen una particularidad, se desarrollan para resolver dificultades específicas de entidades, provocando que existan soluciones de diferentes fabricantes para la gestión empresarial.

Debido a la importancia que tienen estos sistemas para la economía, la máxima dirección del país en el año 2008 le encomendó la tarea de la creación de un Sistema de Planificación de Recursos Empresariales a la Universidad de Ciencias Informáticas (UCI), surgiendo así el sistema integral de gestión Cedrux, con el objetivo de ser una única solución a nivel nacional, con funcionalidades generales adaptadas a las particularidades de la economía, y también de eliminar las enormes sumas de capital en cuestión de pagos de licencias a compañías productoras de sistemas ERP.

Cedrux se encuentra dividido por subsistemas que manejan los recursos materiales, humanos y financieros para lograr un mejor control, distribución y planificación de los recursos. Absorber gradualmente la información contenida en el resto de los sistemas utilizados, es premisa fundamental en el desarrollo, surgiendo la idea de implementar un Componente de Interoperabilidad, el cual tiene la tarea de desplazar a todos los sistemas de este tipo que se usan en las diferentes entidades cubanas y así lograr un proceso de intercambio de información que permita la comunicación oportuna, íntegra y segura de todos sus datos. Logística es uno de los subsistemas mencionados anteriormente y en la actualidad no cuenta con una solución que permita la interoperabilidad de los datos contenidos en los diferentes documentos de inventario, entre los que se encuentran: apertura, recepción, informe de diferencias,

inventario físico entre otros, utilizados en el control de las existencias, así como los movimientos de los productos dentro del almacén.

Por lo antes señalado se plantea como ***problema a resolver*** la necesidad de desarrollar una solución informática para lograr la transferencia de datos del proceso inventario entre el sistema Cedrux y las aplicaciones de gestión usadas en Cuba.

Tomando en este caso como ***objeto de estudio*** la interoperabilidad en sistemas integrales de gestión.

Para darle solución al problema antes planteado se define como ***objetivo general*** desarrollar la interoperabilidad del proceso inventario en el sistema Cedrux, el cual se desglosa en los siguientes ***objetivos específicos:***

- Realizar el marco teórico de la investigación.
- Realizar el análisis y diseño de la solución.
- Implementar la solución.
- Validar la solución.

Teniendo como ***campo de acción*** la interoperabilidad del proceso inventario en sistemas integrales de gestión en Cuba.

Idea a defender

El desarrollo de una solución informática de interoperabilidad, permitirá que se logre la transferencia de datos del proceso inventario entre el sistema Cedrux y las demás aplicaciones de gestión usadas en Cuba.

Para dar solución a los objetivos especificados se plantean las siguientes ***tareas investigativas***:

- Estudio bibliográfico sobre la interoperabilidad de software y su desarrollo a nivel nacional e internacional.
- Realizar un estudio del estado del arte de las principales herramientas y framework relacionados con el objeto de estudio de la investigación.
- Describir los procesos de negocio a automatizar.
- Obtener y especificar los requisitos.

- Definir los escenarios arquitectónicos de la herramienta.
- Evaluar la calidad del diseño de la solución a través de métricas de validación.
- Evaluar la calidad de la solución desarrollada a través de las pruebas definidas por el Grupo de Calidad del CEIGE (Centro de Informatización de la Gestión de Entidades).
- Validar la solución mediante su aplicación en un entorno real.
- Realizar la documentación de transferencia para el proceso Inventario.

El documento se encuentra estructurado de la siguiente manera:

Capítulo 1: En este capítulo se realiza la fundamentación teórica**,** en la misma se incluyen las estrategias utilizadas en el mundo para que se logre la interoperabilidad de los sistemas así como un estudio de las diferentes tecnologías que serán empleadas para la realización de la solución.

Capítulo 2: En este se abarca la propuesta de solución y se exponen los procesos de negocios y los requisitos a cumplir por el componente además de los artefactos generados durante el diseño de la misma.

Capítulo 3: En su contenido se reflejan las pruebas realizadas y analizan los resultados obtenidos. Además se hace una validación por especialista de los resultados.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se hace referencia al marco teórico de la investigación, donde se realiza un estudio del estado del arte y se aborda el concepto de interoperabilidad, sus ventajas, buenas prácticas y las formas de interoperar. Así como su puesta en práctica en aplicaciones web de gestión. Además se describen las tecnologías y herramientas que se van a utilizar en la implementación del componente de interoperabilidad.

1.1 Definición de interoperabilidad

"La interoperabilidad es la capacidad de los sistemas de información y de los procedimientos a los que éstos dan soporte, de compartir datos y posibilitar el intercambio de información y conocimiento entre ellos". (1)

La interoperabilidad es definida por la IEEE como "la habilidad de dos o más sistemas o componentes para intercambiar información y para usar la información que ha sido intercambiada". (2)

Según el Lic. Keilyn Rodríguez Perojo lo define así: (3)

- **Interoperabilidad sintáctica**: Se refiere a la capacidad de los sistemas de información para leer datos procedentes de otros similares y obtener una representación que pueda ser compatible.
- **Interoperabilidad semántica**: Es la capacidad de los sistemas de información para intercambiar información basándose en un común significado de los términos y expresiones que se usan.
- **Interoperabilidad estructural**: Es la capacidad de los sistemas de información de interactuar en ambientes no solo cerrados, sino distribuidos, soportados por protocolos de intercambio y acceso comunes a redes de datos tales como TCP/IP.

De modo general se puede definir a la interoperabilidad como la capacidad de dos sistemas informáticos de intercambiar datos entre sí, con el objetivo de utilizarlo para beneficio propio.

1.1.1 Formas de interoperar

En la mayoría de los casos una gran dificultad para lograr la interoperabilidad entre los sistemas heterogéneos[1] es que suelen ser desarrollados independientemente sin ningún requisito para interoperar, presentando en la mayoría de los casos, diferentes arquitecturas, plataformas de hardware, sistemas operativos, lenguajes de máquina y modelos de datos. Lográndose por lo general, mediante estándares de marcado, lenguajes de consulta y servicios web.

- **Estándares de marcado**

XML el lenguaje de marcado extensible (Extensible Markup Language XML) es un metalenguaje extensible de etiquetas, desarrollado por el World Wide Web Consortium (W3C). Este no es realmente un lenguaje en particular, sino una manera de definir lenguajes para distintas necesidades. XML no nació sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Este en la actualidad tiene un papel importante, ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. (4)

- **Lenguajes de consulta**

SQL el lenguaje de consulta estructurado (Structured Query Language) es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional, permitiendo consultar con el fin de recuperar, de una forma sencilla, información de interés de una base de datos, así como también hacer cambios sobre la misma. (4)

XPath (XML Path Language) es un lenguaje que permite construir expresiones que recorren y procesan un documento XML. La idea es parecida a las expresiones regulares para seleccionar partes de un texto sin atributos. Este permite buscar y seleccionar teniendo en cuenta la estructura jerárquica del XML. Se creó para su uso en el estándar de lenguaje extensible de hojas de estilo (Extensible Stylesheet Language Transformations XSLT), en el que se usa para seleccionar y examinar la estructura del documento de entrada de la transformación. (4)

[1] Heterogéneo: Utilizado para referirse a un sistema informático, quiere expresar que sus componentes presentan arquitecturas diferentes. Generalmente, cualquier sistema informático en una empresa es heterogéneo, puesto que integra desde PCs a mainframes o miniordenadores.

XQuery (XML *Query*) es un lenguaje de consultas diseñado para realizar búsquedas en colecciones de datos XML. Es semánticamente similar al SQL, pero incluye algunas capacidades de programación. Proporciona los medios para extraer y manipular información de documentos XML, o de cualquier fuente de datos que se puedan representar mediante XML, por ejemplo, bases de datos relacionales o documentos ofimáticos. XQuery utiliza expresiones XPath para acceder a determinadas partes del documento XML. (4)

- **Servicios web**

Un **servicio web** es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software, desarrolladas en diferentes lenguajes de programación y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son las responsables de la arquitectura y reglamentación de los servicios web. (4)

Se caracteriza principalmente por 4 estándares de comunicación, como son XML para la representación de datos, SOAP (Protocolo Simple de Acceso a Objetos) para el intercambio de datos, el lenguaje WSDL (Lenguaje de Descripción de Servicios Web) para describir las funcionalidades de un servicio Web y UDDI (Descripción, Descubrimiento e integración Universal). (5)

> El protocolo **SOAP** utiliza mensajes XML para invocar métodos remotos. Un servicio web podría interactuar con servicios remotos a través de los métodos get y post de http, pero SOAP es mucho más robusto y flexible. SOAP es un protocolo liviano, basado en XML, para el intercambio de información estructurada en un ambiente descentralizado y distribuido. Sin embargo, no define la aplicación, ni la semántica de implementación. En vez de esto, proporciona un modelo de empaquetamiento modular y los mecanismos para la codificación de los datos dentro de los módulos. Esto permite que el protocolo simple se utilice en una amplia variedad de sistemas modulares y que cumpla su propósito primario de facilitar la interoperabilidad entre componentes de software heterogéneos. (5)
>
> **UDDI** es una especificación para un registro distribuido de información acerca de los servicios web. Define la forma en la cual se publica y descubre información acerca de éstos. Un registro basado

en UDDI es donde se pueden descubrir los servicios web. El método utilizado por UDDI para el descubrimiento de servicios es tener un registro de aquellos servicios que se encuentran distribuidos a través de la web. En el registro distribuido, los negocios y los servicios se describen utilizando un formato XML común. Los datos estructurados en esos documentos XML son de fácil búsqueda, análisis y manipulación. (5)

WSDL es el lenguaje común utilizado para la descripción de los servicios web. Es un lenguaje basado en XML que describe totalmente la forma en la cual los clientes externos pueden interactuar con los servicios web existentes en una máquina dada, los métodos que soportan y la sintaxis de los protocolos de comunicación (http, SOAP). En términos generales, un documento WSDL contiene información acerca de la interfaz, la semántica y los aspectos administrativos involucrados en una solicitud (llamado) a un servicio web. (5)

A continuación se muestran beneficios de los servicios web: (5)

- Promueven la interoperabilidad: La interacción entre un proveedor y un solicitante de servicio está diseñada para que sea completamente independiente de la plataforma y el lenguaje.
- Permiten la integración: El proceso de descubrimiento se ejecuta dinámicamente, a medida que los solicitantes de servicio utilizan a los agentes para encontrar proveedores de servicio. Una vez el solicitante y el proveedor de servicio se han ubicado, se utiliza el documento WSDL del proveedor para enlazar al solicitante con el servicio. Esto significa que los solicitantes, los proveedores y los agentes actúan en conjunto para crear sistemas que son auto-configurables, adaptativos y robustos.
- Reducen la complejidad por medio del encapsulamiento: Los solicitantes y los proveedores del servicio se preocupan por las interfaces necesarias para interactuar. Como resultado, un solicitante de servicio no sabe cómo fue implementado el servicio por parte del proveedor, y éste a su vez, no sabe cómo utiliza el cliente el servicio. Estos detalles se encapsulan en los solicitantes y proveedores. El encapsulamiento es crucial para reducir la complejidad.
- Disminuyen el tiempo de desarrollo de las aplicaciones: Pues gracias a la filosofía de orientación a objetos utilizada, el desarrollo se convierte más bien en una labor de composición.

Los estándares de marcado y los servicios web son tecnologías que han alcanzado un auge importante en cuestiones de interoperabilidad debido que presumen una solución sencilla, que evita en su mayor parte las dependencias de plataformas y fabricante.

1.1.2 Ventajas de la interoperabilidad

Ventajas de la interoperabilidad para sistemas informáticos:

- Mecanismos eficientes de intercambio de información.
- Sistemas que interoperan de una forma transparente y coherente.
- Logra la disponibilidad y visibilidad de los datos entre ambos sistemas.
- Aumenta la escalabilidad que es la propiedad deseable de un sistema, que indica su habilidad para extender el margen de operaciones sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida.
- Aumenta su reusabilidad en otros proyectos. Posibilitando integrar sistemas en el futuro sin necesidad de grandes inversiones.
- Mejora el intercambio de datos basado en estándares abiertos y publicados de forma libre.
- Ignora la heterogeneidad de los sistemas.

1.1.3 Buenas prácticas

En la actualidad, las tecnologías han ido avanzando en temas relacionados con el intercambio de datos y existen varias formas o propuestas que ayudan a desarrollar la interoperabilidad de un software. A continuación se hace mención de buenas prácticas con el objetivo de que puedan ser aplicadas en su desarrollo:

- Buscar la accesibilidad de los contenidos y el conocimiento, con independencia de la tecnología que se use para acceder a ellos. (6)

- Establecer políticas de seguridad y privacidad coherentes y fáciles de llevar a cabo, acordes con el nivel de amenaza y tipo de información a proteger. (6)

- Hacer uso exclusivo de estándares abiertos, convirtiendo a estos estándares toda la información que se encuentre en formatos privativos, o que se encuentre almacenada usando estándares abiertos en situación de abandono. (6)

- Siempre que sea posible, se debería optar por el uso de software libre. Debido que suele usar preferentemente estándares abiertos. La disponibilidad del código fuente también tiene como consecuencia el debate abierto y democrático sobre sus especificaciones y estándares, lo que lo suele hacer que el software libre sea más robusto e interoperable que las aplicaciones privativas equivalentes. (6)

- Si es posible, usar aplicaciones, o herramientas de desarrollo, que puedan funcionar en más de una plataforma tecnológica. En este sentido, las tecnologías como el XML, serán de gran ayuda para lograr los objetivos de interoperabilidad sin caer en las trampas de la dependencia tecnológica. (6)

- Aplicación de servicios web, esta tecnología mencionada anteriormente utiliza un conjunto de estándares y protocolos para el intercambio de información entre aplicaciones de software, con independencia de las directrices tecnológicas de las mismas.

- Se debe confeccionar un marco o arquitectura de interoperabilidad donde se rige un conjunto de estándares y directrices. Dicho marco de interoperabilidad no es un documento estático y podrá ser adaptado con el tiempo con los cambios de tecnologías, estándares y requerimientos administrativos.

1.2 Interoperabilidad en aplicaciones de gestión

Los sistemas de gestión, se encargan de la planificación, organización y control de los recursos tanto humanos como físicos que tienen que ver con el apoyo al desarrollo, mejoría y mantenimiento a los servicios de procesamiento, transformación, distribución, almacenamiento y recuperación de la información para una empresa.

En el mundo existen diversas soluciones ERP, entre la que se encuentran el JD Edwards, IFS, Microsoft Dynamics, SAP entre muchos más. Este último goza de una gran reputación a nivel mundial ya que es uno de los más usados. A continuación se muestra como estos sistemas llevan a cabo la interoperabilidad:

SAP NetWeaver proporciona servicios completamente habilitados para los procesos de negocio y una plataforma integral que ofrece el aprovisionamiento, producción, consumo y gestión de los servicios Web. Las normas que sirven de base para la apertura y la interoperabilidad permiten los fundamentos básicos de la plataforma de SAP de procesos de negocio. Dentro de la plataforma tecnológica hay categorías de normas de metadatos para la infraestructura, mensajería, marcos de componentes, y la fundación. La interoperabilidad de los servicios de la empresa sin duda requiere el apoyo de las normas pertinentes de servicios Web para describir los servicios y el intercambio de mensajes de una manera confiable y segura. Pero también requiere lenguajes comunes de negocios que puede ser entendido por todas las partes involucradas en el diseño, aprovisionamiento, la composición y el consumo de los servicios de la empresa. (7)

Microsoft Dynamics NAV es capaz de consumir y exponer servicios web. Para acceder a los servicios web, es un requisito para utilizar el nivel de servicio y SQL Server. La diferencia más importante entre los servicios Web y MSMQ[2] es que los servicios Web se ejecutan en tiempo real y sólo cuando la conexión entre las dos solicitudes es activa, mientras que la cola de MSMQ se solicita hasta que la conexión ha sido establecida. (8)

JD Edwards EnterpriseOne apoya mucho las transacciones EDI[3] más comunes. Base de datos activa es un método muy rápido para lograr la interoperabilidad de JD Edwards (especialmente cuando se usa entre aplicaciones diferentes). Sin embargo, la velocidad de lograr esta solución lleva consigo ciertas oportunidades que deben ser tomadas en consideración antes de su completa confianza en esta metodología. En primer lugar, la base de datos activa no utiliza el middleware[4] de JD Edwards. En el nivel de base de datos, pasando de Oracle[5] para Oracle es muy simplista, pero al pasar de Oracle para otro

[2] Microsoft Message Queue Server o MSMQ es un mensaje a la cola de la aplicación desarrollada por Microsoft

[3] EDI (Electronic Data Interchange). Transferencia electrónica de datos entre dos empresas, para eliminar el intercambio de documentación, facturas, etc.

[4] Programa que se realiza para conectar dos aplicaciones o sistemas Software.

[5] Oracle es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), desarrollado por Oracle Corporation.

sistema de gestión de datos se requieren programas sustanciales, o la introducción de un producto middleware de terceros para facilitar. Obviamente, esto introduce una complejidad adicional a la solución. (9)

IFS Applications debido a que sus componentes presentan una arquitectura orientada a servicios, otras aplicaciones y soluciones de punta pueden acceder fácilmente a la información y funcionalidades. Tanto la capa de servicios y el núcleo de la aplicación se pondrá a disposición de otras aplicaciones y entornos a través de los proveedores de acceso para .NET, Java ™, COM, y SOAP. Esto hace que sea mucho más fácil integrar IFS Applications con otra empresa, paquetes de software, tales como SAP ® u Oracle. (10)

Después de analizar como desarrollaron la interoperabilidad los sistemas ERP anteriores, los servicios web juegan un importante papel pues aportan interoperabilidad entre aplicaciones, independientemente de sus propiedades o de las plataformas sobre las que se instalen.

1.3 Modelo de desarrollo

Definen un conjunto de actividades, acciones, tareas, fundamentos y productos de trabajo que se requieren para desarrollar software de alta calidad. Debe llenar cada actividad del marco de trabajo con un acumulado de acciones de ingeniería de software, y definir cada acción en cuanto a un grupo de tareas que identifique el trabajo que deben completarse para alcanzar las metas de desarrollo.

1.3.1 Modelo de desarrollo propuesto

La propuesta de modelo de desarrollo fue elaborada por el equipo de producción del centro CEIGE teniendo en cuenta los principales riesgos con los que se cuentan en el proyecto.

Este modelo está orientado a la reutilización de componentes parametrizables, donde se simplifican las actividades y se eleva el nivel de especialización de los involucrados. Proporciona una guía para regir el proceso de desarrollo de software tecnológico, centrado en la arquitectura.

Dicho modelo está basado en principios, buenas prácticas propuestas y algunos elementos de las metodologías SCRUM y RUP, fue elaborado teniendo en cuenta las características especiales que presenta la UCI, por ejemplo: casi todos los proyectos están compuestos en su mayoría por estudiantes.

Tiene un valor social representativo, ya que implica mejoría en aspectos importantes como la planeación, formación y satisfacción del equipo de trabajo.

En general propone una solución sencilla y novedosa, que se centra en el desarrollo de componentes como base tecnológica, con una mayor calidad y en menor tiempo, para su posterior uso en la construcción de productos concretos; además propone dividir el trabajo y el equipo de desarrollo para lograr mayor especialización. Su buena aplicación proporcionará en gran medida la independencia tecnológica de los sistemas finales, pudiendo así ahorrar grandes sumas de dinero al país. (11)

1.3.2 Características

- **Centrado en la arquitectura**

La arquitectura determina la línea base, los elementos de software estructurales a partir de los elementos de la arquitectura de negocio. Interviene en la gestión de cambios y diseña la evolución e integración del producto. La arquitectura orienta las prioridades del desarrollo y resuelve las necesidades tecnológicas y de soporte para el desarrollo.

- **Orientado a componentes**

Las iteraciones son orientadas por el nivel de significancia arquitectónicas de los componentes, los mismos son abstracciones arquitectónicas de los procesos de negocio y requisitos asociados que modelan, el componente es la unidad de medición y ordenamiento de las iteraciones.

- **Iterativo e incremental**

Las iteraciones son planificadas y coordinadas con el equipo de arquitectura, los clientes y la alta gerencia. Cada iteración constituye el desarrollo de componentes, los cuales son integrados al término de la iteración, permitiendo de esta manera la evolución incremental del producto.

- **Ágil y adaptable al cambio**

El desarrollo de las partes formaliza solamente las características principales de la solución, priorizando los talleres y las comunicaciones entre las personas. Los clientes y funcionales están involucrados en el

proyecto y poseen parte de la responsabilidad del éxito del mismo. Los cambios son conciliados semanalmente, discutidos y aprobados.

1.4 Tecnologías empleadas

Debido a que este trabajo forma parte de un proceso productivo iniciado en el CEIGE las tecnologías que ahora serán descritas han sido definidas y adoptadas por el grupo de desarrolladores del centro, de acuerdo con las características del software.

1.4.1 Lenguajes de programación

JavaScript

Es un lenguaje basado en objetos y guiado por eventos, diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet. Los programas escritos con este lenguaje se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios, convirtiéndolo en un lenguaje interpretado. Su uso se basa fundamentalmente en la creación de efectos especiales en las páginas y la definición de interactividades con el usuario. Ventajas de JavaScript: Los programas escritos en este lenguaje no requieren de mucha memoria ni tiempo adicional de transmisión, por ser pequeños y compactos. 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. (12)

PHP

PHP, acrónimo de "Preprocesador de Hipertexto" por sus siglas en inglés, es un lenguaje interpretado de alto nivel embebido en páginas HTML. La mayoría de su sintaxis es similar a C, Java y Perl, con solamente un par de características PHP específicas. La meta de este lenguaje es permitir escribir a los creadores de páginas web, páginas dinámicas de una manera rápida y fácil. Una de sus características más potentes es su soporte para gran cantidad de bases de datos. Entre su soporte pueden mencionarse InterBase, mSQL, MySQL, Oracle, Informix, PostgreSQL, entre otras. PHP también ofrece la integración con varias bibliotecas externas. Como producto de código abierto, goza de la ayuda de un gran grupo de

programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparan rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP. (13)

1.4.2 Librerías y marcos de trabajo

Doctrine 0.11

Doctrine Framework 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 o superior con una base de datos con capas de abstracción incorporada. Brinda la posibilidad de exportar una base de datos existente a sus clases correspondientes y también a la inversa, es decir, convertir clases a tablas de una base de datos. Su principal ventaja radica en poder acceder a la base de datos utilizando la programación orientada a objetos debido a que doctrine utiliza el patrón Active Record para manejar la base de datos, tiene su propio lenguaje de consultas y trabaja de manera rápida y eficiente. (14)

ExtJS 2.2

Es una librería Java Script open-source de alto rendimiento para la creación y desarrollo de aplicaciones web dinámicas. Provee interfaces gráficas de usuario que brindan experiencias parecidas o iguales a las que se tienen con aplicaciones de escritorio. Permite la creación de aplicaciones complejas utilizando componentes predefinidos. Es extensible para la gran mayoría de los navegadores, evitando el tedioso problema de validar el código para cada uno de estos. Entre sus principales ventajas se encuentra el balance entre Cliente-Servidor, distribuyendo la carga de procesamiento en el último, y este al tener menor carga, maneja los clientes de manera más eficiente. La comunicación asíncrona permite el intercambio de información con el servidor sin necesidad de pedirle una acción al usuario, dando la libertad de cargar la información sin que este lo note. (15)

Zend Framework 1.8

Zend Framework no es más que un framework MVC (Modelo-Vista-Controlador) open-source para el desarrollo de aplicaciones Web con PHP. Brinda soluciones para construir sitios web modernos, robustos y seguros. Posee un bajo acoplamiento entre sus componentes, lo que posibilita la utilización de los mismos a conveniencia, aunque todos estos en conjunto conforman un potente y extensible framework

para aplicaciones web. Brinda una alta abstracción de bases de datos, haciendo extremadamente simple la interacción con estas, sin necesidad de escribir ninguna consulta SQL. Cuenta con módulos para el manejo de ficheros PDF, canales RSS, entre otros. Cuenta con clientes para el acceso a WS y robustas clases para la autenticación y el filtrado de entrada; completa documentación y test de alta calidad. (16)

1.5 Herramientas de desarrollo

Apache 2.2.9

Apache es el servidor web hecho por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Es una tecnología gratuita de código fuente abierta, se usa en una multitud de Sistemas Operativos, lo que lo hace prácticamente universal. Es un servidor altamente configurable de diseño modular. Actualmente existen muchos módulos que son adaptables a este. Te permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor y es posible configurarlo para que ejecute un determinado script cuando ocurra un error en concreto. (17)

NetBeans 6.9

El IDE NetBeans es un entorno integrado de desarrollo galardonado disponible para Windows, Mac, Linux y Solaris. El proyecto NetBeans consiste en un IDE de código abierto y una plataforma de aplicaciones que permiten a los desarrolladores crear rápidamente web, empresa, escritorio y aplicaciones móviles utilizando la plataforma Java, así como JavaFX, PHP, JavaScript y Ajax, Ruby y Ruby on Rails, Groovy y Grails, y C / C + +. (18)

PostgreSQL 8.3

PostgreSQL versión 8.3 es un gestor de bases de datos relacional orientado a objetos, libre y gratuito. Presenta las siguientes propiedades: (19)

- Atomicidad: Asegura la realización de una operación, por lo que ante un fallo del sistema, esta no queda a medias.

- Consistencia: Posibilita la ejecución de aquellas operaciones que no van a romper las reglas y directrices de integridad de la base de datos.

- Aislamiento: Mediante un sistema de acceso concurrente multiversión (Multiversion concurrency control, MVCC) asegura que una operación no pueda afectar a otras, de esta manera dos transacciones sobre la misma información no genera error.

Visual Paradigm 4.3

Visual Paradim For UML ayuda a los equipos de desarrollo de software para sobresalir todo el modelo de acumulación de trabajo así y desplegar el proceso de desarrollo de software, lo que permite maximizar y acelerar tanto las contribuciones individuales como las de equipo. Es una herramienta Case que soporta las últimas versiones del mismo, (Lenguaje de Modelado Unificado) y la Notación y Modelado de Procesos de negocios, además de un generador de mapeo de objetos-relacionales para los lenguajes de programación Java .NET y PHP. Se integra con las siguientes herramientas de java: Eclipse, WebSphere, JBuilder, NetBeans IDE, Oracle JDeveloper, BEA Weblogic. (20)

1.6 Conclusiones parciales

Una vez concluido el marco teórico de la investigación se arribó a las siguientes conclusiones:

- Una tecnología que ha tenido en los últimos tiempo un gran auge con respecto a la interoperabilidad entre sistemas informáticos son los servicios web, debido que es una solución sencilla, que evita en su mayor parte las dependencias de plataformas, fabricantes y además es una de las más usadas a nivel mundial.

- Hacer uso exclusivo de estándares abiertos, como la utilización de lenguaje de marcas extensibles XML, debido que da respuestas al problema de intercambio de información estructurada entre diferentes plataformas.

- Las herramientas y tecnologías descritas anteriormente resuelven el desarrollo de la solución.

Por lo antes expuesto se propone la implementación de un componente que resuelva la interoperabilidad del proceso inventario de Cedrux, entre los sistemas de gestión usados en Cuba.

CAPÍTULO 2: PROPUESTA DE SOLUCIÓN

El siguiente capítulo muestra los resultados obtenidos durante el proceso de desarrollo de la solución, así como algunos artefactos generados. En primer lugar son descritos los procesos de negocios, el modelo conceptual, los requisitos funcionales para entender lo que se quiere lograr y los requisitos no funcionales. Posteriormente se realiza una descripción del diseño para alcanzar las metas trazadas.

2.1 Descripción de los procesos de negocio

Un proceso de negocio es un grupo relacionado de etapas o actividades en la que especialistas utilizan recursos para crear valor para consumidores externos o internos. La descripción de los procesos de negocio crea una vista clara al paso a las actividades del análisis ya que posibilita una compresión visible de los procesos en cuestión y así contribuir a que los requisitos que se especifiquen satisfagan las necesidades de los usuarios.
A continuación son descritos cuatro procesos de negocio que deben ser asistidos por la solución. Teniendo en cuenta la opinión de especialistas y de los usuarios finales.

2.1.1 Descripción del proceso: Exportar documentos inventarios de apertura

Un documento de apertura refleja la cantidad y el importe de cada producto inventariado inicialmente en los almacenes y sirve de base para la comparación con los datos del Submayor de Inventario[6], con el fin de determinar las diferencias que resulten del conteo.

La solución debe permitir a los usuarios poder exportar documentos inventarios de apertura. El usuario solicita al proveedor un listado de los documentos inventarios de apertura que desea, especificando el depósito de los mismos, el proveedor por su parte revisa dicha solicitud recibida y crea el listado de los documentos inventarios de apertura dado el depósito y le envía al usuario el listado de documentos inventarios de apertura. El usuario notifica la entrega de los documentos inventarios de apertura recibidos y guarda dicho listado. En la figura 1 se muestra el diagrama del proceso.

[6] Submayor de Inventario: Controla las existencias en el almacén, de los productos adquiridos o producidos, en unidades físicas y valor, mediante el registro del movimiento de entradas, salidas y saldo en existencia de los mismos.

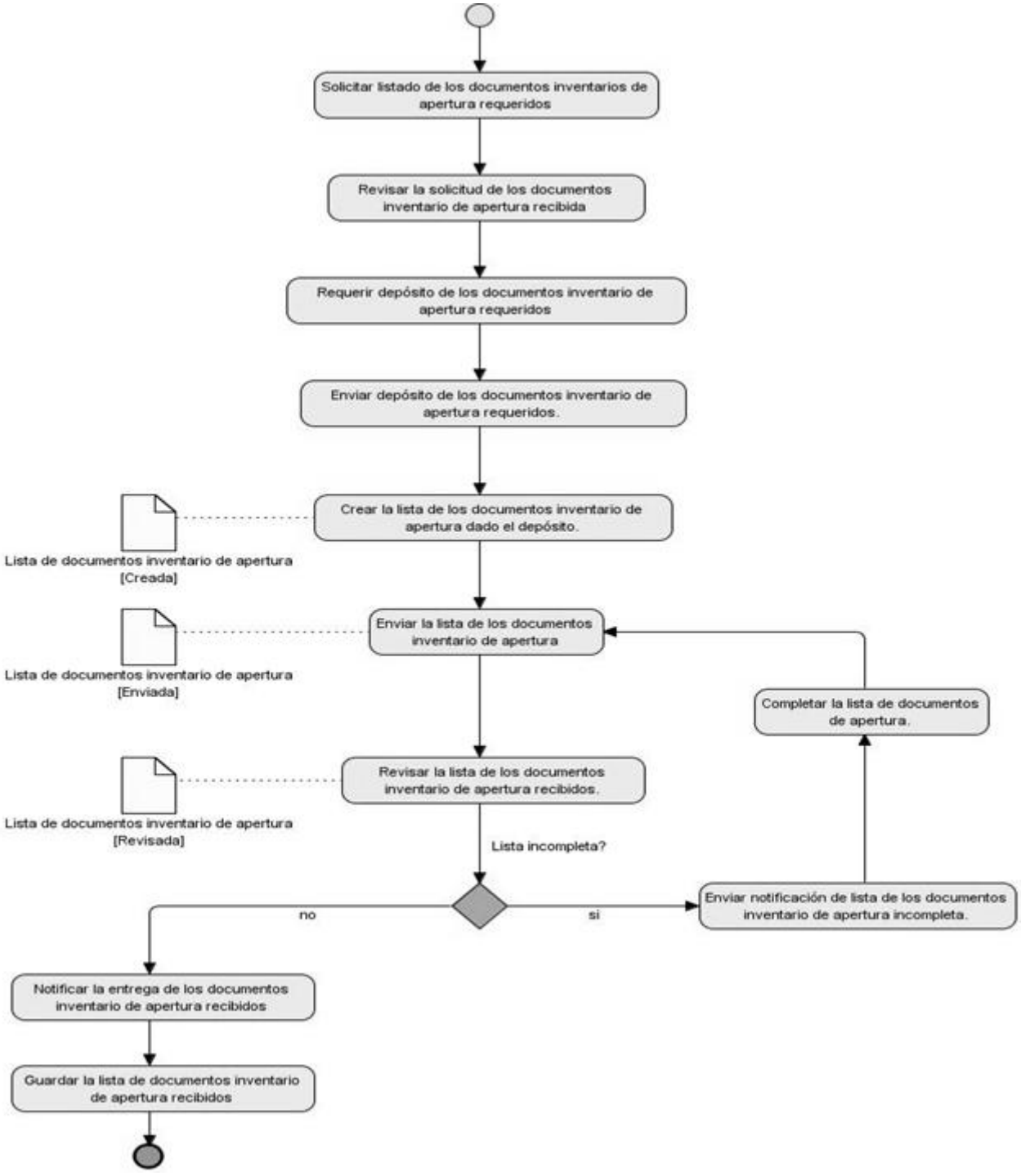


Figura 1 Proceso Exportar documentos inventarios de apertura.

2.1.2 Descripción del proceso: Exportar documentos inventarios de recepción

Un documento de recepción se encarga de formalizar la recepción en el almacén de los productos adquiridos de otras entidades.

La solución debe permitir a los usuarios poder exportar documentos inventarios de recepción. El usuario solicita al proveedor un listado de los documentos inventarios de recepción que desea, especificando el depósito de los mismos. El proveedor por su parte revisa dicha solicitud recibida y crea el listado de los documentos inventarios de recepción dado el depósito y le envía al usuario el listado de documentos inventarios de recepción. El usuario notifica la entrega de los documentos inventarios de recepción recibidos y guarda dicho listado. En la figura 2 se muestra el diagrama del proceso.

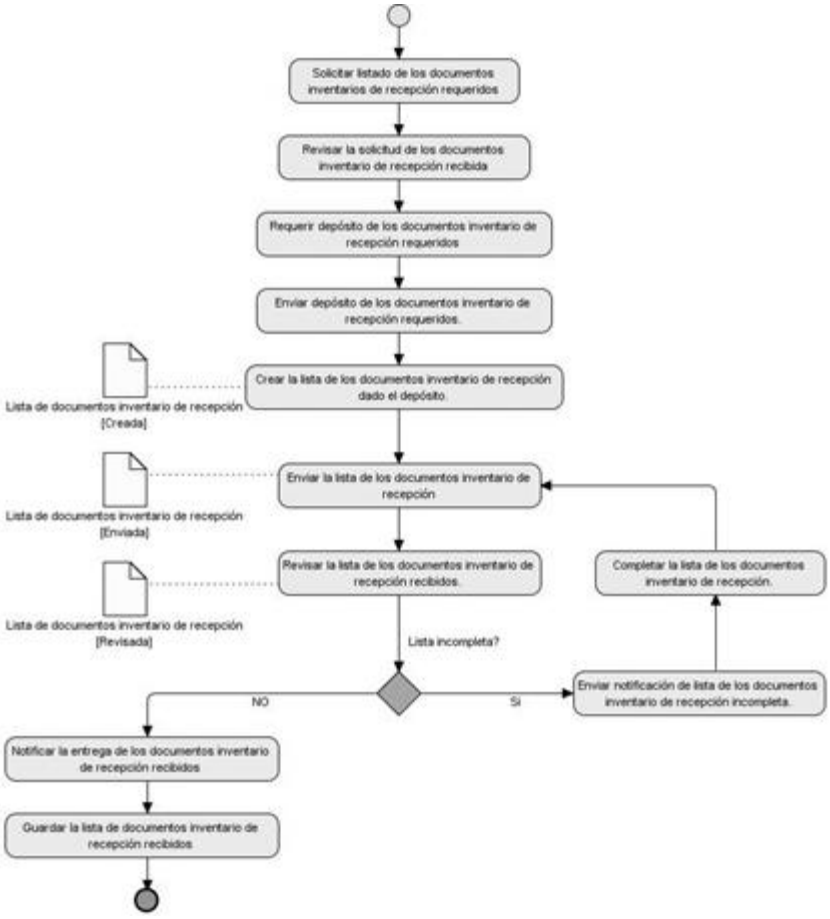


Figura 2 Proceso Exportar documentos inventarios de recepción.

2.1.3 Descripción del proceso: Exportar documentos inventarios de informe de diferencia

Un documento de informe de diferencia se encarga de notificar al suministrador o al transportador las reclamaciones originadas por errores en precios y cálculos del documento recepción y por averías, roturas o faltantes en ambos documentos.

La solución debe permitir a los usuarios poder exportar documentos inventarios de informe de diferencia. El usuario solicita al proveedor un listado de los documentos inventarios de informe de diferencia que desea, especificando el depósito de los mismos. El proveedor por su parte revisa dicha solicitud recibida y crea el listado de los documentos inventarios de informe de diferencia dado el depósito y le envía al usuario el listado de documentos inventarios de informe de diferencia. El usuario notifica la entrega de los documentos inventarios de informe de diferencia recibidos y guarda dicho listado. En la figura 3 se muestra el diagrama del proceso.

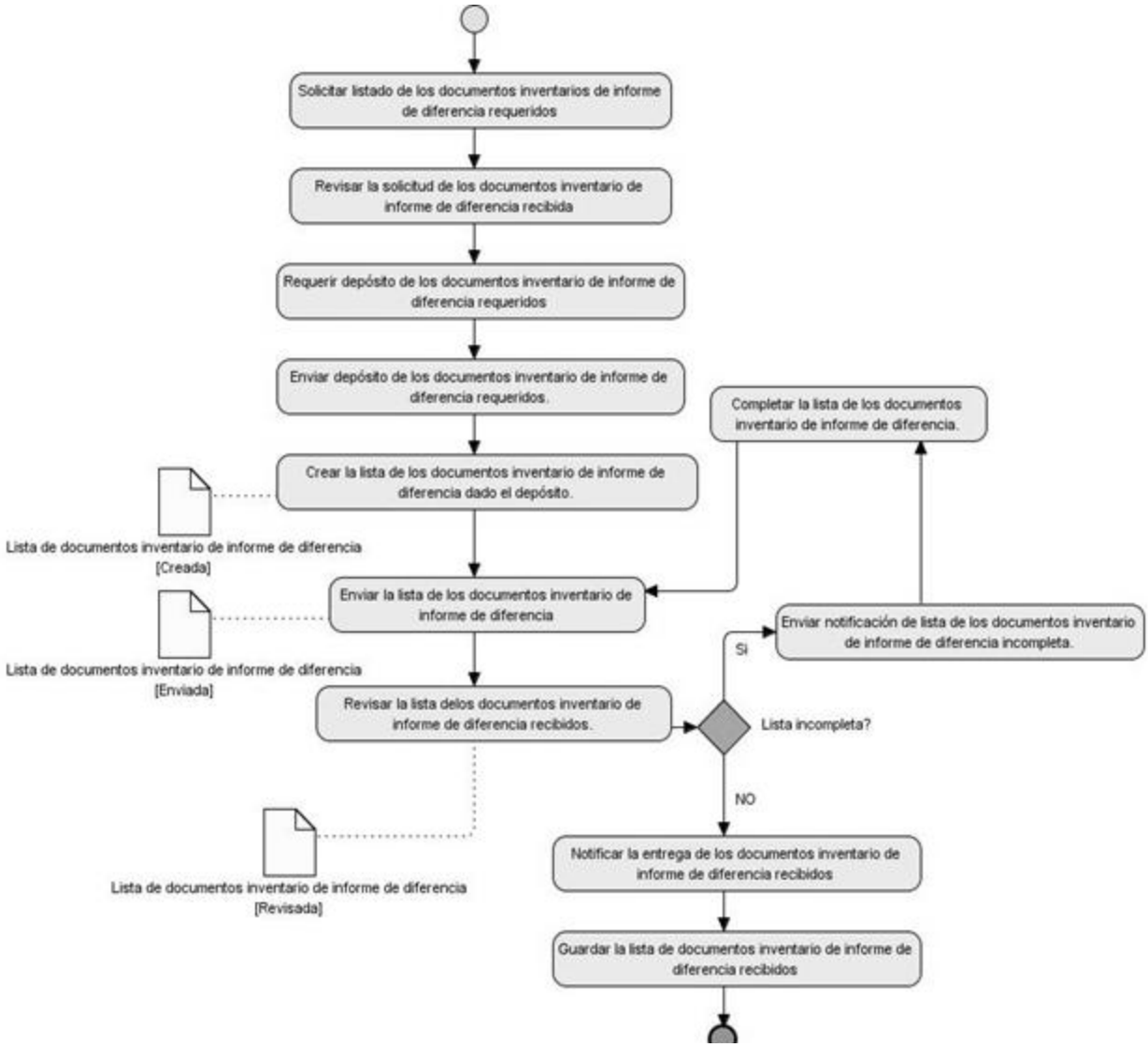


Figura 3 Proceso Exportar documentos inventarios de informe de diferencia.

2.1.4 Descripción del proceso: Exportar documentos de inventarios físico

Un documento de inventario físico se encarga de reflejar la cantidad y el importe de cada producto inventariado físicamente en los almacenes y sirve de base para la comparación con los datos del Submayor de Inventario, con el fin de determinar las diferencias que resulten del conteo.

La solución debe permitir a los usuarios poder exportar documentos de inventarios físicos. El usuario solicita al proveedor un listado de los documentos de inventarios físicos que desea, especificando el depósito de los mismos, el proveedor por su parte revisa dicha solicitud recibida y crea el listado de los documentos de inventarios físicos dado el depósito y le envía al usuario el listado de documentos de inventarios físicos. El usuario notifica la entrega de los documentos de inventarios físicos recibidos y guarda dicho listado. En la figura 4 se muestra el diagrama del proceso.

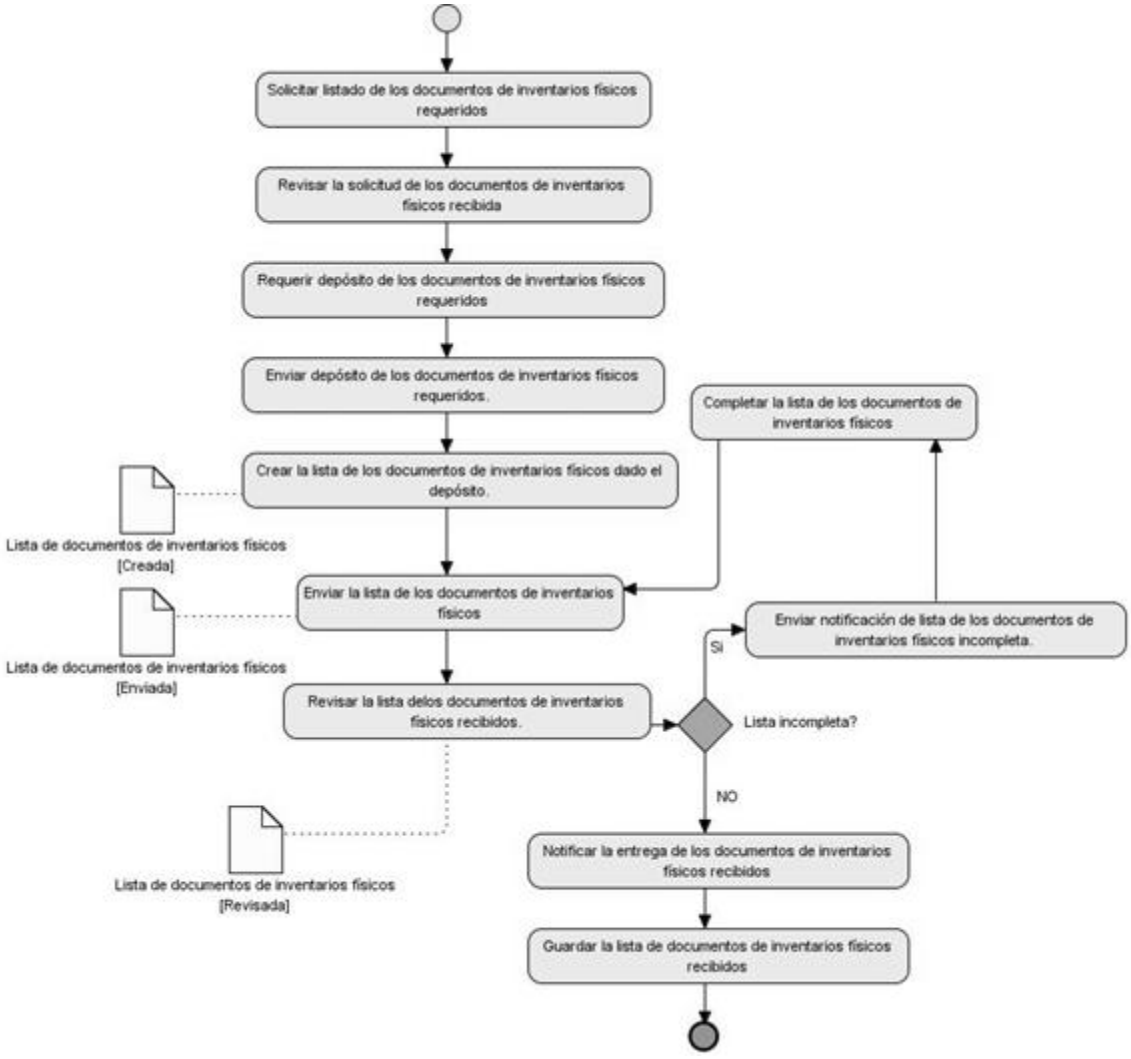


Figura 4 Proceso Exportar documentos de inventarios físico

2.2 Modelo conceptual

El modelo conceptual es una representación de conceptos en un dominio del problema. Este modelo muestra asociaciones entre conceptos y atributos de conceptos. Se puede ver como un modelo que comunica los términos importantes y cómo se relacionan entre sí. En la Figura 5 se representan los conceptos asociados al dominio del problema, y a continuación se describen los principales:

Se conoce al sistema de gestión Cedrux como un tipo de software que va a permitir a las empresas controlar la información que se genera en cada departamento y en cada nivel de la misma. Este sistema tiene un componente de interoperabilidad, en el mismo se desarrollan un conjunto de soluciones tecnológicas que facilitan el intercambio de información.

Los subsistemas son los diferentes módulos en los que ha sido dividido el proyecto para su construcción y a través de ellos se obtiene los datos a interoperar.

Los servicios son las funcionalidades que brinda el Componente de interoperabilidad. Dentro de estos servicios se encuentra exportar el cual toma los elementos seleccionados y los exporta.

Los procesos son la fuente que va a proporcionar los datos que se van a exportar.

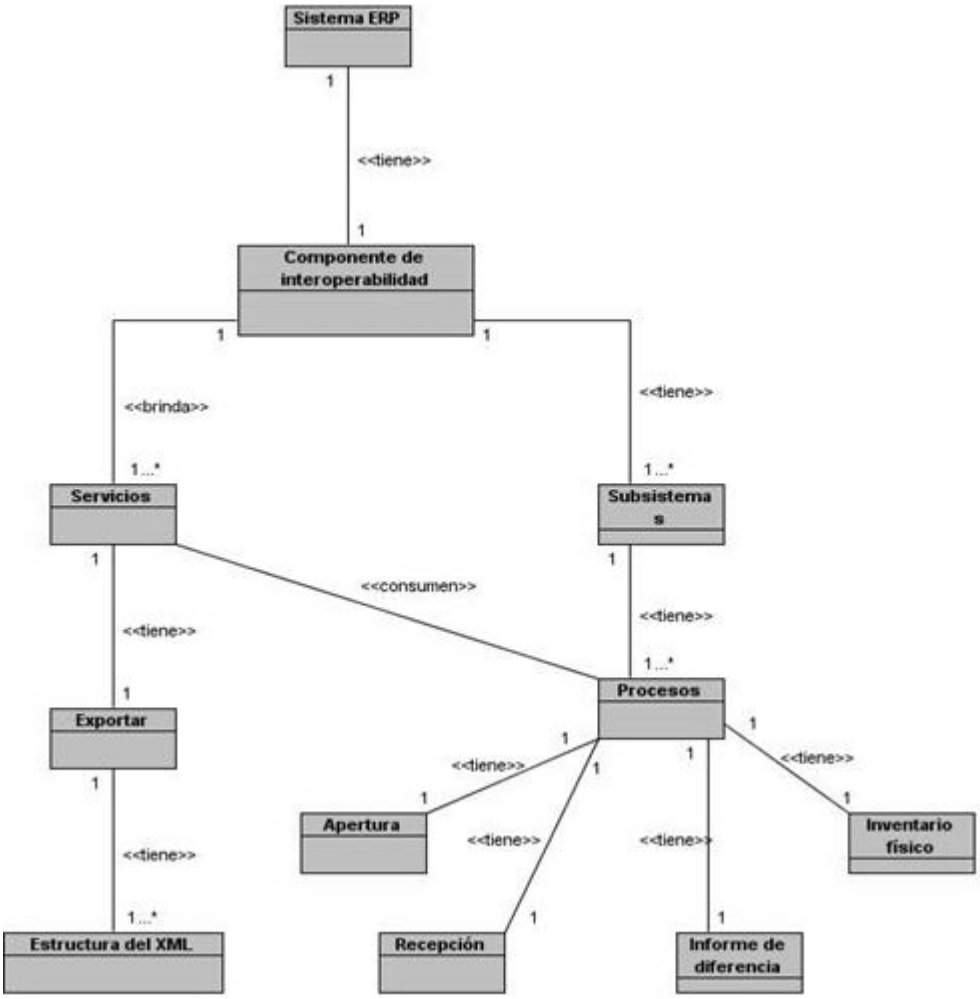


Figura 5 Modelo conceptual

2.3 Requisitos de software

Los requisitos cumplen un papel primordial en el proceso de producción de software, ya que se enfoca en un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta, el comportamiento del sistema; de esta manera, se pretende minimizar los problemas relacionados al desarrollo de sistemas.

Los requisitos de software se clasifican en funcionales y no funcionales. Los requisitos funcionales describen qué es lo que el sistema debe hacer para dar soporte a las funciones y objetivos del usuario.

Los requisitos no funcionales imponen restricciones de cómo los requisitos funcionales deben ser implementados. (21)

2.3.1 Requisitos funcionales

Los requisitos funcionales definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

De acuerdo a la experiencia de los analistas del centro CEIGE y con participación de los usuarios se identificaron un total de 12 requisitos funcionales que debe cumplir el componente, a continuación se listan dichos requisitos funcionales:

1. Gestionar apertura
2. Gestionar recepción
3. Gestionar inventario físico
4. Gestionar informe de diferencias
5. Configurar atributos de apertura
6. Configurar atributos de recepción
7. Configurar atributos de inventario físico
8. Configurar atributos de informe de diferencias
9. Descargar archivo de apertura
10. Descargar archivo de recepción
11. Descargar archivo de inventario físico
12. Descargar archivo de informe de diferencias

2.3.1.1 Requisito funcional: Gestionar apertura

Tabla 1 Especificación del requisito Gestionar apertura

Precondiciones	El usuario tiene que estar autenticado en el sistema y debe poseer permisos para ejecutar esta acción.
Descripción	
En este requisito comienza cuando el usuario selecciona el subsistema logística, que contiene el proceso Inventario y así sucesivamente hasta llegar al estándar definido para interoperar apertura, a continuación debe presionar la opción Exportar la cual muestra la interfaz correspondiente a Gestionar apertura donde se debe seleccionar el parámetro de	

búsqueda depósito, una vez seleccionado, se cargan todos los documentos apertura existentes para dicho parámetro, en caso de no existir documentos el sistema notificará al usuario. Una vez cargados todos los documentos el usuario debe seleccionar el/los que desea exportar.	
Flujo de eventos	
Flujo básico	
1	Seleccionar del subsistema Logística, el nodo Inventario, dar clic en Apertura, y seleccionar Apertura.
2	Dar clic en la opción Exportar.
3	Seleccionar los parámetros de búsqueda como son depósitos.
4	Seleccionar de la interfaz Gestionar apertura el/los documentos(s) de apertura que se desean exportar.
5	Se tiene que marcar al menos un documento apertura, el que el usuario quiera exportar, los cuales tienen número, año, estado, productos, fecha, creado por, aprobado por.
6	Dar clic en la opción Aceptar.
Flujos alternativos	
	Flujo alternativo 4.a No existen documentos apertura.
1	Después que el usuario selecciona los parámetros de búsqueda que son: depósitos, no se cargan de la base de datos la información requerida.
2	Volver al paso 3 del flujo básico.
	Pos-condiciones
1	**“**No hay documentos apertura asociados a los parámetros definidos.**”**
	Flujo alternativo *.a El usuario cancela la acción.
1	Concluye el requisito.
	Pos-condiciones
	N/A
Validaciones	
1	Se validan los datos según lo establecido en el Modelo conceptual CSG-CNP Modelo conceptual Componente de Interoperabilidad.
Conceptos	Visible en la interfaz: número, año, estado, productos, fecha, creado por, aprobado por, depósito.
Requisitos especiales	N/A
Asuntos pendientes	N/A

2.3.1.2 Requisito funcional: Gestionar recepción

Tabla 2 Especificación del requisito Gestionar recepción

Precondiciones	El usuario tiene que estar autenticado en el sistema y debe poseer permisos para ejecutar esta acción.
Descripción	
En este requisito comienza cuando el usuario selecciona el subsistema logística, que contiene el proceso Inventario y así sucesivamente hasta llegar al estándar definido para interoperar recepción, a continuación debe presionar la opción Exportar la cual muestra la interfaz correspondiente a Gestionar recepción donde se debe seleccionar el parámetro de búsqueda depósito, una vez seleccionado, se cargan todos los documentos recepción existentes para dicho parámetro, en caso de no existir documentos el sistema notificará al usuario. Una vez cargados todos los documentos el usuario debe seleccionar el/los que desea exportar.	
Flujo de eventos	
Flujo básico	
1	Seleccionar del subsistema Logística, el nodo Inventario, dar clic en Recepción, y seleccionar Recepción.
2	Dar clic en la opción Exportar.
3	Seleccionar los parámetros de búsqueda como son depósitos.
4	Seleccionar de la interfaz Gestionar recepción el/los documentos(s) de recepción que se desean exportar.
5	Se tiene que marcar al menos un documento recepción, el que el usuario quiera exportar, los cuales tienen número, año, estado, productos, fecha, creado por.
6	Dar clic en la opción Aceptar.
Flujos alternativos	
	Flujo alternativo 4.a No existen documentos de recepción.
1	Después que el usuario selecciona los parámetros de búsqueda que son: depósitos, no se cargan de la base de datos la información requerida.
2	Volver al paso 3 del flujo básico.
	Pos-condiciones
1	**“**No hay documentos de recepción asociados a los parámetros definidos.**”**
	Flujo alternativo *.a El usuario cancela la acción.
1	Concluye el requisito.
	Pos-condiciones
	N/A
Validaciones	
1	Se validan los datos según lo establecido en el Modelo conceptual CSG-CNP Modelo conceptual Componente de Interoperabilidad.
Conceptos	Visibles en la interfaz: depósitos, número, año, estado, productos, fecha, creado por.

Requisitos especiales	N/A
Asuntos pendientes	N/A

2.3.1.3 Requisito funcional: Gestionar inventario físico

Tabla 3 Especificación del requisito Gestionar inventario físico

Precondiciones	El usuario tiene que estar autenticado en el sistema y debe poseer permisos para ejecutar esta acción.
Descripción	
En este requisito comienza cuando el usuario selecciona el subsistema logística, que contiene el proceso Inventario y así sucesivamente hasta llegar al estándar definido para interoperar inventario físico, a continuación debe presionar la opción Exportar la cual muestra la interfaz correspondiente a Gestionar inventario físico donde se debe seleccionar el parámetro de búsqueda depósito, una vez seleccionado, se cargan todos los documentos de inventario físico existentes para dicho parámetro, en caso de no existir documentos el sistema notificará al usuario. Una vez cargados todos los documentos el usuario debe seleccionar el/los que desea exportar.	
Flujo de eventos	
Flujo básico	
1	Seleccionar del subsistema Logística, el nodo Inventario, y seleccionar Inventario.
2	Dar clic en la opción Exportar.
3	Seleccionar los parámetros de búsqueda como son depósitos.
4	Seleccionar de la interfaz Gestionar inventario físico el/los documentos(s) de inventario físico que se desean exportar.
5	Se tiene que marcar al menos un documento inventario físico, el que el usuario quiera exportar, los cuales tienen número, año, estado, tipo, productos, creado por, aprobado por.
6	Dar clic en la opción Aceptar.
Flujos alternativos	
	Flujo alternativo 4.a No existen documentos inventarios físicos.
1	Después que el usuario selecciona los parámetros de búsqueda que son: depósitos, no se cargan de la base de datos la información requerida.
2	Volver al paso 3 del flujo básico.
	Pos-condiciones
1	**“**No hay inventarios físicos asociados a los parámetros definidos.**”**
	Flujo alternativo *.a El usuario cancela la acción.
1	Concluye el requisito.

	Pos-condiciones
	N/A
Validaciones	
	Se validan los datos según lo establecido en el Modelo conceptual CSG-CNP Modelo conceptual Componente de Interoperabilidad.
Conceptos	Visible en la interfaz: número, año, estado, tipo, productos, creado por, aprobado por.
Requisitos especiales	N/A
Asuntos pendientes	N/A

2.3.1.4 Requisito funcional: Gestionar informe de diferencias

Tabla 4 Especificación del requisito Gestionar informe de diferencias

Precondiciones	El usuario tiene que estar autenticado en el sistema y debe poseer permisos para ejecutar esta acción.
Descripción	
En este requisito comienza cuando el usuario selecciona el subsistema logística, que contiene el proceso Inventario y así sucesivamente hasta llegar al estándar definido para interoperar informe de diferencias, a continuación debe presionar la opción Exportar la cual muestra la interfaz correspondiente a Gestionar informe de diferencias donde se debe seleccionar el parámetro de búsqueda depósito, una vez seleccionado, se cargan todos los documentos informe de diferencias existentes para dicho parámetro, en caso de no existir documentos el sistema notificará al usuario. Una vez cargados todos los documentos el usuario debe seleccionar el/los que desea exportar.	
Flujo de eventos	
Flujo básico	
1	Seleccionar del subsistema Logística, el nodo Inventario, dar clic en Recepción, y seleccionar Diferencias.
2	Dar clic en la opción Exportar.
3	Seleccionar los parámetros de búsqueda como son depósitos
4	Seleccionar de la interfaz Gestionar Informe de diferencias el/los documentos(s) de Informe de diferencias que se desean exportar.
5	Se tiene que marcar al menos un documento informe de diferencias, el que el usuario quiera exportar, los cuales tienen número, año, estado, productos, fecha.
6	Dar clic en la opción Aceptar.
Flujos alternativos	
	Flujo alternativo 4.a No existen documentos inventarios físicos.

1	Después que el usuario selecciona los parámetros de búsqueda que son: depósitos, no se cargan de la base de datos la información requerida.
2	Volver al paso 3 del flujo básico.
	Pos-condiciones
1	**"**No hay informe de diferencias asociados a los parámetros definidos.**"**
	Flujo alternativo *.a El usuario cancela la acción.
1	Concluye el requisito.
	Pos-condiciones
	N/A
Validaciones	
	Se validan los datos según lo establecido en el Modelo conceptual CSG-CNP Modelo conceptual Componente de Interoperabilidad.
Conceptos	Visible en la interfaz: número, año, estado, productos, fecha, depósito.
Requisitos especiales	N/A
Asuntos pendientes	N/A

2.3.1.5 Requisito funcional: Configurar atributos de apertura

Tabla 5 Especificación del requisito Configurar atributos de apertura

Precondiciones	El usuario haya realizado correctamente el Flujo de eventos del requisito Gestionar apertura.
Descripción	
En este requisito el usuario selecciona los atributos de los documentos inventarios de apertura que desea exportar. El sistema permite desmarcar los atributos que no desee exportar. Para finalizar se presiona el botón Aceptar.	
Flujo de eventos	
Flujo básico	
1	Seleccionar los atributos del XML a exportar.
2	Dar clic en el botón Aceptar.
Flujos alternativos	
Flujo alternativo 1.a No selecciona atributos.	
1	El usuario no selecciona al menos un atributo del XML.
2	Se muestra un mensaje informativo.
3	Volver al paso 1 del flujo básico.
Pos-condiciones	

1	Debe seleccionar al menos un parámetro.
Flujo alternativo *.a El usuario cancela la acción	
1	Concluye el requisito.
Pos-condiciones	
1	N/A
Validaciones	
	Se validan los datos según lo establecido en el Modelo conceptual <<Paquete de herramientas de soporte al desarrollo de aplicaciones web de gestión.doc>>
Conceptos	Visible en la interfaz:
Requisitos especiales	N/A.
Asuntos pendientes	N/A.

2.3.1.6 Requisito funcional: Configurar atributos de recepción

Tabla 6 Especificación del requisito Configurar atributos de recepción

Precondiciones	El usuario haya realizado correctamente el Flujo de eventos del requisito Gestionar recepción.
Descripción	
En este requisito el usuario selecciona los atributos de los documentos de recepción que desea exportar. El sistema permite desmarcar los atributos que no desee exportar. Para finalizar se presiona el botón Aceptar.	
Flujo de eventos	
Flujo básico	
1	Seleccionar los atributos del XML a exportar
2	Dar clic en el botón Aceptar.
Flujos alternativos	
Flujo alternativo 1.a No selecciona atributos.	
1	El usuario no selecciona al menos un atributo del XML.
2	Se muestra un mensaje informativo.
3	Volver al paso 1 del flujo básico.
Pos-condiciones	
1	N/A
Flujo alternativo *.a El usuario cancela la acción	
1	Concluye el requisito.
Pos-condiciones	
1	No se realiza la acción de exportar recepción
Validaciones	

1	Se validan los datos según lo establecido en el Modelo conceptual <<Paquete de herramientas de soporte al desarrollo de aplicaciones web de gestión.doc>>	
Relaciones	**Requisitos Incluidos**	Aplicar técnica.
	Extensiones	N/A.
Atributos	Visibles en la interfaz:	
Requisitos especiales	N/A.	
Asuntos pendientes	N/A.	

2.3.1.7 Requisito funcional: Configurar atributos de informe de diferencias

Tabla 7 Especificación del requisito Configurar atributos de informe de diferencias

Precondiciones	El usuario haya realizado correctamente el Flujo de eventos del requisito Gestionar informe de diferencias.	
Descripción		
En este requisito el usuario selecciona los atributos de los documentos de informe de diferencias que desea exportar. El sistema permite desmarcar los atributos que no desee exportar. Para finalizar se presiona el botón Aceptar.		
Flujo de eventos		
Flujo básico		
1	Seleccionar los atributos del XML a exportar	
2	Dar clic en el botón Aceptar.	
Flujos alternativos		
Flujo alternativo 1.a No selecciona atributos.		
1	El usuario no selecciona al menos un atributo del XML.	
2	Se muestra un mensaje informativo.	
3	Volver al paso 1 del flujo básico.	
Pos-condiciones		
N/A		
Flujo alternativo *.a El usuario cancela la acción		
1	Concluye el requisito.	
Pos-condiciones		
1	No se realiza la acción de exportar informes de diferencias.	
Validaciones		
1	Se validan los datos según lo establecido en el Modelo conceptual <<Paquete de herramientas de soporte al desarrollo de aplicaciones web de gestión.doc>>	
Relaciones	**Requisitos Incluidos**	Aplicar técnica.

	Extensiones	N/A.
Atributos	Visibles en la interfaz:	
Requisitos especiales	N/A.	
Asuntos pendientes	N/A.	

2.3.1.8 Requisito funcional: Configurar atributos de inventario físico

Tabla 8 Especificación del requisito Configurar atributos de inventario físico

Precondiciones	El usuario haya realizado correctamente el Flujo de eventos del requisito Gestionar inventario físico.	
Descripción		
En este requisito el usuario selecciona los atributos de los documentos de inventario físico que desea exportar. El sistema permite desmarcar los atributos que no desee exportar. Para finalizar se presiona el botón Aceptar.		
Flujo de eventos		
Flujo básico		
1	Seleccionar los atributos del XML a exportar	
2	Dar clic en el botón Aceptar.	
Flujos alternativos		
Flujo alternativo 1.a No selecciona atributos.		
1	El usuario no selecciona al menos un atributo del XML.	
2	Se muestra un mensaje informativo.	
3	Volver al paso 1 del flujo básico.	
Pos-condiciones		
N/A		
Flujo alternativo *.a El usuario cancela la acción		
1	Concluye el requisito.	
Pos-condiciones		
1	No se realiza la acción de exportar inventario físico.	
Validaciones		
1	Se validan los datos según lo establecido en el Modelo conceptual <<Paquete de herramientas de soporte al desarrollo de aplicaciones web de gestión.doc>>	
Relaciones	**Requisitos Incluidos**	Aplicar técnica.
	Extensiones	N/A.
Atributos	Visibles en la interfaz:	

Requisitos especiales	N/A.
Asuntos pendientes	N/A.

2.3.1.9 Requisito funcional: Descargar archivo de apertura

Tabla 9 Especificación del requisito Descargar archivo de apertura

Precondiciones	El usuario tiene que haber cumplido con los parámetros del Flujo de eventos Configurar atributos de apertura.	
Descripción		
En este requisito el usuario busca una dirección en memoria para guardar los documentos de apertura que se desean exportar.		
Flujo de eventos		
Flujo básico		
1	El usuario selecciona el lugar donde será guardado el archivo.	
2	Dar clic en la opción Aceptar.	
Flujos alternativos		
Flujo alternativo *.a El usuario cancela la acción		
1	Concluye el requisito.	
Pos-condiciones		
1	No se guarda el compactado con los XML exportados.	
Validaciones		
1	Se validan los datos según lo establecido en el Modelo conceptual CSG-CNP Modelo conceptual Componente de Interoperabilidad.	
Relaciones	**Requisitos Incluidos**	Aplicar técnica.
	Extensiones	N/A.
Atributos	Visibles en la interfaz:	
Requisitos especiales	N/A.	
Asuntos pendientes	N/A.	

2.3.1.10 Requisito funcional: Descargar archivo de recepción

Tabla 10 Especificación del requisito Descargar archivo de recepción

Precondiciones	El usuario tiene que haber cumplido con los parámetros del Flujo de eventos Configurar atributos de recepción.

Descripción		
En este requisito el usuario busca una dirección en memoria para guardar los documentos de recepción que se desean exportar.		
Flujo de eventos		
Flujo básico		
1	El usuario selecciona el lugar donde será guardado el archivo.	
2	Dar clic en la opción Aceptar.	
Flujos alternativos		
Flujo alternativo *.a El usuario cancela la acción		
1	Concluye el requisito.	
Pos-condiciones		
1	No se guarda el compactado con los XML exportados.	
Validaciones		
	Se validan los datos según lo establecido en el Modelo conceptual CSG-CNP Modelo conceptual Componente de Interoperabilidad.	
Relaciones	**Requisitos Incluidos**	Aplicar técnica.
	Extensiones	N/A.
Atributos	Visibles en la interfaz:	
Requisitos especiales	N/A.	
Asuntos pendientes	N/A.	

2.3.1.11 Requisito funcional: Descargar archivo de inventario físico

Tabla 11 Especificación del requisito Descargar archivo de inventario físico

Precondiciones	El usuario tiene que haber cumplido con los parámetros del Flujo de eventos Configurar atributos de inventario físico.	
Descripción		
En este requisito el usuario busca una dirección en memoria para guardar los documentos de inventario físico que se desean exportar.		
Flujo de eventos		
Flujo básico		
1	El usuario selecciona el lugar donde será guardado el archivo.	
2	Dar clic en la opción Aceptar.	
Flujos alternativos		
Flujo alternativo *.a El usuario cancela la acción		
1	Concluye el requisito.	
Pos-condiciones		

1	No se guarda el compactado con los XML exportados.	
Validaciones		
1	Se validan los datos según lo establecido en el Modelo conceptual CSG-CNP Modelo conceptual Componente de Interoperabilidad.	
Relaciones	**Requisitos Incluidos**	Aplicar técnica.
	Extensiones	N/A.
Atributos	Visibles en la interfaz:	
Requisitos especiales	N/A.	
Asuntos pendientes	N/A.	

2.3.1.12 Requisito funcional: Descargar archivo de informe de diferencia

Tabla 12 Especificación del requisito Descargar archivo de informe de diferencia

Precondiciones	El usuario tiene que haber cumplido con los parámetros del Flujo de eventos Configurar atributos de informe de diferencias.	
Descripción		
En este requisito el usuario busca una dirección en memoria para guardar los documentos de informe de diferencia que se desean exportar.		
Flujo de eventos		
Flujo básico		
1	El usuario selecciona el lugar donde será guardado el archivo.	
2	Dar clic en la opción Aceptar.	
Flujos alternativos		
Flujo alternativo *.a El usuario cancela la acción		
1	Concluye el requisito.	
Pos-condiciones		
1	No se guarda el compactado con los XML exportados.	
Validaciones		
1	Se validan los datos según lo establecido en el Modelo conceptual CSG-CNP Modelo conceptual Componente de Interoperabilidad.	
Relaciones	**Requisitos Incluidos**	Aplicar técnica.
	Extensiones	N/A.
Atributos	Visibles en la interfaz:	
Requisitos especiales	N/A.	

Asuntos pendientes	N/A.

2.3.2 Requisitos no funcionales

El presente trabajo forma parte de un proceso productivo iniciado por el CEIGE y los resultados que se obtengan formarán parte del marco de trabajo desarrollado en el mismo. De esta manera los requisitos no funcionales a los que se debe acoger la aplicación a desarrollar son los que fueron establecidos por el centro al inicio del proceso de desarrollo, a continuación son descritos algunos de los más importantes.

2.3.2.1 Software

Para el cliente:

- Navegador Mozilla Firefox 3.0 o superior.
- Sistema operativo Windows 98 o superior o Linux.

Para el servidor:

- Sistema operativo Linux en cualquiera de sus distribuciones.
- Un servidor Apache 2.0 o superior con módulo PHP 5.0 disponible. Este debe estar configurado con la extensión “pgsql” incluida.
- Un servidor de base de datos PostgreSQL 8.3.

2.3.2.2 Rendimiento

Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos, no mayores de 5 segundos para las actualizaciones y 20 para las recuperaciones.

2.3.2.3 Seguridad

Autenticación y Autorización (Contraseña de acceso). Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos. La atención al sistema incluyendo el mantenimiento de las bases de datos, así como la salva de la información, se realizará de forma centralizada por el administrador.

2.3.2.4 Hardware

Para el servidor:

- Requerimientos mínimos: Procesador Pentium IV a 2GHz de velocidad de procesamiento y 1Gb de memoria RAM.
- Al menos 40Gb de espacio libre en disco duro.
- Tarjeta de red.

Para el cliente:

- Requerimientos mínimos: Procesador Pentium III a 1GHz con 256Mb de memoria RAM.
- Tarjeta de red.

2.4 Prototipo de interfaces de usuarios

Para un mejor entendimiento y desarrollo de la implementación de este componente se comienza a definir y diseñar las principales vistas que van a interactuar con el usuario final, en busca de aminorar la complejidad y abstracción que puede conllevar la implementación de un software para los desarrolladores y también con el objetivo de evaluar el nivel de satisfacción del cliente y garantizar el entendimiento con este. A continuación se muestra el prototipo desarrollado para la interfaz principal (Figura 6). Los demás prototipos desarrollados se encuentran en el Anexo 1.

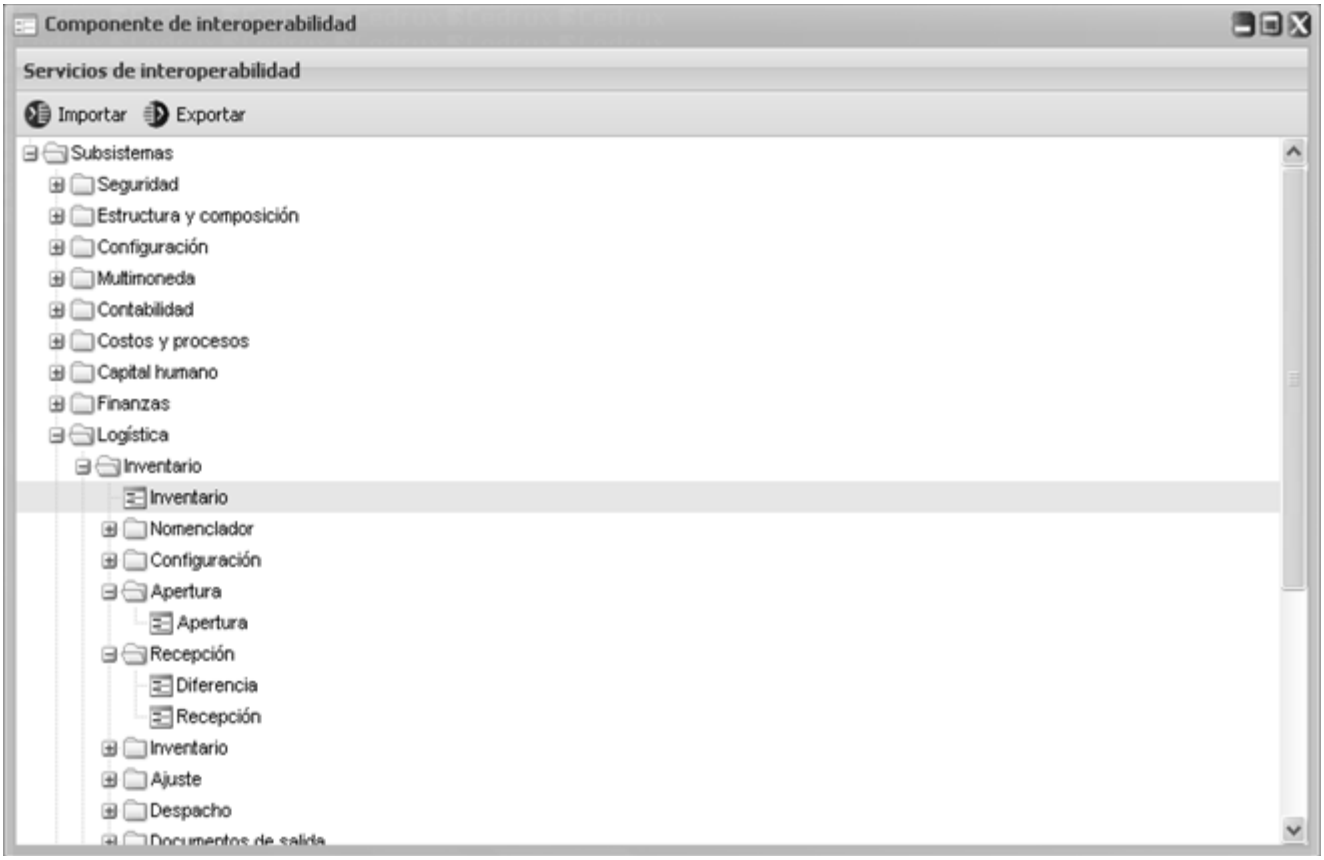


Figura 6 Interfaz principal.

2.5 Diagrama de clases del diseño

El diagrama de clases del diseño describe la estructura del sistema, mostrando sus clases, asociaciones, atributos, métodos y sus relaciones entre ellos. A continuación en la figura 7 se muestra el diagrama de clases del diseño con estereotipos web:

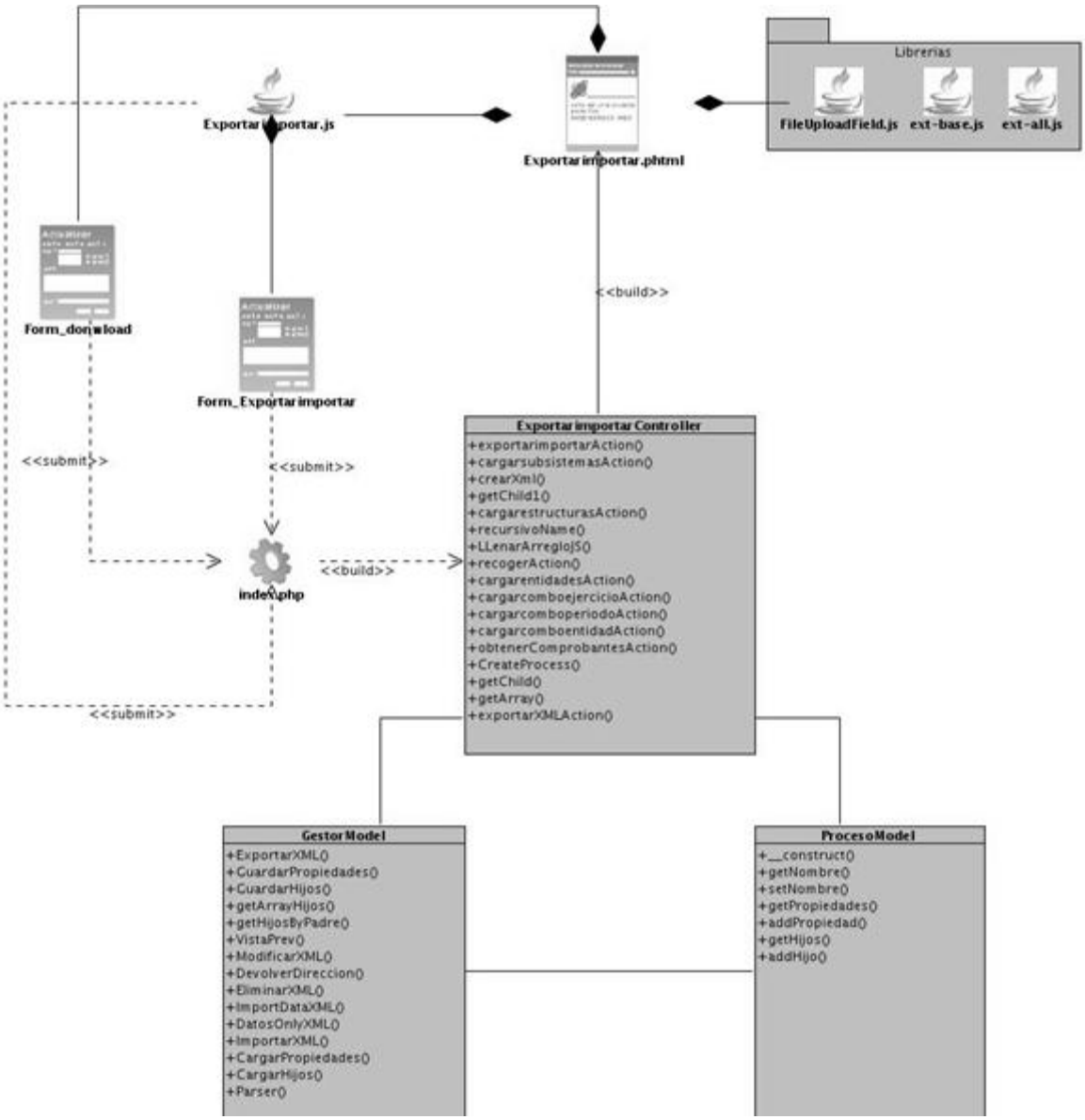


Figura 7 Diagrama de clases del diseño con estereotipos web

2.6 Patrones utilizados

2.6.1 Patrón arquitectónico Modelo-Vista-Controlador (MVC)

El Modelo Vista Controlador (MVC) es un patrón arquitectónico que tiene como principal característica separar los datos de una aplicación en tres módulos identificables y con funcionalidades bien definidas: el Modelo, las Vistas y el Controlador, permitiendo flexibilidad y facilidad a la hora de hacer futuros cambios.

El **modelo** es un conjunto de clases que representan la información del mundo real que el sistema debe procesar. (22) En el mismo se encuentran todas las funciones para consultar, insertar y actualizar información de la base de datos.

Las **vistas** son el conjunto de clases que se encargan de presentar al usuario la información contenida en el modelo. Una vista está asociada a un modelo, pudiendo existir varias vistas asociadas al mismo modelo. Estas pueden ser una página web o una parte de una página y obtiene del modelo solamente la información que necesita para desplegar y se actualiza cada vez que el modelo del dominio cambia por medio de notificaciones.

El **controlador** es un objeto que se encarga de dirigir el flujo del control de la aplicación debido a mensajes externos, como datos introducidos por el usuario u opciones del menú seleccionadas por él. A partir de estos mensajes, el controlador se encarga de modificar el modelo o de abrir y cerrar vistas. El controlador tiene acceso al modelo y a las vistas, pero las vistas y el modelo no conocen de la existencia del controlador. (23)

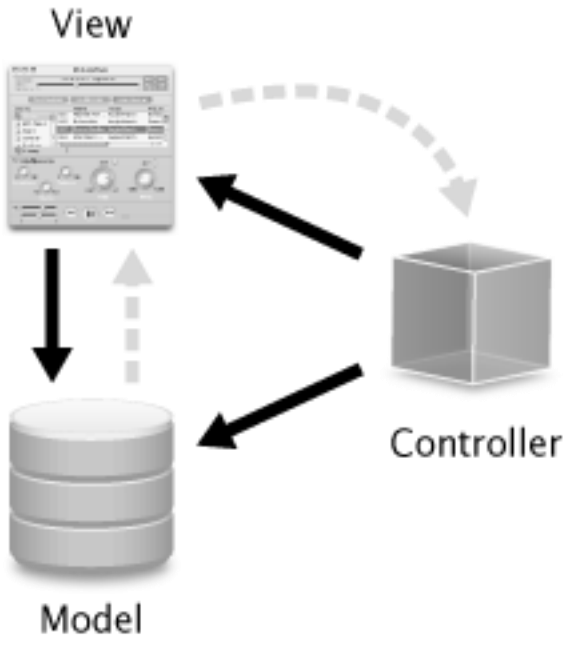


Figura 8 Patrón arquitectónico Modelo Vista-Controlador

2.7 Conclusiones parciales

Con la realización de este capítulo se lograron los objetivos trazados, al realizar un profundo análisis y diseño de la solución. En el mismo se exponen los diferentes artefactos realizados como son la descripción de los procesos de negocios, el diagrama de despliegue, la descripción de los requisitos funcionales, no funcionales, y el diagrama de clases del diseño, además de mostrar los prototipos de interfaces. Una vez concluida esta fase de la solución se procederá a la realización de los flujos de implementación y prueba de la misma.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

En el siguiente capítulo se muestran en primer lugar los estándares de codificación, así como los diferentes artefactos que se desarrollaron en la fase de implementación, como el diagrama de componente y el diagrama de despliegue. Y se especifica el conjunto de validaciones y pruebas que evalúan la calidad del componente desarrollado.

3.1 Estándares de codificación

Se definen como estándares de codificación a un estilo de programación en un proyecto determinado permitiendo que todos los participantes lo puedan entender en menos tiempo y que el código en consecuencia sea mantenido.
Debido a el numeroso personal involucrado en la implementación del sistema CEDRUX, su complejidad y el alto nivel de integración, el grupo arquitectónico del proyecto definió normas de codificación con el fin de obtener un estándar en la implementación por el equipo de desarrollo que permitiera asegurar la calidad del software y de esta forma obtener un código más legible y reutilizable. A continuación se describen estos estándares empleados en la implementación del componente interoperabilidad.

3.1.1 PascalCasing

El estándar PascalCasing establece que los identificadores, nombres de clases, variables, métodos o funciones están compuestos por múltiples palabras juntas, iniciando cada palabra con letra mayúscula.
La nomenclatura de las clases del componente interoperabilidad se realizó sobre la base de este estándar, usando palabras compuestas sugerentes acordes al propósito de la misma.

Nomenclatura de clases según su tipo

- **Controllers:** clases controladoras del negocio.

El nombre de la clase controladora debe estar estructurado por el nombre propio de la misma en mayúsculas seguido por la palabra Controller y heredar siempre de la super clase del framework ZendExt_Controller_Secure. Ejemplo: ExportarimportarController.

- **Clases de los modelos**

- **Business:** Clases modelo del negocio.

Las clases modelo tendrán por identificador el nombre de la tabla en la que trabajan seguido por la palabra Model y heredarán de la super clase del framework Zend_Ext llamada ZendExt_Model. Ejemplo: NomProcesoModel.

- **Domain:** clases entidades del dominio.

Los archivos situados en el domain tienen el mismo nombre de las tablas que representan, definiéndolas como clases php, pueden incluir los prefijos Dat o Nom para diferenciar entre sus usos. Ejemplo: NomProceso.

- **Generated:** Clases bases del dominio

Las clases que se encuentran dentro de Generated comienza su nombre con la palabra: “Base”, seguido del nombre de la tabla en la Base de Datos. Ejemplo: BaseNomProceso.

- **Validators:** Clases validadoras.

El nombre de la clase validadoras se especifica con nombres compuestos con el distintivo que terminan en Validator. Ejemplo: ProcesoValidator.

- **Services:** Clases que ofrecen los servicios de los componentes.

Estas clases, de acuerdo con las operaciones que realizan y prestaciones que brindan, se definen con calificativos sugerentes, agregándoles la terminación Service. Ejemplo: ProcesoService.

3.1.2 CamelCasing

El estándar CamelCasing es parecido al PascalCasing con la particularidad de que la letra inicial del identificador no comienza con mayúscula. Esta notación se utilizó para el nombre de funciones y atributos.

- **Nomenclatura de las funciones**

El identificativo a emplear para las funciones o métodos se escribe con la primera palabra en minúscula utilizando la notación CamelCasing y nombres que deduzcan su propósito.
Ejemplo: cargarSubsistemas.
Los denominadores de las acciones de las clases controladoras tienen la peculiaridad de ir seguidos por la palabra "Action". Ejemplo: cargarsubsistemasAction.

- **Nomenclatura de los atributos**

Se definió que el nombre a emplear para las variables se escribe en minúscula y en caso de ser un nombre compuesto se escriben ambas palabras en minúscula separadas por un guión bajo. Ejemplo: id_subsistema y nombre_subsistema.

3.1.3 Notación húngara

Esta convención, también conocida como notación: REDDICK por el nombre de su creador, se basa en definir prefijos para cada tipo de datos según el ámbito de las variables con el fin de brindar mayor información al nombre de la variable, método o función.
La notación húngara fue utilizada para la definición de variables de acuerdo con los siguientes prefijos:

Tabla 13 Prefijos para la creación de variables.

Tipo de datos	Prefijos
Arreglos	arr
Objetos	obj
Enteros	int
Cadena	str
Float	flt
Boolean	bool

3.2 Diagrama de componente

Un diagrama de componentes permite visualizar con más facilidad la estructura general del sistema y el comportamiento de los servicios que estos componentes proporcionan y utilizan a través de las interfaces.

La solución consta de un solo componente denominado componente de interoperabilidad que interactúa con otros del marco de trabajo Sauxe. A continuación en la figura 9 se muestra el diagrama de componentes.

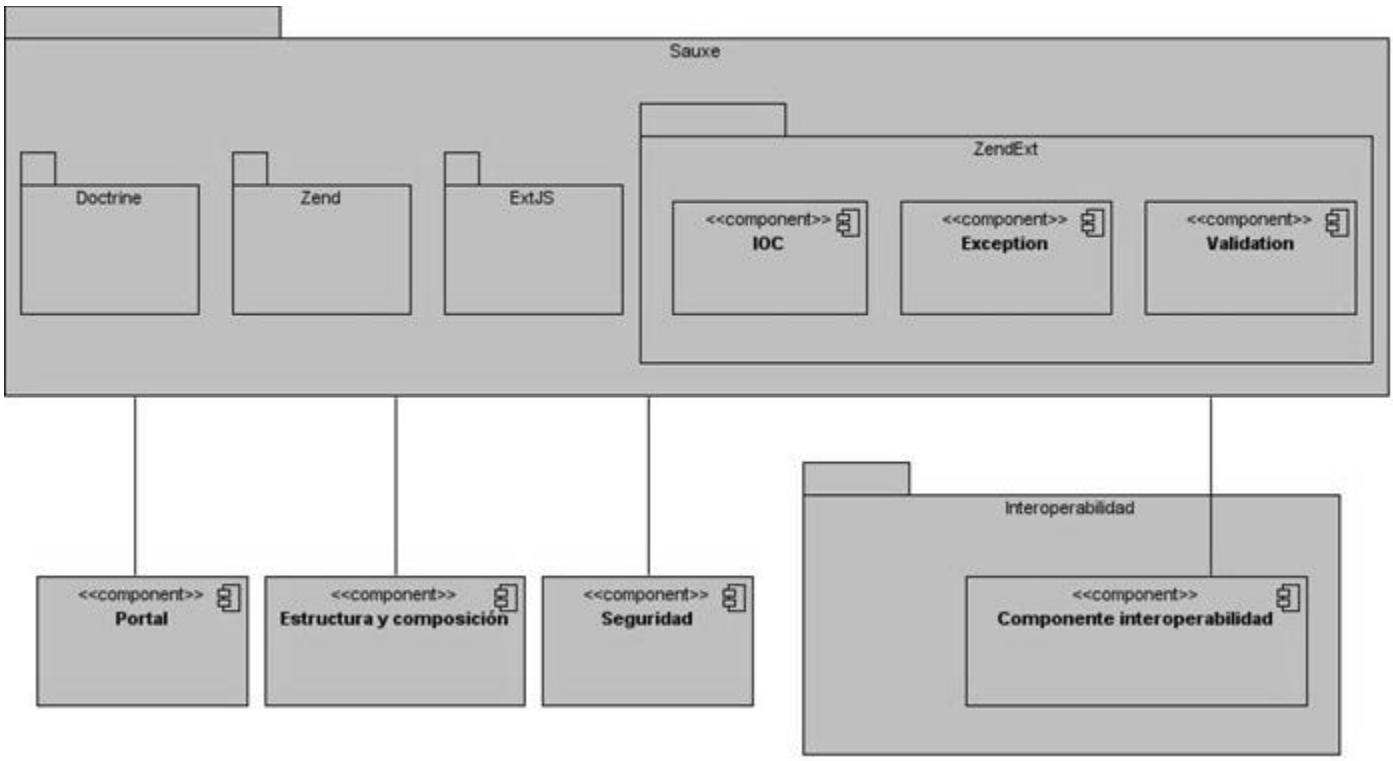


Figura 9 Diagrama de componentes

3.3 Diagrama de despliegue

Un diagrama de despliegue es un tipo de diagrama UML con el cual queda representado un modelado del hardware utilizado en la elaboración del software. En la figura 10 se describen los dispositivos necesarios para la utilización del producto, es decir, desde dónde se encuentra instalado, dónde se puede utilizar, hasta dónde se conecta para tomar los datos. Es una vista panorámica que describe los requisitos de hardware y software mínimos, necesarios para que esta herramienta funcione.

Figura 10 Diagrama de Despliegue.

3.4 Métricas de software

En el libro "Ingeniería del software, un enfoque práctico", Pressman plantea:

"El proceso del software y las métricas del producto son una medida cuantitativa que permite a la gente del software tener una visión profunda de la eficacia del proceso del software y de los proyectos que dirigen utilizando el proceso como un marco de trabajo. Se reúnen los datos básicos de calidad y productividad. Estos datos son entonces analizados, comparados con promedios anteriores, y evaluados para determinar las mejoras en la calidad y productividad. Las métricas son también utilizadas para señalar áreas con problemas de manera que se puedan desarrollar los remedios y mejorar el proceso del software". (24)

Las métricas empleadas están diseñadas para evaluar los siguientes atributos de calidad:

- **Responsabilidad:** se le asigna a una clase en un marco de modelado de un dominio o concepto, de la problemática propuesta.
- **Complejidad de implementación:** grado de dificultad en la implementación de un diseño de clases determinado.
- **Reutilización:** nivel de reutilización que tiene una clase o estructura de clase, dentro de un diseño de software determinado.
- **Acoplamiento:** valor de dependencia de una clase o estructura de clase con otras. Este atributo está muy ligado al de Reutilización.
- **Complejidad del mantenimiento:** categoría de esfuerzo para realizar un arreglo, mejora o rectificación de algún error de un diseño de software. Puede influir indirecta, pero fuertemente en los costes y la planificación del proyecto.

- **Cantidad de pruebas:** número de esfuerzos para realizar las pruebas de calidad (Unidad) del producto (componente, clase, conjunto de clases, etc.) diseñado.

Debido a que este software se realizó bajo la programación orientada a objetos (POO) y las clases constituyen la unidad básica y fundamental de un sistema orientado a objetos, es indiscutible que la validación del mismo se centró en la aplicación de métricas dirigidas a sus clases de forma individual, sus jerarquías y colaboraciones. A continuación se encuentran desarrolladas dichas métricas:

Tamaño operacional de clase (TOC): está dado por el número de métodos asignados a una clase y evalúa los siguientes atributos de calidad:

Tabla 14 Tamaño operacional de clase (TOC).

Atributo de calidad	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 del grado de reutilización de la clase.

Para los cuales están definidos los siguientes criterios y categorías de evaluación:

Tabla 15 Rango de valores para los criterios de evaluación de la métrica Tamaño Operacional de Clase (TOC).

Atributo	Categoría	Criterio
Responsabilidad	Baja	<=Promedio
	Media	Entre Promedio y 2*Promedio
	Alta	>2*Promedio
Complejidad implementación	Baja	<=Promedio
	Media	Entre Promedio y 2*Promedio
	Alta	>2*Promedio
Reutilización	Baja	>2*Promedio
	Media	Entre Promedio y 2*Promedio
	Alta	<=Promedio

3.4.1 Resultados obtenidos al aplicar la métrica de Tamaño Operacional de Clase (TOC)

Al aplicar la métrica TOC se determinó que la aplicación consta de 3 clases y 46 procedimientos, reflejados en la siguiente tabla junto a los atributos de calidad de Responsabilidad, Complejidad de

Implementación, Reutilización. Se obtuvo, un promedio de procedimientos por clase de 15, y los siguientes datos sirvieron para categorizar los atributos por cada clase:

Promedio de Procedimientos por clase = 15.
Responsabilidad: Baja <=15, 15< Media< 30, Alta> 30.
Complejidad: Baja <=15, 15< Media < 30, Alta> 30.
Reutilización: Baja> 30 ,15< Media < 30, Alta<=15.

Tabla 16 Instrumento de evaluación de la métrica TOC.

Clase	Cantidad de Procedimientos	Responsabilidad	Complejidad	Reutilización
ExportarimportarController	18	Media	Media	Media
GestorModel	20	Media	Media	Media
ProcesoModel	8	Baja	Baja	Alta

De acuerdo a los resultados obtenidos anteriormente se construyeron las siguientes gráficas para un mejor entendimiento de los mismos:

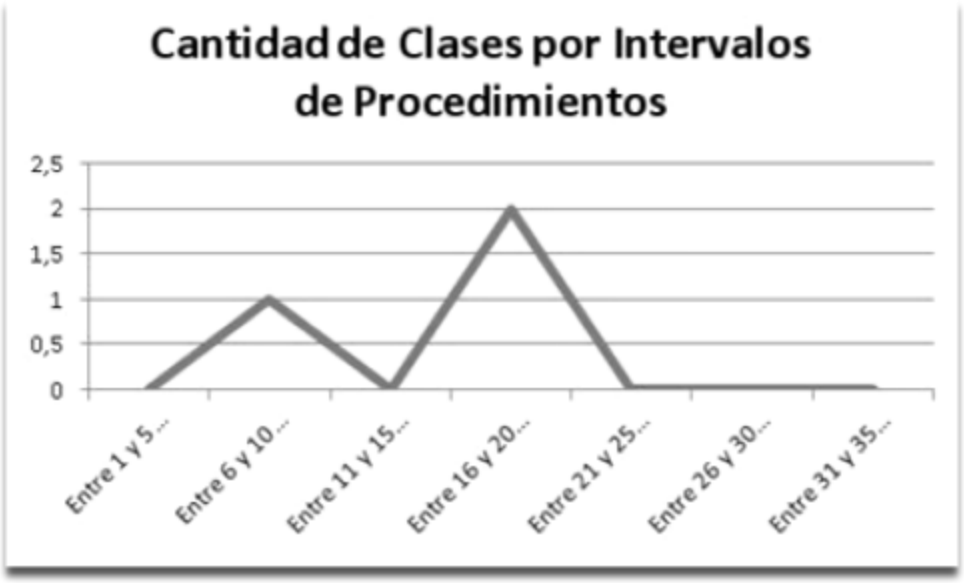


Figura 11 Cantidad de clases por intervalos de procedimientos.

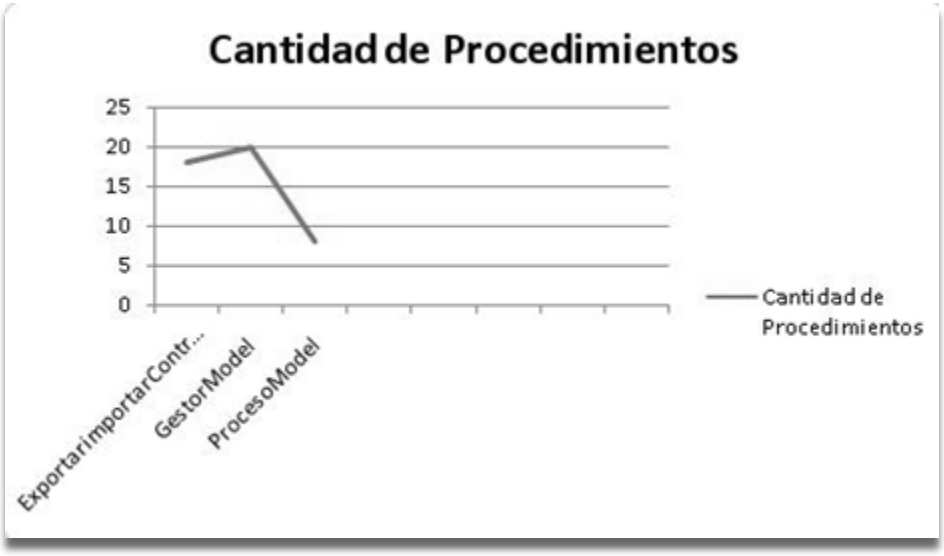


Figura 12 Cantidad de procedimientos por clases.

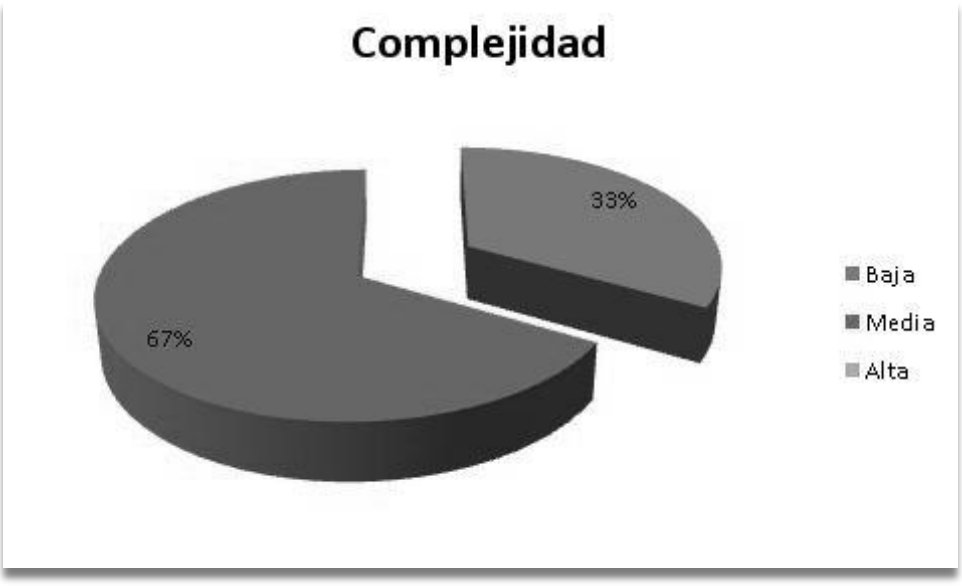


Figura 13 Resultados obtenidos de la evaluación de la métrica TOC para el atributo Complejidad.

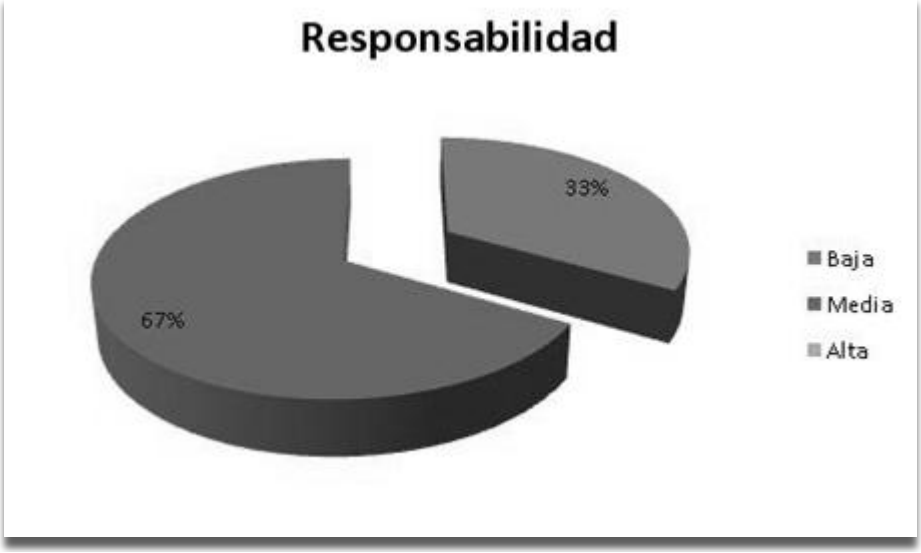


Figura 14 Resultados obtenidos de la evaluación de la métrica TOC para el atributo Responsabilidad.

Figura 15 Resultados obtenidos de la evaluación de la métrica TOC para el atributo Reutilización.

Relaciones entre clases (RC): Está dado por el número de relaciones de uso de una clase con otra y evalúa los siguientes atributos de calidad:

Tabla 17 Atributos de calidad evaluados por la métrica RC.

Atributo de calidad	Modo en que lo afecta
Acoplamiento	Un aumento del RC implica un aumento del Acoplamiento de la clase.
Complejidad de mantenimiento	Un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
Reutilización	Un aumento del RC implica una disminución en el grado de reutilización de la clase.
Cantidad de pruebas	Un aumento del RC implica un aumento de la Cantidad de pruebas de unidad necesarias para probar una clase.

Para los cuales están definidos los siguientes criterios y categorías de evaluación:

Tabla 18 Criterios de evaluación para la métrica RC.

Atributo	Categoría	Criterio
Acoplamiento	Ninguno	0
	Bajo	1
	Medio	2
	Alto	>2
Complejidad de mantenimiento	Baja	<=Promedio
	Media	Entre Promedio y 2*Promedio
	Alta	>2*Promedio
Reutilización	Baja	>2*Promedio
	Media	Entre Promedio y 2*Promedio
	Alta	<=Promedio
Cantidad de pruebas	Baja	<=Promedio
	Media	Entre Promedio y 2*Promedio
	Alta	>2*Promedio

3.4.2 Resultados obtenidos al aplicar la métrica de Relaciones entre Clases (RC)

De 3 clases existentes constan de 12 dependencias entre ellas, a continuación en la siguiente tabla estará reflejada junto a los atributos de calidad de Acoplamiento, Complejidad de Mantenimiento, Reutilización y Cantidad de Pruebas. Se obtuvo un promedio de relaciones de dependencia por clase de 4.

Promedio Cantidad de Relaciones de Uso: 4.

Acoplamiento: Ninguno=0, Bajo=1, Medio=2, Alto>2.

Complejidad Mantenimiento: Baja<=4, 4<Media<16, Alta>16.

Reutilización: Baja>16, 4<Media<16, Alta<=4.

Cantidad de Pruebas: Baja<=4, 4<Media>16, Alta>16.

Tabla 19 Instrumento de evaluación de la métrica RC.

Clase	Cantidad de Relaciones de Uso	Acoplamiento	Complejidad Mantenimiento.	Reutilización	Cantidad de Pruebas
ExportarimportarController	4	Alto	Baja	Alta	Baja
GestorModel	8	Alto	Media	Media	Media
ProcesoModel	0	Ninguno	Baja	Alta	Baja

De acuerdo a los resultados obtenidos anteriormente se construyeron las siguientes gráficas para un mejor entendimiento de los mismos:

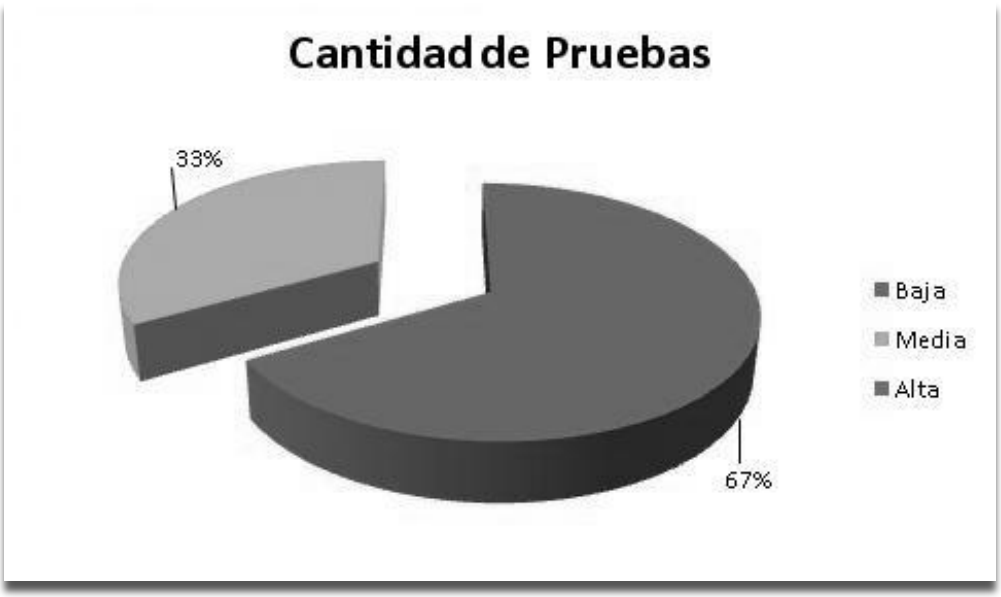


Figura 16 Resultados obtenidos de la evaluación de la métrica RC para el atributo Cantidad de Pruebas.

Figura 17 Resultados obtenidos de la evaluación de la métrica RC para el atributo Reutilización.

Figura 18 Resultados obtenidos de la evaluación de la métrica RC para el atributo Complejidad de mantenimiento.

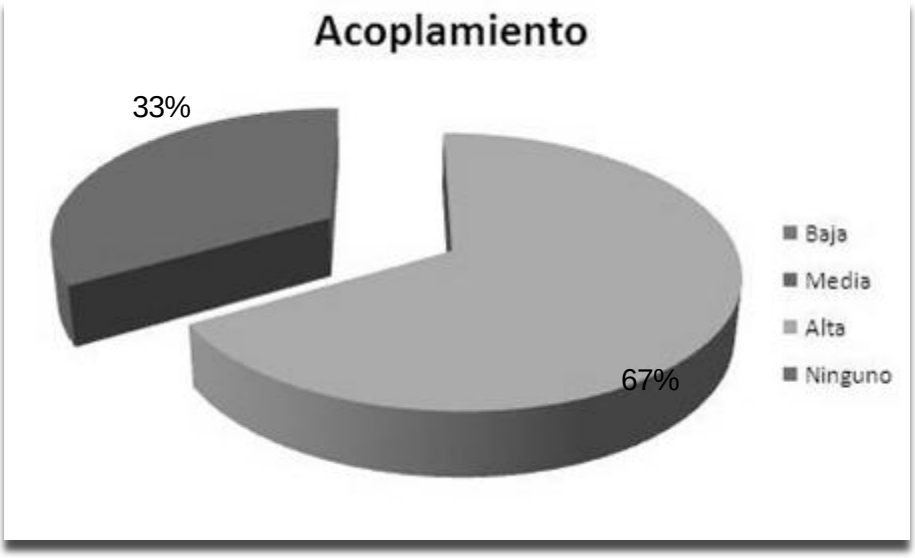


Figura 19 Resultados de la evaluación de la métrica RC para el atributo Acoplamiento.

3.4.3 Matriz de cubrimiento o matriz de inferencia de indicadores de calidad

La matriz de cubrimiento o matriz inferencia de indicadores de calidad es una representación estructurada de los atributos de calidad y métricas utilizadas en los epígrafes anteriores para evaluar la calidad del diseño de los componentes que integran la solución propuesta. La misma permite conocer si el resultado obtenido de la relación atributo/métricas para cada componente es positivo o negativo. Llevando estos resultados a una escala numérica donde, si los resultados son positivos tendrá un valor de 1, si son negativos de 0 y si no existe relación alguna se tomará como nula (-). Una vez completado los datos de dicha relación se realiza un cálculo donde se promedia la sumatoria de los valores obtenidos de un atributo por cada métrica evaluada, y la división de dicha sumatoria por la cantidad de métricas evaluadas (solo se promedian las que arrojan un resultado, las nulas no). Este valor es el que va a tener el atributo dentro de una tabla que medirá si los atributos fueron buenos, regulares o malos.

A continuación se muestran los resultados obtenidos:

Tabla 20 Resultados de la evaluación de la relación atributo/métrica.

Atributos\Métricas	TOC	RC	Promedio
Responsabilidad	1	(-)	1
Complejidad de	1	(-)	1

Implementación			
Reutilización	1	1	1
Acoplamiento	(-)	1	1
Complejidad de Mantenimiento	(-)	1	1
Cantidad de pruebas	(-)	1	1

Tabla 21 Rango de valores para la evaluación de la relación atributo/métrica

Categoría	Rango de valores
Malo	<=0.4
Regular	>0.4 y <0.7
Bueno	>=0.7

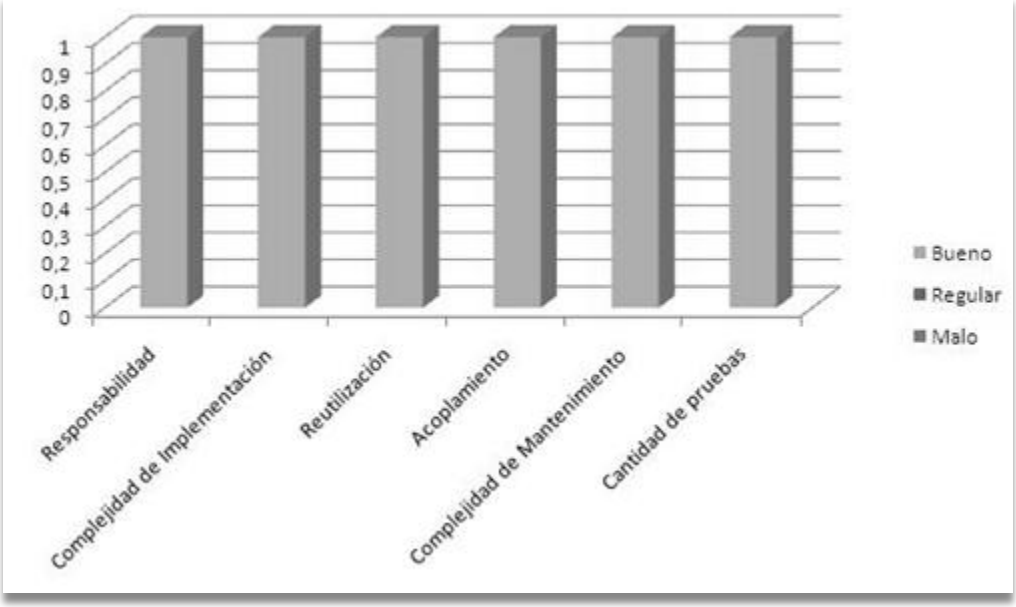


Figura 20 Resultados obtenidos de la evaluación de los atributos de calidad.

3.5 Pruebas de software

La prueba de software es un elemento esencial y crítico para la garantía de la calidad del software. El objetivo de la etapa de pruebas es garantizar la calidad del producto desarrollado. La prueba se enfoca sobre la lógica interna del software y las funciones externas. La prueba es un proceso que tiene como objetivo la ejecución de un programa con la intención de descubrir un error. Un buen caso de prueba es aquel que tiene alta probabilidad de mostrar un error no descubierto hasta entonces.

3.5.1 Pruebas de caja blanca

Permiten examinar la estructura interna del programa. Se diseñan casos de prueba para examinar la lógica del programa. Es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivar casos de prueba que garanticen que:

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

A continuación en la figura 21 se procede a enumerar las sentencias del código:

```
function cargarsubsistemasAction() {
    $idnodo = $this->_request->getPost('node'); (1)
    $certif = $this->global->Perfil->certificado;  (1)
    $entidad = $this->global->Estructura->idestructura;  (1)
    $sistemaArr = array();  (1)

    $subsistemas = $this->integrator->seguridad->getSistems($certif, $entidad, $idnodo); (1)
    $cant = 0;  (1)
    if(count($subsistemas)) {  (2)
        foreach ($subsistemas as $valor) {  (3)
            $sistemaArr[$cant]['idsistema'] = $valor['idsistema'];  (4)
            $sistemaArr[$cant]['text'] = $valor['text'];  (4)
            $sistemaArr[$cant]['idpadre'] = $valor['idpadre'];  (4)
            $sistemaArr[$cant]['id'] = $valor['id'] . $valor['idpadre'];  (4)
            $cant++;  (4)
        } (5)
    }
    echo json_encode($sistemaArr); (6)
    return;  (6)
}
```

Figura 21 Código que pertenece a la función cargarsubsistemasAction.

A continuación en la figura 22 se muestra el grafo de flujo asociado a la funcionalidad.

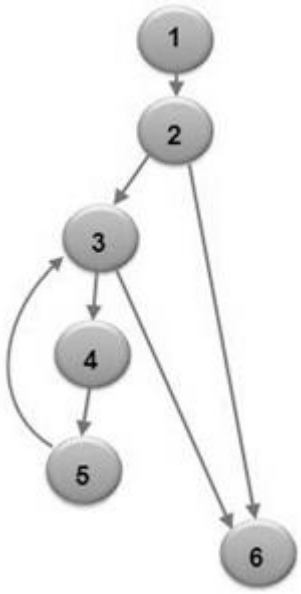


Figura 22 Grafo de flujo asociado a la funcionalidad cargarsubsistemasAction.

Una vez construido el grafo, la complejidad ciclomática es calculada aplicando las 3 fórmulas existentes para el análisis de la complejidad de algoritmos, garantizando que los resultados obtenidos en cada una de ellas sean los mismos para que el cálculo de la complejidad sea correcto. Estas fórmulas son:

- V (G) = (A (Aristas) – N (Nodos)) + 2.
- V (G) = P (Nodos Predicados) + 1.
- V (G) = R.

A continuación los resultados aplicando dichas fórmulas:

Aplicando la fórmula 1:

V (G) = (7 – 6) + 2

V (G) = 3.

Aplicando la fórmula 2:

P (Nodos Predicados): Son los nodos de los cuales parten dos o más aristas.

V (G) = 2 + 1

V (G) = 3

Aplicando la fórmula 3:

Siendo "R" la cantidad total de regiones.

V (G) = 3

De acuerdo con los resultados, la complejidad ciclomática del código es 5, significando que existen 5 posibles caminos por donde el flujo puede circular, este valor representa el límite mínimo del número total de casos de pruebas para el procedimiento seleccionado. A continuación se muestran los caminos básicos extraídos del grafo.

Caminos básicos:

Camino # 1: (1-2-3-4-5-3-6)

Camino # 2: (1-2-3-6)

Camino # 3: (1-2-6)

Para cada camino se realiza un caso de prueba, y es preciso 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.
- **Resultados Esperados:** se expone el resultado que se espera que devuelva el procedimiento.
- **Evaluación de los resultados:** se exhibe la evaluación que dio el resultado final del procedimiento.

- **Caso de prueba para el camino básico # 1.**

Descripción: Los datos que van adoptar las variables idnodo, certif, y entidad cumplirán con los siguientes requisitos: No pueden ser nulos.

Condición de ejecución:

Variables $idnodo, $certif, $entidad no pueden ser nulos.

($certif = 12, $idnodo=1, $entidad=uci)

Resultados esperados: Al ejecutar el procedimiento se cargarán los subsistemas dado el certificado, la entidad y el id del nodo en donde se va a cargar.

Evaluación de los resultados obtenidos: Los resultados obtenidos de la realización del caso de prueba de caja blanca para el caso de cargar los subsistemas, fueron satisfactorios al lograr que se cargaran los subsistemas.

- **Caso de prueba para el camino básico # 2.**

Descripción: Los datos que van adoptar las variables idnodo, certif, y entidad cumplirán con los siguientes requisitos: No pueden ser nulos.

Condición de ejecución:

Variables $idnodo, $certif, $entidad no pueden ser nulos.
($certif = 23, $idnodo=2, $entidad=copextel)

Resultados esperados: Al ejecutar el procedimiento no se cargarán los subsistemas dado el certificado, la entidad y el id del nodo en donde se va a cargar, debido a que no existen subsistemas para ese certificado y la entidad.

Evaluación de los resultados obtenidos: Los resultados obtenidos de la realización del caso de prueba de caja blanca para el caso de cargar los subsistemas, fueron satisfactorios al lograr los resultados esperados.

- **Caso de prueba para el camino básico # 3.**

Descripción: Los datos que van adoptar las variables idnodo, certif, y entidad cumplirán con los siguientes requisitos: No pueden ser nulos.

Condición de ejecución:

Variables $idnodo, $certif, $entidad no pueden ser nulos.

($certif = 56, $idnodo=3, $entidad=etecsa)

Resultados esperados: Al ejecutar el procedimiento no se cargarán los subsistemas dado el certificado, la entidad y el id del nodo en donde se va a cargar, debido a que no existen subsistemas para ese certificado y la entidad.

Evaluación de los resultados obtenidos: Los resultados obtenidos de la realización del caso de prueba de caja blanca para el caso de cargar los subsistemas, fueron satisfactorios al lograr los resultados esperados.

3.5.2 Pruebas de caja negra

Se denominan pruebas funcionales o Functional Testing, a las pruebas de software que tienen por objetivo probar que los sistemas desarrollados, cumplan con las funciones específicas para los cuales han sido creados, es común que este tipo de pruebas sean desarrolladas por analistas de pruebas con apoyo de algunos usuarios finales, esta etapa suele ser la última etapa de pruebas y al dar conformidad sobre esta el paso siguiente es el pase a producción. (25)

A este tipo de pruebas se les denomina también pruebas de comportamiento o pruebas de caja negra, ya que los testers[7] o analistas de pruebas, no enfocan su atención a como se generan las respuestas del sistema, básicamente el enfoque de este tipo de prueba se basa en el análisis de los datos de entrada y en los de salida, esto generalmente se define en los casos de prueba preparados antes del inicio de las pruebas. (25)

Algunas técnicas utilizadas en la prueba de caja negra son:

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

- ***Análisis de valores límite:*** La experiencia muestra que los casos de prueba que exploran las condiciones límite producen mejor resultado que aquellos que no lo hacen. Las condiciones límite

[7] Persona que realiza las pruebas.

son aquellas que se hallan en los márgenes de la clase de equivalencia, tanto de entrada como de salida. Por ello, se ha desarrollado el análisis de valores límite como técnica de prueba. Esta técnica lleva a elegir los casos de prueba que ejerciten los valores límite.

- ***Grafos de causa-efecto:*** Es una técnica que permite al encargado de la prueba validar complejos conjuntos de acciones y condiciones.

A continuación se muestran los casos de prueba que se definieron para cada uno de los requisitos del sistema. Su objetivo principal es demostrar la reacción que corresponderá por parte del sistema luego de realizar alguna acción en el mismo. Para un mejor entendimiento de las respuestas o posibles funcionalidades que brindará el sistema, según la necesidad del usuario.

Tabla 22 Escenario de prueba del requisito Gestionar apertura.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
Gestionar apertura.	Seleccionar los documentos de apertura que se desean interoperar a través del parámetro **depósito**.	EP 1.1: **Interfaz principal.**	-Dar clic en la interfaz principal, en el botón **Inicio.** -Seleccionar en el menú del botón **Inicio** la opción **Interoperabilidad.** -Seleccionar **Componente de interoperabilidad.** -El sistema muestra la interfaz **Servicios de interoperabilidad.**
		EP 1.2: **Servicios de interoperabilidad.**	-Seleccionar en el árbol de los subsistemas **Logística** y dentro de él se seleccionará **Inventario** -Dentro de **Inventario** seleccionar **Apertura**. -Seleccionar dentro de **Apertura** el nodo **Apertura.** -Presionar la opción **Exportar**. -El sistema muestra la interfaz **Gestionar apertura.**
		EP 1.3: **Gestionar**	-Seleccionar una opción en el combo **Depósito.** Esta selección nos mostrará

		apertura.	todos los documentos inventario de aperturas disponibles a interoperar que cumplen con el dato anteriormente seleccionado. -Chequear los documentos inventario de aperturas que se desean interoperar. -Presionar el botón **Aceptar.**

Tabla 23 Escenario de prueba del requisito Configurar atributos de apertura.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
Configurar atributos de apertura.	Se realizará la configuración de los atributos de los documentos inventario de apertura a interoperar que fueron seleccionados según el parámetro de **depósitos.**	EP 1.1: **Estructura del XML a exportar.**	-La acción a realizar es configurar los atributos de los documentos inventario de apertura que fueron seleccionados en el **Gestionar apertura**. -En el árbol de configuración de atributos deberá seleccionar al menos un atributo el cual conformará la estructura de los documentos inventario de apertura a exportar en .ZIP. -Presionar el botón **Aceptar**.

Tabla 24 Escenario de prueba del requisito Descargar apertura.

Nombre del requisito	Descripción general	Escenarios de pruebas	Flujo del escenario
Descargar apertura.	Descargar Archivo .ZIP con los documentos inventario de apertura seleccionados y configurados.	EP 1.1: **Descargar Archivo.**	-Descargar archivo XML con los atributos de los documentos inventario de apertura seleccionado y configurado en escenarios anteriores.

El resto de los casos de prueba correspondientes a los demás requisitos del sistema se encuentran descritos en el anexo 2 de este documento. El componente fue probado por el departamento de calidad del centro CEIGE donde se comprobó el buen funcionamiento del mismo y queda avalado por el acta de liberación expuesta en el anexo 3.

3.6 Conclusiones parciales

En el capítulo fueron expuestos los artefactos generados en la etapa de implementación de la solución, tales como los estándares de codificación, el diagrama de componente y el diagrama de despliegue. Además se aplicaron métricas dirigidas a evaluar la calidad del diseño del software, basadas en diferentes atributos de calidad que avalaron sus resultados. Se realizaron las pruebas del camino básico como parte de las pruebas de caja blanca para garantizar el buen funcionamiento del código implementado, así como su eficiencia y solidez. Y por último para demostrar el cumplimiento de los requisitos definidos, se aplicaron pruebas de caja negra guiándose por los casos de prueba que se definieron para cada uno de los requisitos del sistema.

CONCLUSIONES

Al desarrollar el componente utilizando las herramientas y tecnologías propuestas, se logró que esta formara parte del sistema CEDRUX. Fue posible gracias al estudio del estado del arte realizado sobre las herramientas, tecnologías y técnicas aplicadas a nivel mundial, lo cual sentó las condiciones para su posterior análisis, diseño e implementación.

Los resultados obtenidos fueron analizados mediante la validación de su diseño y una etapa de pruebas que reveló el cumplimiento de los requerimientos propuestos.

El componente de interoperabilidad permitirá que productos claves para el desarrollo del país, como CEDRUX, puedan interoperar con otras aplicaciones en las entidades en las cuales son desplegados.

RECOMENDACIONES

- Una vez concluido el proceso de desarrollo se recomienda la realización de una versión posterior donde se incluyan el resto de los procesos del ERP.

- Se recomienda el uso del componente de Interoperabilidad para el intercambio de datos entre el Sistema Cedrux y el resto de los sistemas de gestión usados en Cuba.

- Se recomienda que las entidades cubanas que aún no se encuentran inmersas en este proceso de intercambio, utilicen como material de estudio la documentación generada por el componente de interoperabilidad para su auto preparación.

REFERENCIAS

1. *BOLETÍN OFICIAL DEL ESTADO.* **MINISTERIO DE LA PRESIDENCIA.** 2010, Vol. 25. 0212-033X.

2. **IEEE.** A Compilation of IEEE Standard Computer Glossaries. [En línea] 1990.

3. **Perojo, Lic. Keilyn Rodríguez.** Un nuevo enfoque hacia la Organización de Información en los Sistemas de Gestión de Contenidos. [En línea] [Citado el: 20 de febrero de 2011.] http://www.bibliociencias.cu/gsdl/collect/eventos/index/assoc/HASH01a3/6ff7e68d.dir/doc.pdf.

4. *REVISTA INGENIERÍA E INVESTIGACIÓN. Revisión de la literatura en interoperabilidad entre sistemas heterogéneos de software.* **Zapata, Carlos M. y Calderón, Guillermo González.** 2009, Vol. 29.

5. **C., Liliana M. Arboleda.** *Servicios WEB: Distribución e integración.* 2004.

6. **Acero, Fernando.** Interoperabilidad (y IV): Estrategias para las organizaciones. [En línea] 2009. [Citado el: 23 de febrero de 2011.] http://www.kriptopolis.org/interoperabilidad-4.

7. **Enterprise Integration INC.** *SAP Overview, Getting Started with Standards at SAP, What is an SAP business blueprint (2007 Enterprise Integration,INC.).* 2009.

8. **Nielsen, Michael.** *Microsoft Dynamics NAV.* 2009. 1040610.

9. **Allen Jacot, Joseph E. Miller, Michael Jacot, John A. Stern.** *JD Edwards EnterpriseOne, The Complete Reference.* 2009. 978-0-07-159874-3.

10. *IFS architecture and technology. Engineered for agility.* 2006.

11. **Pérez, Mileidys Sarduy.** *Propuesta de modelo de desarrollo de software tecnológico del Centro de Soluciones de Gestión.* Habana : s.n., 2009.

12. Manuales, Tutoriales y Herramientas. [En línea] [Citado el: 17 de febrero de 2011.] http://max-alva.webs.com/javascript.htm.

13. *Grupo de documentación de PHP. Manual de PHP.* 2001.

14. Doctrine-PHP Object Persistence Libraries and More. [En línea] 2010. [Citado el: 17 de febrero de 2011.] http://www.doctrine-project.org.

15. **Frederick, Shea, Ramsay, Colin y Blades, Steve 'Cutter'.** *Learning Ext JS. Birmingham - Mumbai : PACKT.* 2008. 978-1-847195-14-2.

16. **Esser, Stefan.** Secure Programming with the Zend-Framework. *Dutch PHP Conference.* Amsterdam : s.n : s.n., 2009.

17. Área temática de Linux. Portal de Linux - Ciberaula. [En línea] [Citado el: 17 de febrero de 2011.] http://linux.ciberaula.com/articulo/linux_apache_intro.

18. netbeans. [En línea] [Citado el: 17 de febrero de 2011.] http://netbeans.org/community/releases/69.

19. PostqreSQL. [En línea] 2011. [Citado el: 18 de febrero de 2011.] http://www.postgresql.org/docs/8.3/static/release-8-3.html.

20. Visual Paradigm For Uml. [En línea] [Citado el: 2011 de Marzo de 28.] http://www.slideshare.net/vanquishdarkenigma/visual-paradigm-for-uml.

21. **Sommerville, Ian y Sawyer, Pete.** *Requirements Engineering: A good practice guide.* Lancaster : s.n.

22. **Deacon, John.** Model-View-Controller (MVC) Architecture. [En línea] [Citado el: 5 de marzo de 2011.] http://www.jdl.co.uk/briefings/mvc.pdf.

23. **Burbeck, Steve.** Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller (MVC). [En línea] [Citado el: 6 de marzo de 2011.] http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html.

24. **Pressman, Roger S.** *Ingenieria del Software - Un Enfoque Practico.* s.l. : s.l. : McGraw-Hill Companies, 2002. 8448132149.

25. **B, Ing. Alexander Oré.** Calidad y software. [En línea] [Citado el: 2011 de Marzo de 25.] http://www.calidadysoftware.com/testing/pruebas_funcionales.php.