

**Universidad de las Ciencias Informáticas**

**Facultad 1, Facultad 15**



**Título: Guía metodológica de interoperabilidad para sistemas integrales de gestión en Cuba**

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

**Autor(es):** José Rogel Meneces Rodríguez  
Enrique Ramón Osoria Nápoles

**Tutor(es):** Ing. Pedro Manuel Nogales Cobas  
**Co-tutor:** Ing. Magdanis Galván Rey

Ciudad de La Habana, Junio de 2010

## Declaración de Autoría

---

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_  
**Enrique Ramón Osoria Nápoles**

\_\_\_\_\_  
**José Rogel Meneces Rodríguez**

\_\_\_\_\_  
**Ing. Pedro Manuel Nogales Cobas**

\_\_\_\_\_  
**Ing. Magdanis Galván Rey**

El presente trabajo de diploma es una guía metodológica, para el desarrollo de la interoperabilidad en sistemas integrales de gestión en Cuba, con el propósito de lograr el intercambio de datos e información entre ellos.

El desarrollo de la guía metodológica de interoperabilidad, estará previamente respaldada por un estudio teórico y bien detallado sobre los principales conceptos relacionados con la interoperabilidad entre sistemas integrales de gestión. Se realiza una amplia investigación sobre los aspectos que intervienen en la construcción de la solución, como son: las herramientas, taxonomías y estándares a utilizar de acuerdo con los requisitos del sistema, entre otros aspectos.

La guía metodológica, contará con una serie de pasos a seguir a la hora de interoperar sistemas integrales de gestión. Estos pasos recogerán aspectos relacionados con la arquitectura de la solución a desarrollar, la seguridad, las reglas de interoperabilidad, la implementación, y demás cuestiones. Estos pasos que componen la lógica en la construcción de la solución para lograr la interoperabilidad, estarán evaluados en un caso de estudio aplicado a un componente del Sistema Integral de Gestión CedruX.

El resultado de este trabajo será una guía metodológica, que estandarice el desarrollo de la interoperabilidad entre sistemas integrales de gestión en Cuba, para de esta forma cubrir las dificultades existentes de una solución óptima.

**Palabras claves:** Estándares, Interoperabilidad, Taxonomías.

<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA .....</b>	<b>5</b>
1.1 Introducción.....	5
1.2 Interoperabilidad de software .....	5
1.3 Tendencia a nivel internacional .....	6
1.4 Formas de interoperar .....	8
1.5 Interoperabilidad en los sistemas integrales de gestión.....	10
1.5.1 Interoperabilidad del software en Cuba .....	10
1.6 Buenas prácticas para el desarrollo de la interoperabilidad.....	12
1.7 Arquitectura de software.....	13
1.8 Arquitectura orientada a servicios web .....	16
1.9 Estándares de interoperabilidad.....	17
1.10 Taxonomías .....	19
1.11 Seguridad. Formas de proveer seguridad .....	20
1.12 Entornos de desarrollo .....	22
1.12.1 Lenguajes.....	22
1.12.2 Herramientas.....	24
1.13 Conclusiones parciales.....	27
<b>CAPÍTULO 2: GUÍA METODOLÓGICA.....</b>	<b>28</b>
2.1 Introducción.....	28
2.2 Guía metodológica .....	28
2.3 Conclusiones parciales.....	47
<b>CAPÍTULO 3: CASO DE ESTUDIO APLICADO AL SISTEMA INTEGRAL DE GESTIÓN CEDRUX.....</b>	<b>48</b>
3.1 Introducción.....	48
3.2 Aplicación de la guía metodológica al proceso Comprobante de Operaciones del subsistema Contabilidad.....	48
3.3 Conclusiones parciales.....	64
<b>CONCLUSIONES GENERALES .....</b>	<b>66</b>
<b>RECOMENDACIONES.....</b>	<b>68</b>
<b>BIBLIOGRAFÍA.....</b>	<b>69</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>74</b>

<b>FIGURA 1. GESTIONAR INTEROPERABILIDAD .....</b>	<b>56</b>
<b>FIGURA 2. ADICIONAR PROCESOS .....</b>	<b>57</b>
<b>FIGURA 3. MODIFICAR PROCESOS .....</b>	<b>58</b>
<b>FIGURA 4. CONFIRMACIÓN .....</b>	<b>58</b>
<b>FIGURA 5. GESTIONAR INTEROPERABILIDAD (AUTENTICACIÓN BASE DE DATOS).....</b>	<b>59</b>
<b>FIGURA 6. GESTIONAR INTEROPERABILIDAD (CONFIRMACIÓN DE ADICIÓN) .....</b>	<b>60</b>
<b>FIGURA 7. GESTIONAR COMPROBANTES .....</b>	<b>62</b>
<b>FIGURA 8. EXPORTAR E IMPORTAR.....</b>	<b>63</b>
<b>FIGURA 9. ELEMENTOS DEL PROCESO COMPROBANTE DE OPERACIONES .....</b>	<b>64</b>

## INTRODUCCIÓN

En el mundo globalizado actual, la transmisión oportuna, segura e íntegra de la información de negocios es absolutamente necesaria. El intercambio de información en el marco señalado, ha tenido problemas históricos con los distintos formatos digitales usados, producto de los requisitos específicos que exigen algunos reguladores, la diferencia entre los sistemas computacionales que poseen las empresas y el habitual reprocesamiento de la información. El mercado de software empresarial disponía, al igual que en la actualidad, de productos desarrollados por diferentes compañías las cuales no posibilitaban la emisión de ficheros en formatos compatibles; haciendo que el proceso de captación y consolidación de cualquier informe requiriera de la labor manual de los especialistas. Para darle solución al problema de la emisión de ficheros en formatos compatibles surge el Lenguaje Extensible de Mercado (XML).

En la primera década del siglo XXI, la economía cubana tiende a una recuperación paulatina y las empresas no se encuentran ajenas a este avance, pero este sector debe llegar al perfeccionamiento empresarial en su conjunto y necesita de un sistema para la gestión de los recursos empresariales. Esto ha incentivado la búsqueda de soluciones tanto extranjeras como nacionales. Ante esta situación, la dirección del país propuso la realización de un software de gestión para automatizar los procesos que se realizan en las entidades, surgiendo en el año 2008 el Sistema Integral de Gestión Cedrux, dentro de la Universidad de las Ciencias Informáticas (UCI).

Creándose un sistema que debe tener como una de sus premisas fundamentales, el ser capaz de interoperar con otros sistemas homólogos de Cuba, cuyos beneficios estén orientados a la extensibilidad del producto en diferentes entornos. Esto quiere decir que todos los sistemas pueden interoperar, independientemente de las tecnologías que utilizan. La homogeneidad y el orden es otro de los beneficios que trae consigo la interoperabilidad como funcionalidad de un sistema, ya que se logra uniformidad y organización en diversos aspectos que rigen la interoperabilidad mediante acuerdos entre los sistemas involucrados. La capacidad de distribuir los datos es otra de las ventajas que se obtiene, se logra de forma rápida y eficiente, proporcionando seguridad a la información y resultados palpables para el negocio.

Producto a la necesidad que tiene Cedrux de absorber toda la información existente en los sistemas integrales de gestión en Cuba, para de esta forma integrarlos y automatizar el intercambio de información; se requiere estandarizar la construcción de una plataforma de interoperabilidad.

Teniendo en cuenta lo expuesto previamente se plantea como **problema científico** de la investigación: La necesidad de estandarizar el proceso de construcción de la plataforma de interoperabilidad para sistemas integrales de gestión en Cuba.

Por lo que el **objeto de estudio** comprende la interoperabilidad en sistemas integrales de gestión.

Para darle solución al problema planteado se define como **objetivo general**, desarrollar una guía metodológica que defina como lograr la interoperabilidad entre sistemas integrales de gestión en Cuba.

En correspondencia con ello, se plantean los siguientes **objetivos específicos**:

- Realizar el marco teórico sobre la interoperabilidad en sistemas integrales de gestión.
- Definir una arquitectura tecnológica que permita lograr la interoperabilidad entre sistemas integrales de gestión.
- Analizar los estándares y taxonomías utilizadas en el proceso de interoperabilidad.
- Definir un esquema de seguridad que permita proteger el intercambio de información durante proceso de interoperabilidad.
- Definir un entorno de desarrollo que permita lograr con eficiencia el proceso de interoperabilidad.
- Formalizar la guía metodológica para la interoperabilidad en sistemas integrales de gestión en Cuba.
- Validar la guía metodológica mediante un caso de estudio.

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

Como **idea a defender** se plantea que: El desarrollo de una guía metodológica de interoperabilidad, estandarizará el intercambio de información del sistema CedruX con otros sistemas integrales de gestión en Cuba.

Para dar cumplimiento a los objetivos planteados se trazan las siguientes **tareas investigativas**:

- Estudio bibliográfico sobre la interoperabilidad de software y su desarrollo a nivel nacional e internacional.
- Estudio y definición de una arquitectura tecnológica que permita realizar la interoperabilidad en sistemas integrales de gestión.

- Revisión y propuesta de estándares y taxonomías más usados en el proceso de interoperabilidad de software, así como las herramientas dedicadas a este fin.
- Revisión de los estándares, protocolos y herramientas necesarias para definir un sistema de seguridad que permita proteger los datos en el intercambio de información entre sistemas integrales de gestión.
- Identificación de un ambiente de desarrollo que permita llevar a cabo con éxito el proceso de interoperabilidad.
- Formalización de la guía metodológica para normalizar el proceso de interoperabilidad en Cuba.
- Validación de la guía metodológica mediante un caso de estudio aplicado al Sistema Integral de Gestión Cedrux.

Para cumplir con las tareas antes mencionadas se desarrollará una guía metodológica de interoperabilidad clara, segura y orientada a las necesidades planteadas; acorde con los requerimientos necesarios.

El documento está estructurado de la siguiente manera: resumen, introducción, y tres capítulos que constituyen el cuerpo fundamental del mismo. Los capítulos son:

Capítulo 1 “**Fundamentación Teórica**”: Se exponen los fundamentos generales que sirven de soporte teórico en la solución del problema. Se definen elementos de interoperabilidad, algunas cuestiones sobre arquitectura, los estándares y taxonomías, las formas de proveer seguridad a la solución, así como los entornos de desarrollo.

Capítulo 2 “**Guía Metodológica**”: Se propone una guía metodológica de interoperabilidad, la cual cuenta con una serie de pasos a seguir que sirven como referencia para lograr la interoperabilidad entre sistemas integrales de gestión en Cuba.

Capítulo 3 “**Caso de estudio aplicado al Sistema Integral de Gestión Cedrux**”: Se presenta un caso de estudio donde se aplica la guía metodológica al proceso Comprobante de Operaciones del subsistema Contabilidad para el desarrollo de una plataforma de interoperabilidad.

El documento cuenta además con las **Conclusiones** del trabajo, algunas **Recomendaciones** a tener en cuenta para la puesta en práctica de la guía metodológica de interoperabilidad en el Sistema Integral de Gestión CedruX, las **Bibliografías** y un **Glosario de Términos**.

## **CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA**

### **1.1 Introducción**

En el presente capítulo se fundamentan cada uno de los aspectos que inciden en el desarrollo de una solución para sistemas integrales de gestión en Cuba. Se realiza un estudio sobre diversos temas relacionados con la interoperabilidad entre sistemas de gestión, a nivel nacional e internacional. Se valoran los elementos más significativos para la construcción de la plataforma de interoperabilidad, como son: estándares, taxonomías, seguridad, herramientas; entre otras cuestiones.

Se estudian las características generales de la arquitectura de software y se enfatiza en la arquitectura orientada a servicios web. Se tienen en cuenta además todos aquellos aspectos que complementan el contenido teórico, que sirven como base y estructura en la elaboración de la guía metodológica que será utilizada en la construcción de una solución de interoperabilidad para sistemas integrales de gestión en Cuba.

### **1.2 Interoperabilidad de software**

#### **¿Qué es interoperabilidad?**

Si bien la interoperabilidad puede tener significados diferentes dependiendo del contexto, en el área de las tecnologías de la información y las comunicaciones, el término se suele entender como “la capacidad de diferentes productos y servicios de las Tecnologías de la Información y las Comunicaciones (TIC) para intercambiar y usar datos e información (es decir “hablar”) con el objetivo de funcionar juntos en un entorno conectado en red”. En su acepción más simple, la interoperabilidad trata de asegurar que los sistemas trabajen juntos. (Borges, 2007)

Anteriormente en el año 2006, el asesor en tecnologías de la información para la Administración Federal de Ingresos Públicos en Argentina, Eduardo Poggi; definió la interoperabilidad como la capacidad de los sistemas (con la acepción más amplia posible) de tecnología de la información y las comunicaciones y de los procesos de negocio que soportan, para intercambiar datos entre sí y viabilizar la distribución de la información y el conocimiento. (Poggi, 2006)

En el año 2007 el director de tecnología de Microsoft Ibérica Héctor Sánchez plantea, que existen varios conceptos de interoperabilidad, asegurando que quizás la de la asociación EICTA, nombrada por su acrónimo inglés Asociación Europea de Industrias de Tecnologías de la Información y las Comunicaciones es la más clarificadora, definiéndola como la capacidad de dos o más sistemas, aplicaciones, redes o componentes para intercambiar información y utilizar la información intercambiada.

Luego de analizar esta serie de definiciones, se ha llegado a la conclusión de que interoperabilidad es la característica que poseen los ordenadores que les permite su interconexión y funcionamiento en conjunto de manera compatible. La capacidad de los sistemas de comunicarse con otros sistemas externos para transmitir y recibir información, notificar o gestionar sucesos, así como para exportar e importar datos.

### **1.3 Tendencia a nivel internacional**

#### **Evolución de la interoperabilidad**

Desarrollar la interoperabilidad para lograr la comunicación entre sistemas homólogos con tecnologías heterogéneas ha sido un tema complicado a lo largo de la historia. La informática desde su surgimiento fue ampliamente utilizada por los departamentos de contabilidad de las grandes empresas, que podían pagar los altos costos de la tecnología de aquella época y luego se fue extendiendo a otras áreas de procesos.

Con el transcurso de los años fueron surgiendo en diversas partes del mundo y desarrollados por diversas compañías: estándares, protocolos y otros aspectos que dieron los primeros pasos de avance en el desarrollo del intercambio digital. Sin lugar a dudas comenzaban tiempos que imponían la construcción de soluciones informáticas para automatizar la comunicación entre empresas y compañías.

En 1977, la Organización Internacional de Estándares (ISO) integrada por industrias representativas del medio, creó un subcomité para desarrollar estándares de comunicación de datos que promovieran la accesibilidad universal y una interoperabilidad entre productos de diferentes fabricantes. El resultado de estos esfuerzos es el modelo de referencia de Interconexión de Sistemas Abiertos (OSI). El modelo OSI es un lineamiento funcional para tareas de comunicaciones y, por consiguiente, no especifica un estándar de comunicación para dichas tareas. Sin embargo, muchos estándares y protocolos cumplen con los lineamientos del modelo OSI. (Virgen, 2000)

La llegada de TCP/IP (un protocolo abierto de interconexión entre sistemas, siglas hoy conocida por todo el planeta), permitió un gran avance a la interoperabilidad. Los protocolos TCP e IP, surgidos en la década de los años setenta, con posteriores versiones en el paso de los años, son los verdaderos impulsores del nacimiento de Internet y con ello lo que posteriormente sería el comienzo de la comunicación digital. Un simple protocolo abierto, público y conocido; construido por la comunidad y que todos los proveedores tecnológicos se vieron obligados a implementar, permitió el sueño dorado: correo electrónico, Protocolo de Transferencia de Archivos (FTP), Telnet, gopher y aplicaciones disponibles en las más diversas plataformas que permitían la comunicación entre sistemas muy disímiles.

HTML/http llegó a principios de los noventa con aplicaciones desconocidas llamadas navegadores, para transferir texto, imágenes y componer páginas armónicas y gráficamente atractivas, en una pantalla remota. Esto sin lugar a dudas fue un gran paso de avance a las soluciones de interoperabilidad entre aplicaciones, que vendrían posteriormente con mayores retos y dificultades.

El momento más significativo en cuestiones de interoperabilidad, se produce en el año 1998 cuando Charles Hoffman (contador norteamericano) propuso la creación de un estándar para la comunicación de los estados financieros en formato digital basado en XML. Internet estaba en pleno desarrollo, no había explotado aún la crisis de las empresas.com y se trabajaba intensamente en la estandarización de las tecnologías de intercambio de información digital.

En los entornos informáticos heterogéneos actuales y con el surgimiento de los servicios web y las arquitecturas orientadas a los servicios, la interoperabilidad se ha tornado una necesidad crítica comercial, muy necesaria para las empresas e instituciones; por lo cual las soluciones de interoperabilidad crecen en la actualidad en cuanto a demanda y productos como solución.

## **Estado actual**

En el mundo se han desarrollado software de gestión con la funcionalidad de proporcionar interoperabilidad a diversos sistemas. Las experiencias han sido extraordinarias y con resultados excelentes, incluso han surgido nuevos estándares especializados para intercambiar procesos o módulos específicos, siempre apoyados en el estándar XML.

Un ejemplo es el Sistema Electrónico para la Contratación Pública (SECOP), de la República de Colombia; el cual es capaz de interoperar con otros sistemas como el Sistema Integrado de Información Financiera (SIIF) del ministerio de Hacienda y Crédito Público, y el Sistema de Información para la Contratación Estatal (SICE) de la Contraloría General de la República; posibilitando realizar el intercambio de un gran número de consultas sobre proveedores, contratos y precios, así como listar recursos de estas entidades. (González, 2008)

En la actualidad, la interoperabilidad entre sistemas informáticos se ha desarrollado considerablemente. Sin embargo, la interoperabilidad incide también en niveles de comunicación e intercambios gubernamentales; y sobre estos aspectos hoy en día los gobiernos enfrentan algunas dificultades al compartir e intercambiar información para sus procesos de negocios.

En los momentos actuales, la informática ha ido avanzando en temas relacionados con el intercambio de información y existen varias formas que ayudan a desarrollar la interoperabilidad de un software. Cada forma se implementa en dependencia a la arquitectura que presenta el sistema que será beneficiado.

## 1.4 Formas de interoperar

Para desarrollar la interoperabilidad entre sistemas integrales de gestión desde el punto de vista funcional y no conceptual, hay que contar con algún mecanismo que posibilite la construcción de una solución. Estos a su vez, son estrategias denominadas clases tecnológicas, las cuales tienen características propias para un determinado conjunto de restricciones y necesidades impuestas por la arquitectura. Las estrategias que pueden ser seleccionadas para desarrollar la interoperabilidad de un producto son: enlaces directos entre aplicaciones, software intermediario orientado a mensajes, o los servicios web XML.

**Enlaces directos entre aplicaciones.** Existen casos concretos, en los que se necesita primar el rendimiento frente a una amplia interoperabilidad, y por tanto se puede aceptar una integración más fuerte a cambio de una menor flexibilidad, donde una tecnología de enlace directo aplicación-aplicación es lo más adecuado. (Carriere, 2004)

Por lo general, los sistemas que exigen más rendimiento que flexibilidad e interoperabilidad son los mejores candidatos para conseguir la interacción. El concepto de enlace se refiere a una categoría de soluciones

entre las que se encuentran: la integración con la unidad central, la interoperabilidad .NET/COM (soportada por ejemplo por NET Framework), y enlaces de terceros tipo Java/COM o Java/.NET, así como protocolos a nivel de cable, específicos de ciertas aplicaciones y orientadas a un alto rendimiento, que son los casos más extremos. (Carriere, 2004)

**Software intermediario orientado a mensajes.** Una respuesta madura y a menudo viable para los retos de interoperabilidad es el Software Intermediario Orientado a Mensaje (MOM). Este proporciona un entorno de mensajes asíncronos para formar aplicaciones en relación productor-consumidor fuertemente integradas. La infraestructura de MOM puede ofrecer una solución de alto rendimiento, fiable y auditada para la interacción entre sistemas heterogéneos. Aunque esa infraestructura puede garantizar la consistencia de la interacción; no lo logra, si no se añaden otros componentes como motores de flujo de trabajo, o sin orquestar un sistema completo de procesos de negocio, capaz de gestionar adecuadamente las unidades funcionales con independencia de la plataforma y la aplicación. (Carriere, 2004)

Como se ha visto, una solución basada en software intermediario orientado a mensajes puede brindar una infraestructura más apropiada. En ocasiones, los enlaces directos entre aplicaciones demuestran ser la mejor solución si preocupa el rendimiento y el elevado nivel de integración. En cualquier caso, una estrategia de arquitectura orientada a servicios web, compuesta de servicios que cooperan en estrecha colaboración suele ser el modelo más flexible y ágil. La estrategia basada en estándares que representan los servicios web XML tiene el mayor potencial a largo plazo y está ganando una tremenda popularidad y aceptación.

**Servicios web XML.** Hoy en día existe un conjunto emergente, sencillo y aceptado de forma general, de estándares y tecnologías que permiten a las organizaciones conseguir el nivel de interoperabilidad que se necesita. Este conjunto de estándares y tecnologías se denomina de forma general servicios web XML. Estos ofrecen un medio común, independiente de la plataforma por el cual una serie de unidades atómicas de trabajo (servicios) se pueden integrar, agregar y organizar atravesando las fronteras de los sistemas y las organizaciones. Los servicios web XML han sido ampliamente reconocidos como una vía consistente para permitir la interoperabilidad, y XML es un medio auto-descriptivo que facilita el intercambio de información. Además, los servicios web XML suponen una solución sencilla, independiente de la tecnología, que evita en su mayor parte las dependencias de plataforma y fabricante. (Carriere, 2004)

Este trabajo tiene como objetivo fundamental, estandarizar el desarrollo de la interoperabilidad entre sistemas integrales de gestión, utilizando los servicios web XML como estrategia en la construcción de una plataforma de interoperabilidad.

## **1.5 Interoperabilidad en los sistemas integrales de gestión**

Primeramente hay que enfocarse en la siguiente pregunta; ¿qué son los sistemas integrales de gestión? Los sistemas integrales de gestión se ocupan de integrar, planificar y controlar los aspectos técnicos, humanos, organizativos, comerciales y sociales de una entidad u organización. Son la clave para la estrategia y la táctica del negocio. Permiten ahorrar costos en la medida que evita el reingreso de los datos en cada proceso, asegurando que todos los procesos manejan los mismos datos. Es clave que se verifique la calidad de los datos que se ingresan al sistema mediante el uso de validaciones de los datos.

Los sistemas integrales de gestión dentro de sus características fundamentales, contienen normas interrelacionadas de una empresa u organización por las cuales se administra de forma ordenada el funcionamiento de la misma. Además, deben tener la funcionalidad de poder intercambiar información. Este intercambio de información facilita el desarrollo del negocio y la comunicación con otros sistemas de gestión y por consiguiente se optimiza el intercambio de datos entre los componentes involucrados en el negocio.

Los sistemas integrales de gestión, se asemejan mucho en cuanto a estructura y tipos de módulos o componentes que poseen. Es por esto, que el desarrollo de una solución de interoperabilidad es una funcionalidad eminente para automatizar el intercambio de información entre estos sistemas.

### **1.5.1 Interoperabilidad del software en Cuba**

En Cuba, existen varios sistemas de gestión, pero estos carecen de una plataforma de interoperabilidad, mediante la cual puedan comunicarse y hacer más ágil todo el proceso de negocio. En la UCI hace ya algunos años se viene desarrollando un sistema integral de gestión. Este sistema no posee un producto que le posibilite el intercambio de información, a través de una plataforma con los demás sistemas integrales de gestión del país. Actualmente en la UCI se está desarrollando una plataforma de

interoperabilidad, capaz de proporcionar las funcionalidades para el desarrollo de la comunicación entre estos sistemas.

Lograr convertir el Sistema Integral de Gestión Cedrux en un ERP (Planificación de Recursos Empresariales), que cumpla con la funcionalidad de tener una plataforma de interoperabilidad para estandarizar la comunicación entre las empresas nacionales; puede servir como herramienta para la gestión económica de los recursos, no solo a nivel gubernamental sino también provincial o municipal. Este sistema lograría vencer las barreras de software que se tienen en las entidades presupuestadas o empresariales, para migrar hacia las nuevas posibilidades que brinda el software libre. Además, brindaría las funcionalidades necesarias para que las entidades puedan continuar desarrollando su función social sin problema alguno, mejoraría el control de los procesos y la eficiencia económica.

Este trabajo constituye una referencia en la búsqueda de una solución para la interoperabilidad del software, no solo en el Sistema Integral de Gestión Cedrux, sino en todos aquellos sistemas que deben contar con un producto que les permita interoperar sus componentes.

### **¿Por qué la necesidad de interoperar?**

La interoperabilidad es una característica muy importante que deben cumplir los sistemas integrales de gestión. Estos sistemas están compuestos por varios módulos: ventas, compras, producción, logística, gestión de activos, personal, finanzas, contabilidad, entre otros. Estos procesos en algún momento deben ser negociados con otros sistemas, por lo tanto, es necesario lograr una comunicación basada en el intercambio de datos e información y la única forma de promover esta interacción es interoperando.

Desarrollándose la interoperabilidad entre los sistemas integrales de gestión que existen en Cuba, se lograría estandarizar la comunicación; todas las conciliaciones de negocio se harían por esta vía y fuera muy ventajoso económicamente para las empresas y para el país. Interoperando, se podría absorber en un solo sistema la información contenida en los demás. De esta forma, se mantendrían más integrados en las actividades del negocio y la información que se intercambia estaría 100% automatizada y segura.

### **Ventajas de la interoperabilidad**

El desarrollo de la interoperabilidad, proporciona numerosas ventajas a los sistemas integrales de gestión.

A continuación se muestran algunos de sus principales beneficios, estos son:

- Los sistemas interoperan de forma transparente, coherente y segura.
- Logra la mayor cooperación entre sistemas.
- Ignora la heterogeneidad de los sistemas.
- Paradigma totalmente orientado hacia servicios existentes.
- Posibilidad de integrar sistemas en el futuro sin necesidad de grandes inversiones.
- Fortalece la integridad de la información y elimina su duplicidad.
- Permite la optimización de los procesos de interacción e intercambio de información entre las entidades.
- Cuando la tecnología permite la interoperabilidad, se fabrican componentes que son comunes entre muchos sistemas, y se fortalece la comercialización. Esto quiere decir que el nivel de costos del fabricante puede bajar y por tanto los precios finales con los que se venden a los usuarios.

## **1.6 Buenas prácticas para el desarrollo de la interoperabilidad**

En el mundo existen algunos proyectos de interoperabilidad estancados, principalmente porque no ven esta funcionalidad como un todo; sino como un problema de estandarización. Lo que ha ocurrido es que la mayoría de estos se han tenido que enfocar en los tres pilares fundamentales para el éxito de la interoperabilidad que son los procesos, los protocolos y taxonomías, además de la arquitectura tecnológica.

Para lograr la interoperabilidad dentro de una organización, se debe atender a sus dimensiones organizacionales, semánticas y tecnológicas como se expresó anteriormente. Pero existen algunas estrategias generales que pueden ser de utilidad, por ejemplo:

- Hacer uso de herramientas criptográficas, como los ofuscadores de código que se utilizan para cifrar el contenido de la información que viajara por la red.
- Buscar la accesibilidad de los contenidos y el conocimiento, con independencia de la tecnología que se use para acceder a ellos. (Acero, 2009)

- 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. (Acero, 2009)
- Siempre que sea posible, se debería optar por el uso de software libre. Así mismo, es frecuente que el software libre ayude a definir nuevos estándares abiertos y sus especificaciones están disponibles públicamente. (Acero, 2009)
- Si es posible, se deben usar aplicaciones o herramientas de desarrollo, que puedan funcionar en más de una plataforma tecnológica. En este sentido, las aplicaciones JAVA, o tecnologías como el XML, serán de gran ayuda para lograr los objetivos de interoperabilidad sin caer en las trampas de la dependencia tecnológica. (Acero, 2009)
- Las arquitecturas orientadas a servicios y las aplicaciones web, si se diseñan interoperables y basadas en estándares abiertos, permitirán independizar la plataforma tecnológica del cliente de la del servidor, lo que aumentará la neutralidad tecnológica de la organización y mejorará su capacidad para cambiar de plataforma tecnológica de los clientes, o del servidor, si se considerase necesario. (Acero, 2009)
- Si es posible, se debe utilizar un lenguaje común para que los diferentes sistemas integrales de gestión puedan interactuar e intercambiar información de manera óptima y eficiente. (Rivera, 2007)
- Se debe construir una plataforma como solución de interoperabilidad para el intercambio de información, independiente de hardware o software. (Rivera, 2007)
- Estar familiarizado con guías de buenas prácticas referidas al tema, para facilitar el desarrollo de una aplicación que permita gestionar la interoperabilidad entre sistemas. (XBRL España, 2005)
- Incluso el uso de una guía metodológica de interoperabilidad como esta, puede ser considerada como una buena práctica a la hora de interoperar un sistema integral de gestión, pues el objetivo de la misma es precisamente orientar el desarrollo de la interoperabilidad.

## 1.7 Arquitectura de software

La arquitectura de software es una disciplina joven en el mundo del desarrollo de software, pero se ha demostrado que representa un elemento de vital importancia dentro del ciclo vida de un proyecto de

software. Se ha convertido en un factor de vital importancia para lograr niveles altos de calidad. Poseer una buena arquitectura de software es de suma importancia, ya que ésta representa el corazón de todo producto software.

A continuación se muestran algunas definiciones que se dan sobre la misma en un orden cronológico, para que se vea la evolución que ha tenido:

“Estructura de los componentes de un programa o sistema, sus interrelaciones, y los principios y reglas que gobiernan su diseño y evolución en el tiempo.” (David Perry, 1995)

Más tarde se llegaría al concepto de que: “La arquitectura de software es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones.” (Paul, 1996)

Ya en el siglo XXI aparecen otra serie de definiciones, como por ejemplo: “La arquitectura de software de un programa o sistema de computación es la estructura o estructuras del sistema, que comprende elementos del software, las propiedades externamente visibles de los elementos y las relaciones existentes entre ellos.” (Reynoso, 2004)

Resumiendo, la arquitectura de software es el esquema de más alto nivel de la estructura de un sistema. Básicamente consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción de un sistema de información. Establece los fundamentos para que analistas, diseñadores y programadores trabajen en una línea común que permita alcanzar los objetivos del sistema de información, cubriendo todas las necesidades. Permitiendo la organización en todo el desarrollo, encarnada en sus componentes, las relaciones entre ellos y con su entorno, y los principios que gobiernan su diseño y evolución, define la organización fundamental de un sistema. Se selecciona y diseña en base a los objetivos y restricciones, constituye un puente entre el requerimiento y el código.

## Beneficios de la Arquitectura de Software

La arquitectura de software puede brindar muchas ventajas, principalmente si existe una buena elaboración de esta; es decir, una arquitectura que carezca de una justificación real detrás del desarrollo de cada aspecto que la compone, casi seguro llevaría a una solución sin un ciento por ciento de los beneficios que se plantean a continuación: (Billy, 2004)

- **Comunicación mutua:** La arquitectura de software representa un alto nivel de abstracción común que la mayoría de los participantes, si no todos, pueden usar como base para crear un entendimiento mutuo, formar un consenso y comunicarse entre sí.
- **Decisiones tempranas de diseño:** La arquitectura de software representa la encarnación de las decisiones de diseño más tempranas sobre un sistema.
- **Restricciones constructivas:** Una descripción arquitectónica proporciona modelos parciales para el desarrollo e indica los componentes y las dependencias entre ellos.
- **Reutilización, o abstracción transferible de un sistema:** La arquitectura de software personifica un modelo relativamente pequeño e intelectualmente tratable, de la forma en que un sistema se estructura y sus componentes se entienden entre sí.
- **Evolución:** La arquitectura de software puede exponer las dimensiones a lo largo de las cuales puede esperarse que evolucione un sistema.
- **Análisis:** Las descripciones arquitectónicas aportan nuevas oportunidades para el análisis, lo cual incluye verificaciones de consistencia del sistema, conformidad con las restricciones impuestas por un estilo, conformidad con atributos de calidad, análisis de dependencias y análisis específicos de dominio y negocios.
- **Administración:** La experiencia demuestra que los proyectos exitosos consideran una arquitectura viable como un logro clave del proceso de desarrollo industrial. La evaluación crítica de una arquitectura conduce típicamente a una comprensión más clara de los requerimientos, las estrategias de implementación y los riesgos potenciales.

Existen varios tipos de arquitecturas de software en el mundo, por lo que no es necesario inventar una nueva arquitectura para cada sistema de información. Lo habitual es adoptar una arquitectura conocida en función de sus ventajas e inconvenientes para cada caso en concreto. Dentro de las principales arquitecturas se encuentran: (Gastón Courtois, 2006)

- **Monolítica.** El software se estructura en grupos funcionales muy acoplados.
- **Cliente-servidor.** El software reparte su carga de cómputo en dos partes independientes, pero sin reparto claro de funciones.
- **Arquitectura de tres niveles.** Especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes (o capas) con un reparto claro de funciones: una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se encuentra modelado el negocio) y otra para el almacenamiento (persistencia). Una capa solamente tiene relación con la siguiente.
- **Arquitectura orientada a servicios web.** Es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio. Brinda una forma bien definida de exposición e invocación de servicios, lo cual facilita la interacción entre diferentes sistemas propios o de terceros.

El presente trabajo está orientado a guiar el desarrollo de una plataforma de interoperabilidad, mediante la adopción de una arquitectura orientada al desarrollo de servicios web. Los demás tipos de arquitecturas fueron mencionados porque es importante conocerlos, para entender por qué la arquitectura orientada a servicios web es la más favorable para la construcción de la plataforma.

## 1.8 Arquitectura orientada a servicios web

Una arquitectura orientada a servicios web es un conjunto de servicios interconectados cuyo objetivo es automatizar uno o varios procesos del negocio. Los servicios web son aplicaciones que utilizan estándares para el transporte, codificación y protocolo de intercambio de información. Permiten la intercomunicación entre sistemas de cualquier plataforma y se utilizan en una gran variedad de escenarios de integración, tanto dentro de las organizaciones como con los socios del negocio. Dentro de los principales beneficios que trae consigo el uso de los servicios web, se pueden mencionar que:

- **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. Esta interacción requiere un documento Lenguaje de Descripción de Servicios Web (WSDL) para definir la interfaz y describir el servicio, junto con un protocolo de red, generalmente HTTP. (Lebrún, 2007)

- Disminuyen el tiempo de desarrollo de las aplicaciones.
- Minimizan el costo de implementación.
- Aumentan la productividad debido al incremento de fluidez en las relaciones entre proveedores, socios, empleados y clientes.
- Proporcionan una mejor gestión, mantenimiento y actualización de la información.
- Evitan problemas con los cortafuegos.
- Proveen independencia cliente-servidor. Integración Negocios para Negocios (B2B). Esta es la principal ventaja de los servicios web.

## 1.9 Estándares de interoperabilidad

### ¿Qué es un estándar?

Primeramente hay que tener bien claro la respuesta a la pregunta anterior. Dentro del ámbito de las tecnologías de información, los estándares están orientados a facilitar el arte del desarrollo de esquemas de integración a todo nivel, siendo esta tendencia un reflejo de la situación política, económica y social que se está estableciendo en el mundo.

La interoperabilidad se basa en estándares, que consisten en definiciones, formatos o procesos que han sido aprobados por unas determinadas organizaciones de estandarización o aceptados “de facto” como tales por la industria. (Borges, 2007)

En conclusión, un estándar es un modelo a seguir. Un estándar de interoperabilidad se especifica como tal en aspectos relacionados con la comunicación de procesos, y en la manera que se va a regir la organización y estructura de todo este trayecto a la comunicación y el intercambio entre sistemas.

### Estándares a nivel mundial

Un estándar para el manejo de datos establece un sistema común de terminología y de definiciones para documentar datos. Lógicamente, las estructuras y definiciones de metadatos deben tener su referencia en un estándar. Hay muchos y variados estándares de metadatos disponibles. La razón de que existan tantos estándares es que los metadatos se emplean para diversos aspectos. (CULIACAN, 2005)

Dentro de los principales beneficios que brindan los estándares de forma general, se puede hacer énfasis en que: (CULIACAN, 2005)

- Los estándares permiten la localización rápida de cierto elemento. Si se utiliza un estándar, encontrar la información específica en un catálogo de metadatos será mucho más fácil que si no se utiliza ningún estándar.
- Los estándares permiten búsquedas automatizadas. Cuando se utilizan los estándares, las computadoras se pueden programar permitiendo buscar y encontrar conjuntos de datos útiles.
- Un beneficio de los estándares es que se han generado a través de un proceso de consulta con otros expertos y ofrecen una base a partir de la cual pueden desarrollarse perfiles nacionales u orientados de acuerdo con materias.
- Ayudan a minimizar la duplicación de esfuerzos en la elaboración, recolección, procesamiento o distribución de la información.

Existen numerosos estándares en el mundo que se utilizan para el manejo de datos e información. El uso de cada uno de ellos depende de lo que se desea hacer, y del tipo de información que se necesita intercambiar o comunicar. Algunos de los estándares que se deben mencionar son:

- El MARC (Catalogación Legible por la Máquina) se utiliza para la catalogación bibliográfica y define catálogos de fichas. Entre sus principales ventajas se encuentra su gran nivel de usabilidad. Empleando el MARC muchas bibliotecas pueden tener acceso a los programas comerciales disponibles para manejar las operaciones de una biblioteca. La norma MARC permite reemplazar un sistema por otro haciendo que los datos sean compatibles.
- ODBC (Estándar de conectividad Abierta de Base de Datos) es usado para la conectividad abierta de bases de datos. Es un estándar de acceso a bases de datos desarrollado por la corporación Microsoft. Su principal objetivo es hacer posible acceder a los datos desde cualquier aplicación, sin importar qué sistema gestor de bases de datos se utilice.
- EML (Lenguaje de Metadatos Ecológicos) es un estándar basado en XML para describir datos ecológicos. Consiste en un lenguaje común implementado en XML que permite describir, archivar y transportar datos. Entre sus características destacan, su carácter flexible y extensible, además de que incluye diferentes elementos de otros estándares, lo cual permite cierta interoperabilidad con ellos.

- XML es un lenguaje de etiquetado, fundamental en el intercambio de una gran variedad de datos. Solo puede asegurar una comunicación básica entre aplicaciones y se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas.
- XBRL (Lenguaje Extensible de Informes de Negocios) es usado para generar y transmitir operaciones electrónicas y crear reportes en formato digital, personalizando las etiquetas y metadatos correspondientes a información de negocio de carácter financiero u operacional. Además, automatiza y facilita la interoperabilidad de información entre aplicaciones computacionales distintas.

Los estándares mencionados anteriormente, pueden ser utilizados para la construcción de una plataforma de interoperabilidad. Cada uno tiene características diferentes, lo cual puede ser aprovechado para intercambiar los datos contenidos en cada proceso dentro de un sistema integral de gestión. Se propone la utilización del estándar XML como base tecnológica, para de esta forma lograr la comunicación básica entre los sistemas involucrados durante el proceso de interoperabilidad.

### **1.10 Taxonomías**

A principios de los años 90 del siglo XX, el concepto de taxonomía se incorpora a diversos ámbitos del conocimiento, como la psicología, las ciencias sociales y la informática. En este último, se introduce para designar casi todos los sistemas de acceso a la información que intentan establecer coincidencias entre la terminología del usuario y del sistema. Los primeros especialistas que desarrollaron sistemas de organización de contenidos para la web, formaban parte del área de consultoría en gestión del conocimiento. Estos procedían de ambientes próximos a la informática y la ingeniería-gestión de contenidos y arquitectura de la información; ellos no conocían la tradición de los lenguajes documentales de la bibliotecología y la documentación, asignando el término taxonomía para los sistemas que desarrollaban. Este término se mantiene en uso actualmente para designar los sistemas de organización de contenidos en el contexto de Internet, aunque la teoría y la práctica de los lenguajes documentales se han aplicado de forma intensiva en este contexto. (Hernández, 2007)

En el campo de la informática si se fuera a dar la definición más sencilla posible de una taxonomía, se señalaría que son sistemas de organización de contenidos. Las taxonomías actúan categorizando los

elementos (los datos) en función de la relación que tienen entre sí, evidenciando las relaciones jerárquicas de los mismos.

Resumiendo, una taxonomía es un tipo de vocabulario controlado en que todos los términos están conectados mediante algún modelo estructural (jerárquico, arbóreo...) y especialmente orientado a los sistemas de navegación, organización y búsqueda de contenidos de los sitios web. (Centelles, 2005)

### **Taxonomías existentes**

Existen muchos tipos de taxonomías, y aunque todas tienen un mismo propósito, no actúan de igual forma, ni presentan las mismas características. Las taxonomías son construidas con un fin determinado y van a accionar según al campo que estén dirigidas.

En la informática, como resultado del proceso de estandarización se obtiene una taxonomía de elementos base, que no es más que la categorización de estos en función de la relación que tienen entre sí, evidenciando las relaciones jerárquicas de los mismos. En ocasiones, el proceso de estandarización uniforme para cualquier país es imposible pues contienen normas propias, las cuales, si fueran todas partes de una sola taxonomía la harían complicada y difícil de implementar. Es por ello que se definen taxonomías generales y pautas sobre las cuales se pueden crear nuevas catalogaciones de los conceptos específicos del área.

Por este motivo existen diversas taxonomías que son reconocidas dentro de un mismo estándar y esto permite que luego del reconocimiento por su organización internacional, sean asimiladas dentro de los sistemas informáticos en que se comunican.

Por ejemplo dentro del estándar XBRL, se encuentran una serie de clasificaciones de taxonomías. Las taxonomías aprobadas oficialmente por la asociación XBRL Internacional, se publican en la web bajo una de las categorías siguientes: taxonomías reconocidas, taxonomías aprobadas y taxonomías en desarrollo.

### **1.11 Seguridad. Formas de proveer seguridad**

Se puede entender como seguridad una característica de cualquier sistema o elemento que indica que está o no libre de todo peligro, daño o riesgo y que es en cierta manera infalible.

En los sistemas informáticos, la seguridad es muy difícil de conseguir (según la mayoría de expertos, imposible); es entonces cuando el término seguridad, tiende a convertirse en fiabilidad (probabilidad de que un sistema se comporte tal y como se espera de él). Esto equivale a que se hable más de sistemas fiables, en lugar de sistemas seguros.

De forma general, puede considerarse la seguridad informática como una disciplina que se encarga de proteger la integridad y la privacidad de la información almacenada en un sistema informático, o manejada por un sistema informático; es decir la seguridad se refiere al mantenimiento íntegro, confidencial y auténtico de los datos.

## **Formas de proveer seguridad**

Existen numerosas formas de proveer seguridad en las comunicaciones y transporte de los datos, mediante la criptografía simétrica y asimétrica, firmas digitales y certificados electrónicos. También se les puede ofrecer protección a los datos a través de cortafuegos, la autenticación y los antivirus. Existen tres reglas básicas que se deben cumplir si se quiere proveer seguridad, estas son:

- La seguridad debe cubrir todos los huecos. Se trata por tanto de delimitar con precisión todos los aspectos de seguridad que se deben cubrir y prever todas las contingencias que se puedan. La máxima seguridad que se posee es la del elemento más débil del sistema.
- Los niveles de seguridad deben ser adecuados al entorno ¿Qué medidas de seguridad toman los demás? ¿Cómo se protegen los que tienen un sistema parecido? Las medidas de protección deben ser solo un poco superiores, a las que son normales en el entorno. Una medida de seguridad muy superior a lo normal no solo será muy costosa, sino que puede llegar a ser desfavorable.
- La seguridad debe ser adecuada a la necesidad de protección de lo asegurado y a los recursos disponibles. Se trata de hacer una valoración de riesgos y de los costes para la protección, de forma que en ningún momento los costes superen los riesgos.

La solución que brinda el presente trabajo, es una guía metodológica para el desarrollo de una plataforma de interoperabilidad. Para mantener la seguridad de la información que es intercambiada a través de esta, se debe adoptar una de las formas de seguridad mencionadas anteriormente. Es válido señalar que dicha

forma tiene que cumplir con las reglas básicas indicadas, para de esta forma brindar la protección adecuada a los datos como elemento más vulnerable.

## **1.12 Entornos de desarrollo**

La mayoría de las herramientas, lenguajes y tecnologías que forman parte en el desarrollo del presente trabajo de diploma, fueron seleccionados por la dirección de desarrollo de software del proyecto Cedrux, y están definidas en su manual del marco de trabajo. Todos los demás aspectos que formaron parte de las herramientas que se utilizaron como parte de la construcción de la solución de interoperabilidad, fueron evaluados y seleccionados debido a sus características funcionales para el producto final y sus ventajas para el equipo de desarrollo.

### **1.12.1 Lenguajes**

En la actualidad Internet es la vía de comunicación más usada mundialmente por las ventajas y funcionalidades que ofrece, a partir de las aplicaciones que soporta y al número de usuarios con que cuenta.

Existen un conjunto de lenguajes de programación que proporcionan a las aplicaciones web gran interactividad, ya sea del lado del cliente o del servidor. Para software libre entre las técnicas de desarrollo web las más usadas del lado del cliente, encargadas de visualizar y comprobar la información en el navegador y los formularios, están el HTML, Java Script y Ajax.

Otra tecnología muy usada es XML, un lenguaje etiquetado especializado en la construcción de soluciones de comunicación y en la interoperabilidad entre sistemas. Este estándar es el escogido por el equipo de desarrollo para asegurar la comunicación básica en la solución de interoperabilidad.

- **XML**

Se recomienda el uso de XML porque además de ser un lenguaje, es un estándar que se establece como base tecnológica para la interoperabilidad entre sistemas integrales de gestión. Representa un modo de definir lenguajes para diferentes necesidades. Está diseñado especialmente para los documentos de la web, no sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de

información estructurada entre diferentes plataformas. Su carácter extensible permite que después de diseñado y puesto en producción, se le puedan adicionar nuevas etiquetas.

XML posibilita que los diseñadores creen sus propias etiquetas, lo cual hace posible la definición, transmisión, validación e interpretación de datos entre aplicaciones y organizaciones. Su principal característica consiste en permitir compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes de una manera segura, fiable y fácil. Así pues, XML desempeña un papel primordial en el mundo actual, que tiende a la globalización y la compatibilidad entre los sistemas. (Álvarez, 2001)

- **PHP**

Entre los lenguajes que se ejecutan en el servidor y procesan la lógica del negocio se puede encontrar PHP, JAVA, ASP y Perl. El equipo de desarrollo adopta PHP, debido a que es el lenguaje definido en el marco de trabajo para el desarrollo de soluciones de este tipo en el proyecto. El componente de interoperabilidad está desarrollado con el lenguaje PHP y la construcción de la plataforma de interoperabilidad tiene que adoptar PHP como lenguaje del lado del servidor. No obstante, sus características lo hacen ser el lenguaje de programación web más usado en el mundo, y estas son también ventajas para el equipo de desarrollo que busca la solución de interoperabilidad. Las características más significativas del lenguaje PHP, son las siguientes:

- Alto rendimiento: PHP es muy eficiente, mediante el uso de un único servidor, puede brindar millones de acceso al día.
- Integración de base de datos: PHP dispone de una conexión propia a todos los sistemas de base de datos. Además de MySQL, puede conectarse directamente a las bases de datos de PostgreSQL, mSQL, Oracle, Informix, InterBase y Sybase, entre otras.
- Bibliotecas incorporadas: Como se ha diseñado para su uso en la web, PHP incorpora una gran cantidad de funciones integradas para realizar útiles tareas.
- Bajo coste: PHP es gratuito.
- Facilidad de aprendizaje y uso.
- Portabilidad: PHP está disponible para una gran cantidad de sistemas operativos diferentes. Se puede escribir código PHP en todos los sistemas operativos gratuitos del tipo Unix, como Linux y Fedora, versiones comerciales de Unix o en las diferentes versiones de Microsoft Windows.

- Acceso a código abierto: A diferencia de los productos comerciales de código cerrado, con PHP se pueden realizar modificaciones a los programas con toda libertad.

## 1.12.2 Herramientas

- **Framework**

Un framework es una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente puede incluir soporte de programas, librerías y otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Define una filosofía de trabajo, proporciona funciones que facilitan las acciones del programador, ahorra trabajo y tiempo, produce aplicaciones más fáciles de mantener, evita el código duplicado y puede crear aplicaciones Multi-Capa. Además, agrega funcionalidad extendida a un lenguaje de programación, esta automatiza muchos de los patrones de programación para orientarlos a un determinado propósito.

Existen numerosos framework que se pueden usar para la construcción de una solución de interoperabilidad, dentro del marco de trabajo del sistema Cedrux, se destacan: EXT, Doctrine PHP y Zend Framework.

### **Zend Framework**

Se trata de un framework para el desarrollo de aplicaciones y servicios con PHP, que brinda soluciones para construir sitios web modernos, robustos y seguros. Es código abierto y trabaja con PHP5. Entre sus principales características figuran las siguientes: (Garnet, 2007)

- Trabaja con MVC (Modelo Vista Controlador)
- Cuenta con módulos para manejar archivos PDF y servicios web.
- El marco de Zend también incluye objetos de las diferentes bases de datos, por lo que es extremadamente simple para consultar su base de datos, sin tener que escribir ninguna consulta SQL.
- Una solución para el acceso a base de datos que balancea el ORM (Mapeo Objeto-Relacional) con eficiencia y simplicidad.
- Completa documentación y pruebas de alta calidad.

- **Entorno de desarrollo integrado(IDE)**

Existen numerosos entornos de desarrollo como Aptana, Eclipse, Zend Studio, entre otros. El seleccionado por el equipo de desarrollo ha sido Zend Studio, por las características que presenta y facilidad de uso.

## **Zend Studio**

Este divide sus funcionalidades en dos partes: la del cliente y la del servidor, las cuales se instalan por separado. La parte del cliente contiene la interfaz de edición y la ayuda, además permite hacer depuraciones simples de scripts. La parte del servidor instala Apache y el módulo PHP, o los configura para trabajar juntos en depuración en caso de que ya estén instalados.

Zend Studio dispone de una herramienta muy interesante de depuración, que permite ejecutar páginas y conocer en todo momento el contenido de las variables de la aplicación, las variables del entorno, las recibidas por formulario o en la sesión. (Álvarez, 2003)

- **Herramientas Cases**

Para la modelación de la plataforma de interoperabilidad se pueden utilizar como herramientas Cases: el Rational o el Visual Paradigm, ambas con grandes posibilidades y comodidades de trabajo. Visual Paradigm es la herramienta seleccionada por el equipo de desarrollo para el diseño de la plataforma de interoperabilidad.

## **Visual Paradigm**

Aunque es un software propietario; la UCI posee una licencia para el trabajo con dicha herramienta, ya que es una de las más potentes que existen en la actualidad, multiplataforma y orientado a BPM (Gestión de Procesos de Negocio). Visual Paradigm proporciona soporte al modelado visual con UML 6.0 (Lenguaje Unificado de Modelado), ofreciendo distintas perspectivas del sistema independiente de la plataforma y dotada de una buena cantidad de productos o módulos para facilitar el trabajo durante la confección de un software, así como garantizar la calidad del producto final.

Posee entre sus principales características la posibilidad de crear un amplio conjunto de artefactos, utilizados con frecuencia durante la confección de software. Posibilita generar código a partir de los diagramas, para plataformas como .Net, Java y PHP.

- **Sistema Gestor de Bases de Datos**

Un SGBD (Sistema Gestor de Bases de Datos), es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. Algunos ejemplos de SGBD son Oracle, DB2, PostgreSQL, MySQL, MS SQL Server. Un SGBD debe permitir: (CAVSI, 2009)

Para la construcción de la plataforma de interoperabilidad, el equipo de desarrollo utiliza PostgreSQL, por ser el que está definido en el marco de trabajo del proyecto Cedrux.

## **PostgreSQL**

Es considerado el sistema gestor de base de datos código abierto más avanzado del mundo, gratuito, el más usado en la comunidad de desarrollo de la UCI y se integra perfectamente con PHP. Propone un tamaño ilimitado para las bases de datos, lo que da la medida de un gestor de bases de datos robusto, y con grandes funcionalidades.

- **Control de versiones**

Para el control de versiones se pueden usar herramientas como el TortoiseSVN y el Subversión. El cual ha sido el seleccionado por el grupo de desarrollo de la plataforma de interoperabilidad.

## **Subversión**

Es un software de sistema de control de versiones diseñado específicamente para reemplazar al popular CVS (Sistema Concurrente de Versiones), producto a las numerosas deficiencias de este último. Es un software libre bajo una licencia de tipo Apache/BSD y se le conoce también como SVN (Subversión) por ser ese el nombre de la herramienta de línea de comandos. Una característica importante de este, es que

los archivos versionados no tienen cada uno un número de revisión independiente. Las principales ventajas que provee el uso de esta herramienta son:

- La creación de ramas y etiquetas es una operación muy eficiente. Tiene costo de complejidad constante ( $O(1)$ ).
- Maneja eficientemente archivos binarios.
- Los programas asociados a Subversión que se ejecutan por línea de comandos, pueden ejecutarse tanto en plataformas Unix, Linux, Solaris o Microsoft Windows.

- **Editores de taxonomía**

Algunos de los editores de taxonomías más importantes para la solución y el alcance de este trabajo, son el UBmatrix XBRL Taxonomy Designer, el CoreFiling SpiderMonkey y el Fujitsu XWand; este último ha sido el editor de taxonomías seleccionado por el equipo de desarrollo, por sus facilidades de trabajo.

## **Fujitsu XWand**

Permite crear nuevas taxonomías y documentos instancias, cumpliendo con la especificación XBRL 2.1, así como validar las mismas y posibilitar ver las instancias creadas desde un navegador web. Presenta además abundante bibliografía, y es uno de los más usados en la actualidad.

### **1.13 Conclusiones parciales**

En el presente capítulo se explica la fundamentación teórica del trabajo, la cual constituye la base de soporte para el desarrollo de la guía metodológica que orientará la construcción de la plataforma de interoperabilidad, como solución para el intercambio y la interacción entre sistemas integrales de gestión en Cuba.

Se realiza un estudio sobre los principales aspectos relacionados con la interoperabilidad entre sistemas integrales de gestión. Se define la arquitectura a utilizar para la construcción de la solución, donde se especifica el uso de una arquitectura orientada a servicios web para el desarrollo de la plataforma de interoperabilidad. Se abordan temas significativos como los estándares propuestos, donde define XML como base tecnológica. Entre otros aspectos de importancia, se valora y se justifica el por qué de la selección de las herramientas y lenguajes a utilizar para el desarrollo de la plataforma de interoperabilidad.

## **CAPÍTULO 2: GUÍA METODOLÓGICA**

### **2.1 Introducción**

El presente capítulo es el resultado de un estudio general enfocado de forma organizada y bien estructurada, donde se especifica un orden lógico para lograr desarrollar la interoperabilidad entre sistemas integrales de gestión en Cuba. Tiene como objetivo guiar y orientar la construcción de una plataforma, que sirva como solución para las actuales dificultades de interoperabilidad existentes entre los sistemas integrales de gestión del país.

En el capítulo se abordan temas implícitos dentro de cada paso, como: identificar la estructura metadatos, selección de estándares de interoperabilidad, la arquitectura, especificación de las reglas de interoperabilidad, el esquema de seguridad, así como aspectos relacionados con la implementación del producto.

### **2.2 Guía metodológica**

A continuación se describen los aspectos necesarios que conforman la guía metodológica de interoperabilidad para sistemas integrales de gestión en Cuba. Estos aspectos se encuentran estructurados en una serie de pasos que dan un mayor entendimiento y organización al trabajo; ayudando a encontrar eficazmente el camino a la construcción de una solución con calidad.

#### **Paso 1: Identificar los procesos a interoperar, así como el orden de prioridad de los mismos**

El primer paso para interoperar con sistemas externos, o sea, exportar e importar información, así como notificar o gestionar sucesos para de esta forma lograr la comunicación y el intercambio entre sistemas; es definir los procesos y subprocesos a interoperar así como el orden de prioridad de los mismos.

Dígase de exportar información, aquella funcionalidad que realizará un usuario a través de la plataforma de interoperabilidad de enviar los datos seleccionados a un sistema destino. De similar forma se importa, solo hay que determinar la información a importar, dar la orden de ejecutar y por último recepcionar los datos.

En la identificación de los procesos y subprocesos, los especialistas funcionales son los encargados de seleccionar cuáles de estos van a interoperar con otros sistemas, ya sea por la importancia que tienen para el negocio, o por la necesidad de que intercambien algún tipo de información. Algo parecido sucede cuando se define el orden de prioridad.

Es importante destacar que la identificación de procesos y subprocesos no se realiza solo por parte de los que participan en un sistema, sino por todos los que de alguna forma intervienen en el desarrollo de la interoperabilidad. Así mismo, constituye el motor de arranque para la construcción de una solución, y el fruto del acuerdo que se establece entre todos los sistemas que van a intercambiar información.

Para identificar los procesos a interoperar, los especialistas involucrados se apoyan en un Mapa de Procesos que debe existir como artefacto en los sistemas integrales de gestión. Con el uso de este artefacto se obtiene una visión general del sistema, donde se representan todos sus procesos y la relación que existe entre ellos. Haciendo uso del Mapa de Procesos y a partir de la importancia que tenga cada proceso para el negocio, se selecciona el orden y la prioridad de cada uno de estos para interoperar.

- **Procesos:** Un proceso de negocio es un conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio definido. (España, 2007)
- **Subprocesos:** Un subproceso es parte de un proceso de mayor nivel que tiene su propia meta, propietario, entradas y salidas. (Rico, 2003)

### **Paso 2: Arquitectura**

La arquitectura definida por un equipo de desarrollo tiene como objetivo fundamental dirigir y reglamentar la construcción de una solución.

Una arquitectura es un entramado de componentes funcionales que aprovechando diferentes estándares, convenciones, reglas y procesos, permite integrar una amplia gama de servicios, de manera que pueden ser utilizados eficazmente para obtener una solución.

Según el estándar IEEE1471-2000 (Instituto de Ingenieros Eléctricos y Electrónicos): “La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos, el ambiente y los principios que orientan su diseño y evolución”.

El concepto de arquitectura se usa de forma amplia y en campos muy distintos, por lo que su significado es algo complicado. En el campo del software, como se ha mencionado anteriormente, la arquitectura identifica los elementos más importantes de un sistema así como sus relaciones y su principal función es brindar una visión global del sistema.

¿Por qué entonces es importante la arquitectura? Debido a que se necesita para entender el sistema, organizar su desarrollo, plantear la reutilización del software y hacerlo evolucionar. Las arquitecturas software no responden únicamente a requisitos estructurales, sino que pueden estar relacionadas con aspectos de rendimiento, usabilidad, reutilización, restricciones económicas y tecnológicas, e incluso cuestiones estéticas. (Pérez, 2004)

Por lo dicho anteriormente, es importante definir una adecuada arquitectura de software, ya que ésta es el corazón de todo sistema informático y determina cuáles serán los niveles de calidad asociados al desarrollo de la plataforma de interoperabilidad.

Existen muchos tipos de arquitecturas de software, por lo que no es necesario inventar una nueva arquitectura para construir la plataforma de interoperabilidad. Lo habitual es adoptar una conocida en función de sus ventajas e inconvenientes para el caso en cuestión. Este trabajo guía el desarrollo de una solución basada en una arquitectura orientada a servicios web, debido a que es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio. Además, brinda una forma bien definida de exposición e invocación de servicios, facilitando la interacción entre diferentes sistemas propios o de terceros; por lo cual, es la más favorable para la construcción de la plataforma.

La arquitectura que se define para el desarrollo de la plataforma de interoperabilidad, consta de un elemento primordial, los servicios que se consumen durante la implementación.

### **Definición de los servicios que se necesitan para interoperar**

La arquitectura tecnológica está orientada a servicios web y para la implementación de la plataforma de interoperabilidad se deben definir los servicios o tareas que se necesitan.

Los servicios web son piezas de una lógica de negocio, localizadas en alguna parte de Internet o Intranet, que son accesibles a través de protocolos de comunicación, como es el caso de HTTP. Este tipo de servicios, facilita la interconexión de cualquier número de aplicaciones o componentes software para que puedan interoperar, compartiendo información entre organizaciones o individuos que estén interesados en los datos, mediante una plataforma común y un lenguaje definido.

Como se ha mencionado anteriormente, la arquitectura a utilizar para la construcción de la plataforma es orientada a servicios web; y contará con las funcionalidades de exportar e importar datos, las que serán publicadas como servicios web al alcance de otros sistemas existentes en el país que necesitan interoperar.

Para construir esta solución, se consumen otros servicios que no son web precisamente, son simplemente servicios internos del proyecto, los cuales van a ser seleccionados para la implementación de los servicios web que brindará finalmente la plataforma como funcionalidad principal. Durante el proceso de selección de los servicios o tareas que se necesitan para interoperar, hay que tener bien claro cuales se van a utilizar. No es significativa la cantidad de servicios que se utilicen, sino las funcionalidades que deben cubrir a la hora de ser usados.

Otra opción en el uso de este tipo de servicios, es la siguiente: Puede que en cierto sistema donde se vaya a construir una plataforma de interoperabilidad, él o los desarrolladores encargados de implementar la plataforma decidan ellos mismos implementar los servicios que son necesarios para lograr la solución, para de esta forma tener un mejor control y entendimiento de su trabajo. Esto puede suceder y también constituye una posible solución, siempre y cuando estén bien definidos y desarrollados, para que no existan conflictos internos al ser utilizados.

La última opción, es que los desarrolladores sean capaces de lograr construir la plataforma de interoperabilidad sin una previa realización de los servicios, o sin utilizar servicios ya existentes. Esta opción puede funcionar, pero no cumple mucho objetivo, debido a que si ya existen estos servicios son precisamente para ser utilizados.

Los servicios que se van a consumir para implementar la plataforma de interoperabilidad, generalmente están publicados en algún lugar dentro de cada sistema integral de gestión que se encuentra

desarrollando su solución de interoperabilidad. Puede ser un componente con la funcionalidad de cumplir con esta tarea, o simplemente una cartera de servicios donde se encuentran publicados al alcance de los demás componentes dentro del sistema. Para utilizarlos, solo se debe acceder a ellos, para posteriormente consumirlos y generar las funcionalidades que se desean.

### **Paso 3: Estándares**

La selección del estándar a utilizar, es de acuerdo con las características, las ventajas y el impacto que brinda el componente a interoperar. Siempre a partir del estándar XML como base tecnológica.

### **Soporte tecnológico**

XML es una tecnología en auge y cada vez se está utilizando más en las organizaciones. Pero ¿qué es realmente XML?, ¿qué se puede hacer con él? y ¿qué beneficios aporta? XML es un fichero de texto plano que contiene una serie de datos. Esta información está estructurada siguiendo una gramática adecuada, válida según el estándar de definición XML que entienden los diferentes sistemas gestores de datos. Es decir, es un archivo que contiene información organizada de forma concreta compatible con la gran mayoría de las bases de datos. La información se puede utilizar en general para ser compartida o intercambiada con otros sistemas, y en particular para servir a otros entornos que puedan tratarla a su gusto en función de sus necesidades.

De esta manera, y desde un punto de vista más técnico, se puede considerar el lenguaje XML como el estándar perfecto para compartir e intercambiar información entre sistemas distribuidos y heterogéneos; es decir, para lograr la comunicación básica entre ellos.

### **¿Para qué se utiliza XML?**

XML se utiliza como lenguaje intermedio para intercambiar información (obtenerla o servirla). Una vez que se dispone de la información, se puede realizar una aplicación para gestionar y mantener dichos datos, como puede ser una aplicación web interna. Pero la idea es ir más allá y no quedarse solo en la obtención y visualización de datos; el objetivo fundamental es poder manipularlos y almacenarlos de nuevo en sus sistemas (distribuidos en la red) desde una sencilla aplicación. Así, de la misma manera que se pueden

obtener datos de distintos sistemas, se puede realizar la operación inversa y alimentar los sistemas a partir de un conjunto XML de información.

Un beneficio que se puede obtener utilizando tecnologías XML, es obtener información en tiempo real y de manera simultánea de diferentes entornos o sistemas de información, independientemente de la naturaleza y la compatibilidad entre estos.

¿Por qué XML se puede utilizar para el desarrollo de servicios web? Dentro de los principales aspectos que influyen, se puede citar que: (Juan Garbajosa Sopeña, 2008)

- Es un estándar abierto, muchas compañías tecnológicas integran en su software compatibilidad con dicho lenguaje. Esta compatibilidad está dada: en los software de escritorio, sistemas operativos y aplicaciones móviles que permiten la relación con XML, haciéndolo muy potente a la hora de permitir la comunicación entre distintas plataformas de software y hardware.
- Simplicidad de sintaxis, esto quiere decir que es muy fácil escribir código en XML.
- Independencia del protocolo de transporte. XML es un lenguaje de marcado de texto, por lo cual no necesita de ningún protocolo de transporte especial, solo requiere un protocolo que pueda transferir texto o documentos simples. Actualmente en el mercado de software, existen muchos protocolos con estas características, como los conocidos: HTTP y SMTP (Protocolo Simple de Transferencia de Correo Electrónico). Precisamente una de las características de los servicios web es su independencia de protocolo de transporte.

### **Principales características de XML**

Las características más notables del estándar XML se manifiestan en los siguientes aspectos: (Juan Garbajosa Sopeña, 2008)

- Mayor consistencia, homogeneidad y amplitud de los identificadores descriptivos del documento con XML con los RDF (Marco de Descripción de Recursos), en comparación a los atributos de la etiqueta del lenguaje HTML.
- Integración de los datos de las fuentes más diversas. Se podrá hacer el intercambio de documentos entre las aplicaciones tanto en el propio ordenador, como en una red local o extensa.

- La extensibilidad y flexibilidad de este lenguaje permite agrupar una variedad amplia de aplicaciones, desde páginas web hasta bases de datos.
- Los motores de búsqueda devolverán respuestas más adecuadas y precisas, ya que la codificación del contenido web en XML consigue que la estructura de la información resulte más accesible.
- El concepto de hipertexto se desarrollará ampliamente.
- Exportabilidad a otros formatos de publicación (papel, web, CD-ROM, entre otros).

### **Estándares para el proceso**

La selección del estándar para interoperar un componente específico, como se ha dicho anteriormente depende de las características particulares de este, por ejemplo:

- Si se quiere intercambiar información de carácter financiero y de negocio se seleccionaría un estándar denominado XBRL, el cual va mas allá de la comunicación básica entre aplicaciones que establece un estándar como XML.
- Si lo que se desea es describir datos ecológicos, se usa en estándar basado en XML conocido como EML.
- Sin embargo, para intercambiar información entre aplicaciones en entornos descentralizados y distribuidos, se utiliza el protocolo estándar basado en XML: SOAP (Protocolo de Acceso a Objetos Simples). Este ofrece ventajas, por ejemplo: no depende de un sistema en concreto, puede comunicar aplicaciones que estén escritas en cualquier lenguaje, atraviesa cortafuegos corporativos y permite la interoperabilidad entre distintas aplicaciones. (Isaza E., 2007)
- Además, se puede intercambiar metainformación entre herramientas de modelado, basadas en UML y repositorios de metainformación basadas en MOF (Manual de Organización y Funciones); en distribuidos entornos heterogéneos, utilizando el XMI (XML Intercambio de Metadatos).

Para seleccionar los estándares que se necesitan en la construcción de la plataforma de interoperabilidad, hay que enfocarse en el proceso a interoperar, analizar su funcionamiento, analizar los estándares que se proponen y que se especializan en el tratamiento de los datos, para posteriormente seleccionar el adecuado para el desarrollo de la solución.

### Paso 4: Estructura metadatos

Una estructura metadatos de los conceptos a interoperar, proporciona información organizada y como su nombre lo indica, estructurada, que caracteriza a los datos. Se realiza con el objetivo de organizar los datos a exportar e importar, es decir, la información que va a ser intercambiada. Esta información en cierto momento ya fue estandarizada por algunas personas, principalmente por especialistas funcionales de los distintos sistemas existentes, en coordinación con los involucrados con el proceso que va a interoperar.

En ocasiones algún involucrado con el proceso, hace la propuesta de la estructura metadatos, pero son los especialistas funcionales los encargados de conciliarla. Estableciéndose como acuerdo entre los distintos sistemas la utilización de dicha estructura.

Para desarrollar la estructura metadatos se puede hacer uso del Modelo Conceptual, el cual es un artefacto que debe estar definido en los sistemas integrales de gestión para cada subsistema. Haciendo uso del Modelo Conceptual se facilita la comprensión de los conceptos que conforman los procesos que van a interoperar, dentro de cada subsistema. Además, explican la lógica de cada concepto, teniendo en cuenta otros elementos, como son: los atributos, roles y relaciones.

Una de las herramientas utilizadas para construir una estructura metadatos para un proceso en específico es un gestor de plantilla, el cual se encarga de definir visualmente esta estructura. Para una mejor comprensión de lo que es una estructura metadatos, a continuación se muestra el Ejemplo 1.

- **Estructura metadatos:** Información altamente estructurada que describe características de los datos. (Ricardo Correa F., 2009)
- **Conceptos:** Información contenida dentro de un objeto que puede ser el nombre, un código, una marca. (Ricardo Correa F., 2009)

#### Ejemplo 1:

Si la estructura metadatos tiene como base una escuela, cada escuela es un objeto padre que dentro de sí tendría un conjunto de objetos hijos a identificar. Estos últimos a la vez tendrían sus propios objetos hijos, que son una jerarquía de objetos que puede ser tan profunda como se quiera, donde cada objeto tiene asociado una información o concepto.

- I. Escuela (Objeto Padre)
  - a. Aulas (Obj)
    - i. Sillas (obj)
  - b. Laboratorios (Obj)
    - i. Equipos (obj)

### **Paso 5: Definición del esquema de seguridad para lograr integridad, confidencialidad y autenticidad de los datos**

Como se ha citado anteriormente, la guía metodológica está dirigida a la búsqueda de una solución de comunicación e interacción para los sistemas integrales de gestión, pero que utilizan la estrategia de interoperabilidad, basada en servicios web XML.

Algunas de las características de los servicios web, como la accesibilidad a la información, el intercambio dinámico y la autonomía, no están acordes con los modelos y controles tradicionales de la seguridad. Esta situación genera desafíos para modificar los mecanismos de confidencialidad e integridad de los datos que son transmitidos por los protocolos de servicios web. La funcionalidad del principio de la integridad puede variar por la manera como está representada la información, el modelo de disponibilidad cambia sobre la base de los ataques de denegación de servicio, que ahora puede explotar vulnerabilidades de los protocolos registrados para descubrir servicios. Los sistemas de seguridad perimetrales como los cortafuegos, no son adecuados para proteger las arquitecturas basadas en servicios, debido a que estas técnicas son dinámicas. (Isaza E., 2007)

Lo fundamental en este paso es definir adecuadamente el esquema de seguridad, como parte de la arquitectura que se centra en la seguridad de la información que manejará el producto. El objetivo principal de definir cuidadosamente un esquema de seguridad es lograr la integridad, confidencialidad y autenticidad de los datos.

- **Integridad:** Propiedad que busca mantener los datos libres de modificaciones no autorizadas.
- **Confidencialidad:** Propiedad de prevenir la divulgación de información a personas o sistemas no autorizados.
- **Autenticidad:** Propiedad que busca mantener la originalidad y veracidad de los datos.

Para desarrollar el esquema de seguridad que va a controlar y proteger la plataforma de interoperabilidad, se recomienda tener en cuenta varios aspectos esenciales, tales como:

- **Los estándares de seguridad:** Se determinan según el nivel de garantía que se desee dar al producto. Existen innumerables estándares para la seguridad de los sistemas, pero se recomiendan seis clases de ellos, por su reconocimiento internacional. De estos se eligen los que sean convenientes para la seguridad de la plataforma de interoperabilidad. Se pueden encontrar estándares para la administración de la seguridad de la información, para la evaluación de la seguridad en sistemas, para el desarrollo de aplicaciones, para servicios financieros, para riesgo, para autenticación. Entre los estándares de seguridad XML que soportan los servicios web están: el XML Digital Signature (XML DSig), XML Encryption, XML Key Management, Security Authorization Markup Language (TC-SAML), WS-Security, XML Advanced Electronic Signatures (XAdES).

**XML Digital Signature<sup>1</sup> (XML DSig):** Provee mecanismos y especificaciones para crear firmas digitales basadas en XML. Además, define un esquema para capturar el resultado de la operación de firmas digitales y aplicarlas a los metadatos. Este esquema ofrece autenticación, integridad de los datos y soporte para transacciones sin repudio. (Isaza E., 2007)

Una de las principales características del firmado XML es que permite firmar porciones de código XML y no obliga a aplicarse para todo un documento, además permite firmar diferentes tipos de recursos dentro de estos fragmentos de código, como datos XHTML, datos en formatos binarios (JPG , GIF, entre otros) y datos en formatos en XML nativo. La validación de la firma requiere que el objeto que fue firmado sea accesible. La firma XML indica la localización del objeto original firmado, estas localizaciones pueden ser referenciadas por una URI con la firma XML. (Isaza E., 2007)

**XML Encryption<sup>2</sup>:** XML Encryption es un lenguaje cuya función principal es asegurar la confidencialidad de partes de documentos XML a través de la encriptación parcial o total del documento transportado. Se puede aplicar a cualquier recurso web, incluyendo contenido que no es XML. (Perissé, 2009 )

---

<sup>1</sup> XML Digital Signature (Firma Digital XML)

<sup>2</sup> XML Encryption (Lenguaje de Encriptación XML)

**XML Key Management**<sup>3</sup>: Define una especificación que permite a los clientes obtener información de llaves, certificados y administrar los procesos de registro, revocación, entre otros. Es un protocolo para distribuir y registrar claves públicas, oculta los elementos complejos que surgen en estándares como PKI (Infraestructura de Clave Pública). Posee dos componentes fundamentales que son: el registro de la clave pública (X-KRSS) y la información de clave pública (X-KISS). (Isaza E., 2007)

**Security Authorization Markup Language** <sup>4</sup> **(TC-SAML)**: Para intercambiar información de identificación, como infraestructuras de llave pública PKI. Es un entorno basado en XML que facilita el intercambio de autenticación y autorización. Fundamentalmente, el estándar propone un conjunto de procesos para intercambiar los datos de identificación de los usuarios y de los controles de acceso. (Isaza E., 2007)

**WS-Security**<sup>5</sup>: Especifica cómo firmar y cifrar mensajes. Uno de los puntos de partida de este estándar, es que los mecanismos de seguridad basados en HTTP que se usan en muchos casos para intercambiar mensajes, no son suficientes. Es claro que los principios de autenticación, integridad y confidencialidad deben potencializarse en muchos puntos y saltos de la comunicación y no solamente extremo a extremo, tal como lo hacen protocolos como: SSL (Capa de Conexión Segura), TLS (Seguridad de la Capa de Transporte). WS-Security retoma los elementos de seguridad pero usando las especificaciones y estándares tradicionales, es decir, aprovecha las fortalezas de estándares como X.509, Kerberos, XML Encryption y XML Signature, pero WS-Security adiciona mecanismos a estos estándares para que puedan ser incluidos en mensajes. El estándar define elementos para los encabezados y puede utilizar XML Signature para definir la manera como se firmará el mensaje (las claves usadas y la firma). (Isaza E., 2007)

**XML Advanced Electronic Signatures**<sup>6</sup> **(XAdES)**: Provee un sistema básico de autenticación e integridad a través de sistemas de firmas y certificados digitales. Una versión de XadES es el XadES-T que adiciona marcas de tiempo al estándar tradicional, para proveer protección contra el repudio, ya

---

<sup>3</sup> XML Key Management (Gestión de Claves XML)

<sup>4</sup> Security Authorization Markup Language (Lenguaje de Marcas de Afirmación de Seguridad)

<sup>5</sup> WS-Security(Seguridad en Servicios Web)

<sup>6</sup> XML Advanced Electronic Signatures (Firma electrónica avanzada XML)

que permite ejercer controles de periodos de validez de una firma, certificado o par de claves. Otra modificación es la especificación XadES-C, nombrada: XML Advances Electronic Signature con completa validación de datos, que adiciona al XadES-T las referencias a un conjunto de datos soportados para efectos de validación de firma electrónica. El estándar XadES contiene muchas características de la especificación XMLDSIG (XML Digital Signature); sin embargo, adiciona nuevas propiedades. (Isaza E., 2007)

Es importante destacar, que no es obligatorio usar los estándares de interoperabilidad para el desarrollo de la solución. No obstante, se proponen dos de los seis estándares de interoperabilidad citados, por sus características funcionales y por las ventajas que brindan para la seguridad de la información. Estos son: XML Encryption y WS-Security.

- **Los protocolos de comunicación:** Se utilizan para realizar la transmisión de datos entre dos sistemas; es decir, son las reglas y procedimiento utilizados en una red para establecer la comunicación entre los nodos que disponen de acceso a la red. Los protocolos gestionan dos niveles de comunicación distintos. Las reglas de alto nivel definen como se comunican las aplicaciones, mientras que las de bajo nivel definen como se transmiten las señales por cable.

Para la construcción de una solución de interoperabilidad es imprescindible la utilización de los protocolos de comunicación, debido a que la solución que se pretende obtener es una aplicación donde su función fundamental radica en el desarrollo de la comunicación con otros sistemas.

Dentro de los principales protocolos de comunicación que se pueden utilizar para la construcción de la plataforma, debido a su compatibilidad con XML, se encuentran: TCP, IP, HTTP, HTTPS (Protocolo Seguro de Transferencia de Hipertexto), ARP (Protocolo de Resolución de Direcciones), FTP.

**TCP:** Proporciona la base para muchos servicios útiles, incluyendo correo electrónico, transferencia de ficheros y conexión remota.

TCP es uno de los principales protocolos de la capa de transporte del modelo TCP/IP. En el nivel de aplicación posibilita la administración de datos que vienen del nivel más bajo del modelo, o van hacia él (protocolo IP). Cuando se proporcionan los datos al protocolo IP, los agrupa en datagramas IP, fijando el campo del protocolo en seis (para que sepa con anticipación que el protocolo es TCP). TCP

es un protocolo orientado a conexión, es decir, permite que dos máquinas que están comunicadas controlen el estado de la transmisión. Las principales características del protocolo TCP son las siguientes: (Faría, 2009)

- Permite colocar los datagramas nuevamente en orden cuando vienen del protocolo IP.
- Permite que el monitoreo del flujo de los datos y así evita la saturación de la red.
- Permite que los datos se formen en segmentos de longitud variada para entregarlos al protocolo IP.
- Permite multiplexar los datos, es decir, que la información que viene de diferentes fuentes (por ejemplo, las aplicaciones) en la misma línea pueda circular simultáneamente.
- Por último, TCP permite comenzar y finalizar la comunicación.

**IP:** Es un protocolo no orientado a conexión usado tanto por el origen como por el destino para la comunicación de datos a través de una red de paquetes conmutados.

Los datos en una red basada en IP son enviados en bloques conocidos como paquetes o datagramas (en el protocolo IP estos términos se suelen usar indistintamente). En particular, IP no se necesita ninguna configuración antes de que un equipo intente enviar paquetes a otro con el que no se había comunicado antes.

**HTTP:** Su propósito es permitir la transferencia de archivos (principalmente, en formato HTML) entre un navegador (el cliente) y un servidor web, localizado mediante una cadena de caracteres denominada dirección URL.

Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. Al cliente que efectúa la petición (un navegador web o un spider) se lo conoce como agente del usuario. A la información transmitida se la llama recurso y se la identifica mediante un URL (Localizador Uniforme de Recursos). Los recursos pueden ser: archivos, el resultado de la ejecución de un programa, una consulta a una base de datos, la traducción automática de un documento, entre otros.

**HTTPS:** Es un protocolo de red basado en el protocolo HTTP, destinado a la transferencia segura de datos de hipertexto; es la versión segura de HTTP. Se recomienda el uso de este protocolo, no solo por las ventajas funcionales que tiene, sino por el nivel de seguridad que trae incluido.

**ARP:** El protocolo ARP tiene un papel clave entre los protocolos de capa de internet relacionados con el protocolo TCP/IP, ya que permite que se conozca la dirección física de una tarjeta de interfaz de red correspondiente a una dirección IP. Por eso se llama protocolo de resolución de dirección.

**FTP:** Es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP, basado en la arquitectura cliente-servidor. Desde un equipo cliente se puede conectar a un servidor para descargar archivos desde él o para enviarle archivos, independientemente del sistema operativo utilizado en cada equipo.

De los protocolos de comunicación que se han mostrado, se propone la utilización de HTTPS; debido a que es una versión de HTTP, un protocolo de gran funcionalidad en el intercambio y la transferencia de archivos; además HTTPS contiene un nivel extra de seguridad.

- **Lista de chequeo:** Es una de las principales técnicas de control de la seguridad donde se listan las actividades a realizar o requisitos a cumplir en un sistema. Es decir, la lista de chequeo va a permitir que se mantenga un control sobre los aspectos que inciden en la seguridad de la solución, en este caso para la plataforma de interoperabilidad en desarrollo. Por lo planteado anteriormente, se propone la utilización de esta técnica.
- **Las herramientas criptográficas:** Son una forma de mantener la escritura oculta y garantizar la protección o el ocultamiento de la información frente a observadores no autorizados. Con estas herramientas se puede cifrar y descifrar información mediante técnicas especiales. Se emplea frecuentemente para permitir un intercambio de mensajes que sólo puedan ser leídos por personas a las que van dirigidos y que poseen los medios para descifrarlos. A continuación se muestran algunos ejemplos de herramientas para cifrar y descifrar datos:

**Pretty Good Privacy<sup>7</sup> o PGP:** Provee una privacidad bastante aceptada, es un programa cuya finalidad es proteger la información distribuida a través de internet mediante el uso de criptografía de clave pública, así como facilitar la autenticación de documentos con firmas digitales.

PGP combina algunas de las mejores características de la criptografía simétrica y la criptografía asimétrica. Cuando un usuario emplea PGP para cifrar un texto plano, dicho texto es comprimido. La compresión de los datos ahorra espacio en disco, tiempos de transmisión y más importante aún, fortalece la seguridad criptográfica. La mayoría de las técnicas de criptoanálisis explotan patrones presentes en el texto plano para craquear el cifrador. La compresión reduce esos patrones en el texto plano, aumentando enormemente la resistencia al criptoanálisis.

Luego de comprimir el texto, PGP crea una clave de sesión secreta que solo se empleará una vez. Esta clave es un número aleatorio generado a partir de los movimientos del mouse y las teclas que se presionan durante unos segundos, con el propósito específico de generar esta clave. Esta clave de sesión se usa con un algoritmo simétrico convencional para cifrar el texto plano.

Una vez que los datos se encuentran cifrados, la clave de sesión se cifra con la clave pública del receptor (criptografía asimétrica). La clave de sesión cifrada se adjunta al texto cifrado y el conjunto es enviado al receptor. El descifrado sigue el proceso inverso. El receptor usa su clave privada para recuperar la clave de sesión, que PGP luego usa para descifrar los datos.

**GPG o GNU Privacy Guard<sup>8</sup>:** Es una herramienta para cifrado y firmas digitales, que viene a ser un reemplazo del PGP pero con la principal diferencia que es software libre licenciado bajo la GPL. GPG utiliza el estándar del IETF denominado OpenPGP.

GPG cifra los mensajes usando pares de claves individuales asimétricas generadas por los usuarios. Las claves públicas pueden ser compartidas con otros usuarios de muchas maneras, un ejemplo de ello es depositándolas en los servidores de claves. Siempre deben ser compartidas cuidadosamente para prevenir falsas identidades por la corrupción de las claves públicas. También es posible añadir

---

<sup>7</sup> Pretty Good Privacy (Privacidad Bastante Buena)

<sup>8</sup> GNU Privacy Guard (Guardián de Privacidad de GNU)

una firma digital criptográfica a un mensaje, de esta manera, la totalidad del mensaje y el remitente pueden ser verificados en caso de que se desconfíe de una correspondencia en particular.

GPG no usa algoritmos de software que están restringidos por patentes. En su lugar usa una serie de algoritmos no patentados como ElGamal, CAST5, Triple DES (3DES), AES y Blowfish. GPG es un software de cifrado híbrido, que usa una combinación convencional de criptografía de claves simétricas para la rapidez y criptografía de claves públicas, para compartir fácilmente claves seguras; típicamente usando recipientes de claves públicas para cifrar una sesión de clave que es usada una vez. Este modo de operación es parte del estándar OpenPGP y ha sido parte del PGP desde su primera versión.

**John the Ripper:** Es un programa de criptografía que aplica fuerza bruta para descifrar contraseñas. Es capaz de romper varios algoritmos de cifrado. Es una herramienta de seguridad muy popular, ya que permite a los administradores de sistemas comprobar que las contraseñas de los usuarios son suficientemente buenas.

John the Ripper es capaz de autodetectar tipos de cifrado entre muchos disponibles y se puede personalizar su algoritmo de prueba de contraseñas. Esto ha provocado que sea uno de los más usados en este campo.

**Ofuscador de código:** Es una herramienta criptográfica que puede resultar útil a la hora de realizar una solución de interoperabilidad. Permite el ocultamiento de la información a transmitir, es decir, trata de hacerla ilegible al ojo humano para de esta forma otorgarle seguridad. Con la utilización del ofuscador de código se evita que la información pueda ser modificada por un intruso, ocultándola de tal forma que es prácticamente imposible descompilarla, e incluso se dificulta enormemente interpretarla correctamente aún si se obtiene el código fuente.

Todas las herramientas que se muestran, son muy significativas para mantener la seguridad de la información; pero se propone la utilización de un ofuscador de código, como escudo de protección para los datos que van a ser intercambiados a través de la plataforma.

### **Paso 6: Especificación de las reglas de interoperabilidad definidas para cada proceso**

En este paso se definen cada una de las reglas a tener en cuenta para interoperar cada proceso. Estas reglas se dividen en: reglas del negocio y las de seguridad; y tienen como objetivo principal: especificar las restricciones que rigen los datos y controlar la construcción de la plataforma como solución de interoperabilidad.

Cuando se definen estas reglas, se realiza de forma específica por procesos, debido a que no todos tienen las mismas características, ni requisitos. Los especialistas funcionales son los encargados de definir las reglas, que rigen las condiciones bajo las cuales cada proceso va a interoperar.

No obstante, se tienen que cumplir reglas generales que se utilizan obligatoriamente para todos los procesos. Estas manejan y controlan la construcción de la plataforma como solución de interoperabilidad. Las reglas generales que se van adoptar, son las siguientes:

- 1- Definir el orden de los procesos a interoperar en cada sistema de gestión implicado en la construcción de la solución de interoperabilidad. La aplicación de esta regla tiene como objetivo, mantener el desarrollo de la solución en paralelo entre los procesos de los sistemas involucrados, para de esta forma automatizar la construcción de la solución.
- 2- Implementar la solución de interoperabilidad, regida bajo la adopción de una arquitectura orientada a servicios web. Es decir, cada sistema integral de gestión involucrado en el desarrollo de la interoperabilidad debe asumir este tipo de arquitectura.
- 3- Definir una estructura metadato común para cada proceso a interoperar. Cada sistema integral de gestión, debe mantener la utilización de la estructura metadatos anteriormente conciliada en el desarrollo de la interoperabilidad de cada proceso, para evitar contradicciones técnicas durante la construcción de la solución de interoperabilidad. La información tiene que encontrarse estructurada en metadatos; generalmente una estructura XML o en algún otro caso, un derivado de este estándar que sea común para cada proceso que se desee interoperar.
- 4- Definir un estándar o herramienta de seguridad. El producto como solución de interoperabilidad, tiene que contar con un medio que le provea seguridad a la información que se va a intercambiar.
- 5- Utilizar el estándar XML como base tecnológica para el desarrollo de la solución de interoperabilidad. Este estándar asegura la comunicación básica de todos los procesos involucrados en la interoperabilidad. XML va a ser utilizado como base tecnológica y constituye

otro aspecto dentro de las reglas de negocio. Presenta numerosas ventajas y características que lo hace un estándar primordial en temas de interoperabilidad entre sistemas. Es un fichero de texto plano que contiene una serie de datos. Esta información está estructurada siguiendo una gramática adecuada, válida según el estándar de definición XML que entienden los diferentes sistemas gestores de datos.

- 6- Se debe definir además, la utilización de un estándar específico para cada proceso a interoperar; es decir, cada proceso debe estar estandarizado. La selección del estándar, se realiza según las características particulares de la información contenida en cada uno de los procesos en cuestión.
- 7- Precisar el uso de protocolos de comunicación. Dentro de los más importantes se encuentran TCP e IP, que son los que le dan el nombre a esta familia por ser los primeros en surgir, los más usados e importantes. Aunque hay otros no menos significativos como el HTTP, que es el que se utiliza para acceder a las páginas web, el FTP para transferencia de archivos, entre otros.

Como se había mencionado anteriormente, cada uno de los procesos a interoperar cuenta con reglas propias para el desarrollo de la interoperabilidad, aparte de las reglas generales. Para el caso del proceso Comprobante de Operaciones del subsistema Contabilidad, que es el primero en interoperar y además limita el alcance de este trabajo, se van a definir como reglas específicas de este proceso: la utilización de XBRL como estándar de interoperabilidad y la adopción del ofuscador de código como herramienta para regir la seguridad de la información.

Para soluciones futuras como nuevas versiones de la solución de interoperabilidad, se continúan estudiando las reglas específicas que van a controlar el desarrollo de la interoperabilidad en los demás procesos.

- **Reglas:** Describen políticas que deben cumplirse o condiciones que deben satisfacerse, para regular los aspectos esenciales de cada proceso.

### **Paso 7: Implementación**

En este paso se explica cómo se procede con la implementación de la plataforma de interoperabilidad. Es decir, el desarrollo de todos los aspectos que guían la implementación del producto hasta su funcionalidad operativa.

### **Desarrollo general de la solución**

Existen infinitas formas de implementar la plataforma de interoperabilidad, estas varían según el gusto y la comodidad de los desarrolladores. Ahora, se recomienda tener en cuenta varios aspectos esenciales que forman parte de la implementación; para de esta manera, facilitar el trabajo y tener un mayor desenvolvimiento en la búsqueda de la solución. Primeramente, se deben definir y diseñar las clases con sus relaciones y posteriormente se implementan las interfaces. Continúa con la implementación de la lógica del negocio y la construcción del formato general para los XML, en una clase controladora. Estos puntos son claves para la solución; entre otros aspectos que complementan la implementación de la plataforma de interoperabilidad.

### **Desglose de la solución**

Durante la implementación de la plataforma se deben construir varios aspectos esenciales que conllevan a la solución general, por ejemplo: desarrollar un ambiente de configuración. Este va a estar compuesto por varios servicios que se deben consumir; los cuales cargan las interfaces que conforman la estructura del ambiente donde se gestiona la información que va interoperar a través de la plataforma. Otro punto esencial es realizar un gestor de plantilla, el cual se encarga de organizar lógicamente los elementos que forman las taxonomías XML. Además, facilita una estructura diferente para cada proceso a interoperar.

El ambiente de usuario es otro aspecto clave en el desarrollo de la solución; a través de este, el usuario interactúa directamente con la plataforma y ordena la interoperabilidad de cierto proceso o subproceso. Por último, se procede a la publicación de los servicios web que brinda la plataforma de interoperabilidad, que son las funcionalidades de **Exportar** e **Importar** los datos. Esta publicación se realiza en un lugar destinado por directivos del proyecto.

### **Lenguajes y herramientas**

El lenguaje de programación a utilizar para la implementación de la plataforma de interoperabilidad puede ser variado. En el mundo existen muchos de estos, aptos para cumplir con esta tarea; todo depende de la experiencia en cuanto a su uso por parte de los desarrolladores, para de esta manera facilitar el trabajo. Se recomienda que el equipo de desarrolladores se adapte al lenguaje en el cual está implementado el componente al que pertenece la solución en desarrollo; para evitar complicaciones en cuanto a la compatibilidad de los artefactos y las herramientas.

En caso de no existir dependencias con el uso del lenguaje, se propone la utilización de PHP debido a sus características y por poseer una clase que se encarga de gestionar XML. El framework que se utilice, tiene que ser compatible con el lenguaje de programación adoptado por el equipo de desarrollo, en caso de ser PHP se recomienda Zend framework.

### **2.3 Conclusiones parciales**

En este capítulo se elabora la guía metodológica, la cual constituye una referencia práctica para el desarrollo de soluciones de interoperabilidad en sistemas integrales de gestión en Cuba, que tengan como objetivo desarrollar la comunicación con otros sistemas de gestión.

La guía metodológica cuenta con una serie de pasos lógicos que deben cumplirse para interoperar sistemas; que van desde la selección de los procesos a intercambiar, hasta las definiciones arquitectónicas que rigen la implementación de la solución. Además, se tienen en cuenta otros aspectos como: los estándares de interoperabilidad, las reglas, las políticas de seguridad, entre otros.

### **CAPÍTULO 3: CASO DE ESTUDIO APLICADO AL SISTEMA INTEGRAL DE GESTIÓN CEDRUX**

#### **3.1 Introducción**

En el presente capítulo se realiza la validación de la guía metodológica aplicada al Sistema Integral de Gestión CedruX; para de esta forma brindar una idea más práctica sobre los aspectos que inciden en la construcción de la plataforma de interoperabilidad. Cada paso dentro del caso de estudio, pertenece a la guía metodológica y especifica los aspectos a tener en cuenta para la construcción de la solución que brindará interoperabilidad al proceso Comprobante de Operaciones.

A continuación, se muestran los pasos que conforman la guía metodológica y que fueron aplicados en el Sistema Integral de Gestión CedruX.

#### **3.2 Aplicación de la guía metodológica al proceso Comprobante de Operaciones del subsistema Contabilidad**

##### **Paso 1: Identificar los procesos a interoperar, así como el orden de prioridad de los mismos**

Seguidamente se muestra el orden de los subsistemas, donde cada uno de estos incluye los procesos o subprocesos a interoperar. Se resalta el proceso Comprobante de Operaciones del subsistema Contabilidad por ser el primero y limitar el alcance en la aplicación de la guía metodológica al Sistema Integral de Gestión CedruX.

- Procesos a interoperar:
  - I. **Contabilidad**
    - a. **Comprobante de operaciones**
  - II. Capital Humano
  - III. Cobros y Pagos
  - IV. Caja
  - V. Banco
  - VI. Costos y procesos
  - VII. Inventario

- VIII. Facturación
- IX. Planificación
- X. Activos fijos (tangibles e intangibles)
- XI. Estructura y composición
- XII. Nomencladores generales
- XIII. Seguridad

- Prioridad

El orden de los procesos se establece de acuerdo con la necesidad que tenga el sistema Cedrux en interoperarlos. Siempre son los especialistas funcionales los encargados de establecer la prioridad, por ser los que conocen las características del negocio. En este caso, los especialistas del Cedrux definieron el proceso Comprobante de Operaciones como el primer objetivo a interoperar.

### **Paso 2: Arquitectura**

Como se ha explicado en el capítulo anterior, la construcción de la plataforma de interoperabilidad se apoya en una arquitectura orientada a servicios web. Esta se enmarca en la creación de servicios web a partir de tareas contenidas en aplicaciones nuevas o ya existentes.

Un aspecto importante en este paso, es definir los servicios pertenecientes al Sistema Integral de Gestión Cedrux, que forman parte de la arquitectura definida en la construcción de la plataforma de interoperabilidad. Seguidamente se explica el uso de cada uno de los servicios utilizados para el desarrollo de la solución:

### **Revisión de los servicios que se necesitan para interoperar**

En la plataforma va a estar implementada la interoperabilidad hasta el proceso Comprobante de Operaciones del subsistema Contabilidad. Para la construcción de la misma se utilizan cinco servicios: dos creados por el equipo de desarrollo de Seguridad, dos por Estructura y Composición y uno creado por el equipo de Contabilidad del proyecto. La creación de estos servicios fue encargada por la dirección del equipo de interoperabilidad, teniendo en cuenta el aporte de cada uno de estos en el desarrollo de la plataforma.

El primer servicio que se utiliza brindado por Seguridad fue nombrado **getSystems**, tiene la función de cargar todos los subsistemas del Sistema Integral de Gestión Cedrux en la interfaz, con los componentes y subcomponentes (en caso de tenerlos) asociados a cada uno de estos subsistemas. Uno de estos subsistemas es Contabilidad, con el componente Comprobante de Operaciones incluido; a partir de este se va crear un proceso que será el objetivo a interoperar, nombrado por el equipo de interoperabilidad como Comprobante de Operaciones, por pertenecer a dicho componente.

Otro servicio que se consume perteneciente a Seguridad es **returnConex** y será el encargado de cargar en la interfaz los aspectos de conexión que influyen en el desarrollo de la interoperabilidad del proceso en cuestión. Estos aspectos son: el servidor de conexiones, el servidor de base de datos, el gestor de base de datos, el esquema y las tablas con los atributos. Las tablas son: **Asiento, Pase, Cuenta y Anexo**.

El servicio **obtenerEjercicios** es implementado por Estructura y Composición. Este se utiliza en la construcción de la plataforma de interoperabilidad con el objetivo de facilitar los años a los que pertenecen todos los Comprobantes de Operaciones. El equipo de desarrollo de Estructura y Composición brinda además otro servicio: **obtenerPeriodos**, encargado de suministrar los meses donde se encuentran todos los Comprobantes de Operaciones. Este servicio en conjunto con **obtenerEjercicios**, tiene la tarea de ubicar todos los Comprobantes de Operaciones en la interfaz correspondiente, donde además se selecciona su estado, que puede ser Asentados, Terminados o Ambos Estados.

Por último, Contabilidad implementa el servicio **interoperarComprobanteExportarGeneral**, el cual se consume con el objetivo de asignar valores a los atributos seleccionados que conforman las tablas de los Comprobantes de Operaciones que pueden interoperar. Una vez asignados los valores a dichos atributos, el usuario selecciona el Comprobante de Operaciones que desea interoperar y finaliza la generación del XML correspondiente a este proceso.

El equipo de desarrolladores del componente de interoperabilidad consume los servicios mencionados anteriormente, los cuales se encuentran publicados en el Componente de Inversión de Control (IOC), del Sistema Integral de Gestión Cedrux. El IOC es una cartera de servicios, donde se publican dichos servicios al alcance de los desarrolladores del Cedrux, para implementar las funcionalidades de los demás componentes.

### **Paso 3: Estándares**

En el capítulo 2, se explica el por qué de la utilización de XML como estándar base o base tecnológica. Sus características lo ubican como el estándar perfecto para la implementación de la plataforma de interoperabilidad. Solamente con XML no se puede lograr una completa comunicación entre sistemas integrales de gestión, a la hora de interoperar el proceso Comprobante de Operaciones; debido a que el estándar XML solo garantiza la comunicación básica entre las aplicaciones. Por lo cual, es necesario la adopción de un estándar que se especialice en el desarrollo de la interoperabilidad para la información financiera que está contenida en el proceso a interoperar.

Dentro de las opciones que brinda la guía metodológica presentada en el capítulo anterior, se propone un estándar perfecto para el desarrollo de la interoperabilidad del proceso en cuestión; por sus características funcionales y su compatibilidad con XML. Este estándar es XBRL, el más indicado para soluciones de interoperabilidad basadas en el intercambio de información financiera.

### **XBRL, estándar para el proceso Comprobante de Operaciones del subsistema Contabilidad**

Como se expuso anteriormente, XBRL es el estándar escogido para interoperar Comprobantes de Operaciones, por lo que es necesario un estudio detallado de las características que presenta y de las ventajas que brinda para el desarrollo de la interoperabilidad.

XBRL está definido para facilitar la transmisión y análisis de información financiera y de negocio. Además, permite la automatización avanzada de la cadena de suministro de información, lo que facilita un acceso al contenido de los diversos informes de gestión. Ayuda al uso dinámico de la información y supone un avance hacia una mayor transparencia de los mercados. (Rodolfo Uribe, 2006)

El XBRL es un estándar abierto (sin costo por licencia o uso), basada en el lenguaje XML. Esta característica permite a las empresas reducir los costos de emisión de estados financieros, puesto que un mismo documento XBRL se puede convertir a diversos formatos de presentación (web, PDF, HML, RTF, entre otros) sin intervención humana, lo cual minimiza la existencia de errores en la información entregada, así como la rapidez de los procesos. (Caballero, 2004)

El uso de XBRL trae varios beneficios relacionados con el aumento de valor agregado y con la reducción de costos: (Laboreo, 2005)

### **Beneficios de aumento de valor agregado**

- Permite generar cualquier tipo de información financiera o de gestión.
- Aumenta la velocidad y precisión en la generación y preparación de informes.
- Proporciona conectividad entre los actores del mercado: compañías, entidades reguladoras, analistas, inversionistas, bancos.
- Es extensible porque facilita la adaptación a las necesidades futuras, incluso aquellas que no se han detectado.

### **Beneficios relacionados con la reducción de costos**

- Al independizar los sistemas con el proceso de intercambio de información, reduce los costos de este proceso al soportarse en estándares.
- Se facilita la adaptación ante cambios normativos.
- La aplicación de estándares internacionales y el uso de tecnologías específicas para la generación de información, proporcionan mayor eficiencia en el proceso de reporte a las empresas.
- La aplicación de estándares proporciona agilidad y mayor eficiencia a los consumidores.

En resumen, XBRL es un estándar cuya necesidad y surgimiento se debe a la evolución coherente de la sociedad de la información en la esfera de los reportes financieros y su desarrollo; lejos de opacarse, irá en aumento a medida que la tecnología informática se utilice de manera más amplia para la toma de decisiones en la infraestructura de las relaciones económicas regionales e internacionales. La construcción de la plataforma de interoperabilidad está sin dudas beneficiada con la adopción de XBRL, como el estándar para el desarrollo de la solución.

### **Paso 4: Estructura metadatos**

La estructura metadatos del proceso Comprobante de Operaciones, no es más que la taxonomía que rige y organiza la información que será intercambiada. A continuación se muestra la estructura metadatos general y con la inclusión de XML para este proceso.

- **Estructura metadatos para Comprobante de Operaciones**

La estructura que se presenta a continuación es el desglose que va a tener normalmente el comprobante de operaciones, sin los conceptos.

```
+COMPROBANTEOPERACIONES
  +ASIENTO
    +PASE
      +CUENTA
        + ANEXO
```

- **Estructura metadatos para Comprobante de Operaciones con XML**

La estructura que se presenta a continuación es el desglose que va a tener el comprobante de operaciones, con los conceptos y además en un formato XML para un mejor entendimiento.

```
<COMPROBANTEOPERACIONES numerocomprobante = "" entidad = "" cierreapertura = ""
  descripcion = "" fechaemision = "" subsistema = "" sumaclavecomp = "" sumaimportecomp = ""
  usuario = "" estadocomp = "" tipocomprobante = "" periodo = "" version = "" estructurae = "">
  <ASIENTO clave = "" fecha = "" descripcion = "" codigodocumentoprimary = "" version = ""
    comprobante = "">
    <PASE>
      <CUENTA clave = "" importemonedabase = "" importemonedaoriginal = "" anexo = ""
        cuenta = "" asiento = "" moneda = "" tasa = "" version = "">
        <ANEXO clave = "" importemonedabase = "" importemonedaoriginal = "" version = ""
          pase = "" elementocomun = ""/>
      </CUENTA>
    </PASE>
  </ASIENTO>
</COMPROBANTEOPERACIONES>
```

### **Paso 5: Definición del esquema de seguridad para lograr integridad, confidencialidad y autenticidad de los datos**

- Uno de los aspectos esenciales que componen el esquema de seguridad de la plataforma como solución de interoperabilidad, es la adopción de algún protocolo de comunicación como elemento fundamental para el desarrollo de la interoperabilidad entre sistemas integrales de gestión. De los protocolos de comunicación que propone la guía metodológica, se ha seleccionado HTTPS, debido a que es un protocolo de red destinado a la transferencia segura de datos de hipertexto y basado en el popular HTTP.

HTTPS es utilizado en el mundo entero principalmente por entidades bancarias, tiendas en línea y cualquier tipo de servicio que requiera del envío de datos financieros y seguros; de ahí a que sea el protocolo perfecto a utilizar, para el desarrollo de la interoperabilidad del proceso seleccionado. El principal objetivo de HTTPS, es crear un canal seguro sobre una red insegura. Esto proporciona una protección razonable contra ataques, siempre que se empleen métodos de cifrado adecuados y que el certificado del servidor sea verificado y resulte de confianza.

- Otro aspecto fundamental, es la utilización de alguna herramienta que brinde seguridad a los datos. La información contenida dentro del proceso Comprobante de Operaciones que se manipulará a través de la plataforma de interoperabilidad, estará protegida por un ofuscador de código. El ofuscador es una herramienta criptográfica realizada por el equipo de desarrollo que permite el ocultamiento de la información a transmitir; es decir, hacerla ilegible al ojo humano para de esta forma otorgar seguridad a los datos. Con la utilización del mismo se evita que la información pueda ser modificada por un intruso, ocultándola de tal manera que es prácticamente imposible descompilarla e incluso se dificulta enormemente interpretarla correctamente, aún si se obtiene el código fuente.

La ofuscación se consigue en la práctica por medio de la inclusión de bucles irrelevantes, cálculos innecesarios, comprobaciones absurdas, nombres de funciones y de variables que no tienen nada que ver con su cometido, funciones larguísimas, interacciones inverosímiles entre variables y funciones. De esta forma, se crean anomalías en los datos que se desean proteger que provocan complicaciones a la hora de desenmascarar la información. (Marañón, 1998)

- Por último; se ha utilizado una lista de chequeo como técnica para el control de la seguridad de la plataforma de interoperabilidad; donde se listan las actividades a realizar o los requisitos que se deben cumplir durante la construcción de la solución.

### **Paso 6: Especificación de las reglas de interoperabilidad definidas para cada proceso**

A la hora de interoperar un proceso, es necesario que esté regido por reglas, las cuales se van a denominar reglas de interoperabilidad. Para el proceso Comprobante de Operaciones, se han definido dos reglas específicas, aparte de las reglas generales explicadas en el capítulo anterior las cuales deben adoptarse obligatoriamente. Las reglas específicas definidas para este proceso, son las siguientes:

- 1- Definición de XBRL como estándar de interoperabilidad, por ser el más usado para el intercambio de información de carácter financiero y tener como base tecnológica el estándar XML.
- 2- Utilización del ofuscador de código. Regla dirigida a controlar la seguridad en la transmisión y recepción de la información referente a este proceso. Los ofuscadores se utilizan para transformar el código interpretado en un código que funcione igual pero que sea ilegible.

### **Paso 7: Implementación**

#### **Desarrollo general de la solución**

Para construir la plataforma de interoperabilidad en el caso de estudio propuesto, se han realizado una serie de pasos que componen la implementación de la misma. Primeramente, se han definido y diseñado las clases con sus relaciones y posteriormente se han implementado las interfaces. Continúa con la implementación de la lógica del negocio, donde se cargan los subsistemas y componentes. A partir del componente Comprobante de Operaciones se crea el proceso Comprobante de Operaciones y los demás elementos relacionados. Se construye además el formato de los XML en una clase controladora y se realizan otros aspectos que complementan la implementación de la plataforma.

#### **Desglose de la solución**

- **Ambiente de configuración**

Durante la implementación de la plataforma de interoperabilidad, un aspecto fundamental es desarrollar un buen ambiente de configuración. Para la realización de este, se han consumido dos servicios, ambos elaborados por el equipo de desarrolladores de Seguridad.

El primer servicio nombrado **getSystems**, muestra una interfaz donde aparecen todos los subsistemas del Sistema Integral de Gestión Cedrux, con los componentes o subcomponentes asociados (estos últimos en caso de poseerlos). A continuación, aparece la figura correspondiente:

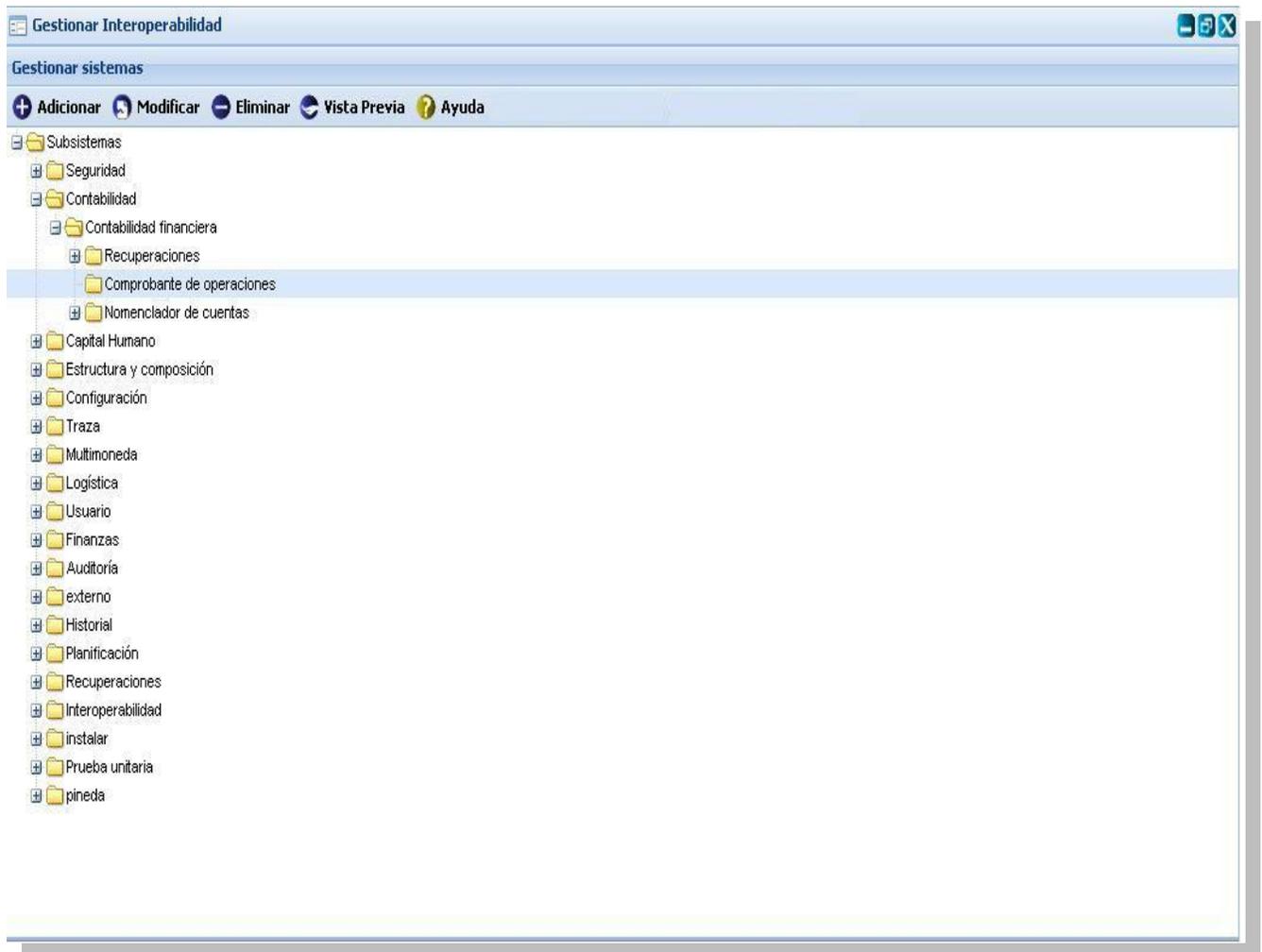


Figura 1. Gestionar Interoperabilidad

De estos componentes que han sido cargados, se **Adiciona** a partir del Comprobante de Operaciones el proceso que posteriormente va a interoperar. Para adicionar un proceso aparece la interfaz que se muestra en la figura 2:

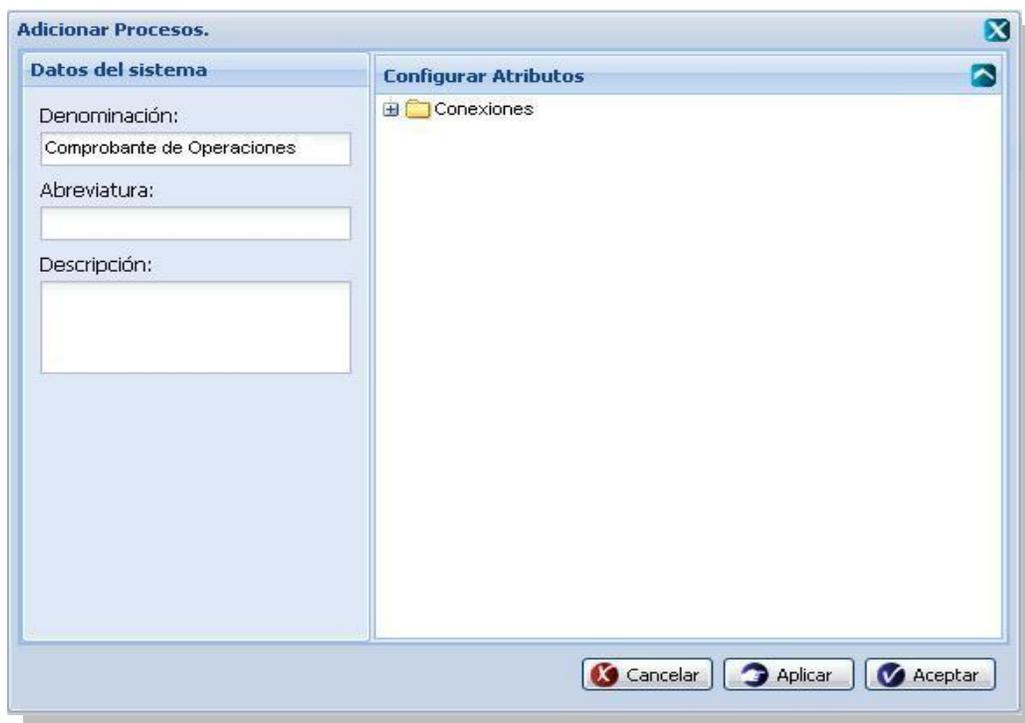


Figura 2. Adicionar Procesos

Dentro de los campos que se encuentran en la interfaz Adicionar Procesos, aparece Denominación. La denominación de un proceso es el nombre que se le designa, generalmente será el mismo de su componente correspondiente, para el caso en cuestión se adopta el nombre: Comprobante de Operaciones. De igual forma se puede **Eliminar** o **Modificar** un proceso existente, tan solo se selecciona y se ejecuta la opción deseada.

Para modificar un proceso que se desee interoperar, inmediatamente de ejecutarse esta opción se carga la interfaz Modificar Procesos, conformada por los mismos campos de su homóloga Adicionar Procesos, como se muestra en la figura 3. Después de ser llenados los campos, se ejecuta **Aceptar** para de esta manera modificar el proceso antes indicado. Si se desea eliminar un proceso, se

selecciona y posteriormente se da la confirmación de eliminar; en la figura 4 se puede observar la ventana que corresponde a la ejecución de dicha operación.

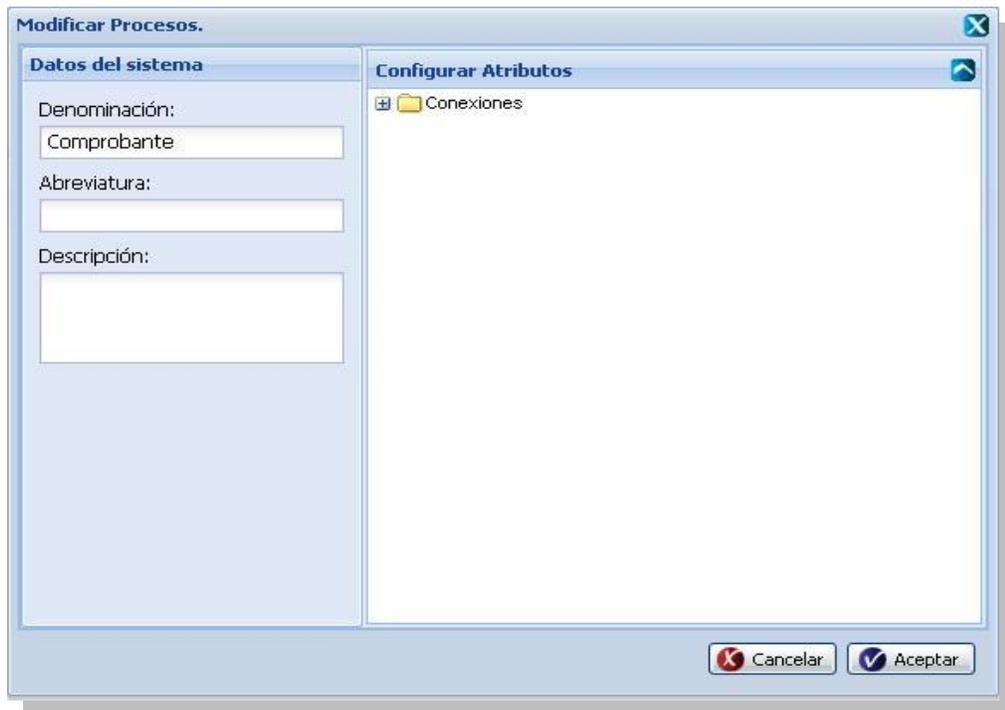


Figura 3. Modificar Procesos

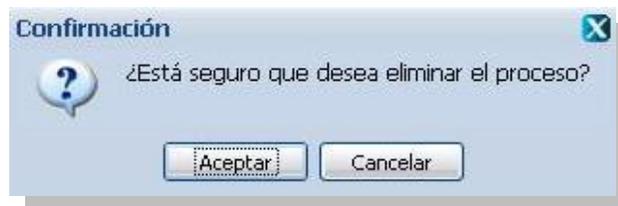


Figura 4. Confirmación

En el instante que se está creando el proceso Comprobante de Operaciones (objetivo del presente trabajo), se deben configurar los atributos que componen el XML para el proceso en cuestión. Para realizar esta operación se cargan en la interfaz, como se aprecia en la figura 5, todos aquellos aspectos de conexión que inciden en el desarrollo de la interoperabilidad del proceso. Estos son: servidor de conexión, servidor de base de datos, gestor de base de datos, base de datos, el esquema y las tablas. Para acceder a la información que existe en la base de datos se requiere de una previa

autenticación. En la siguiente figura se muestra la ventana donde se autentica el responsable de acceder a la base de datos; y por consiguiente al esquema y las tablas, para de esta forma continuar gestionando el proceso de interoperabilidad.

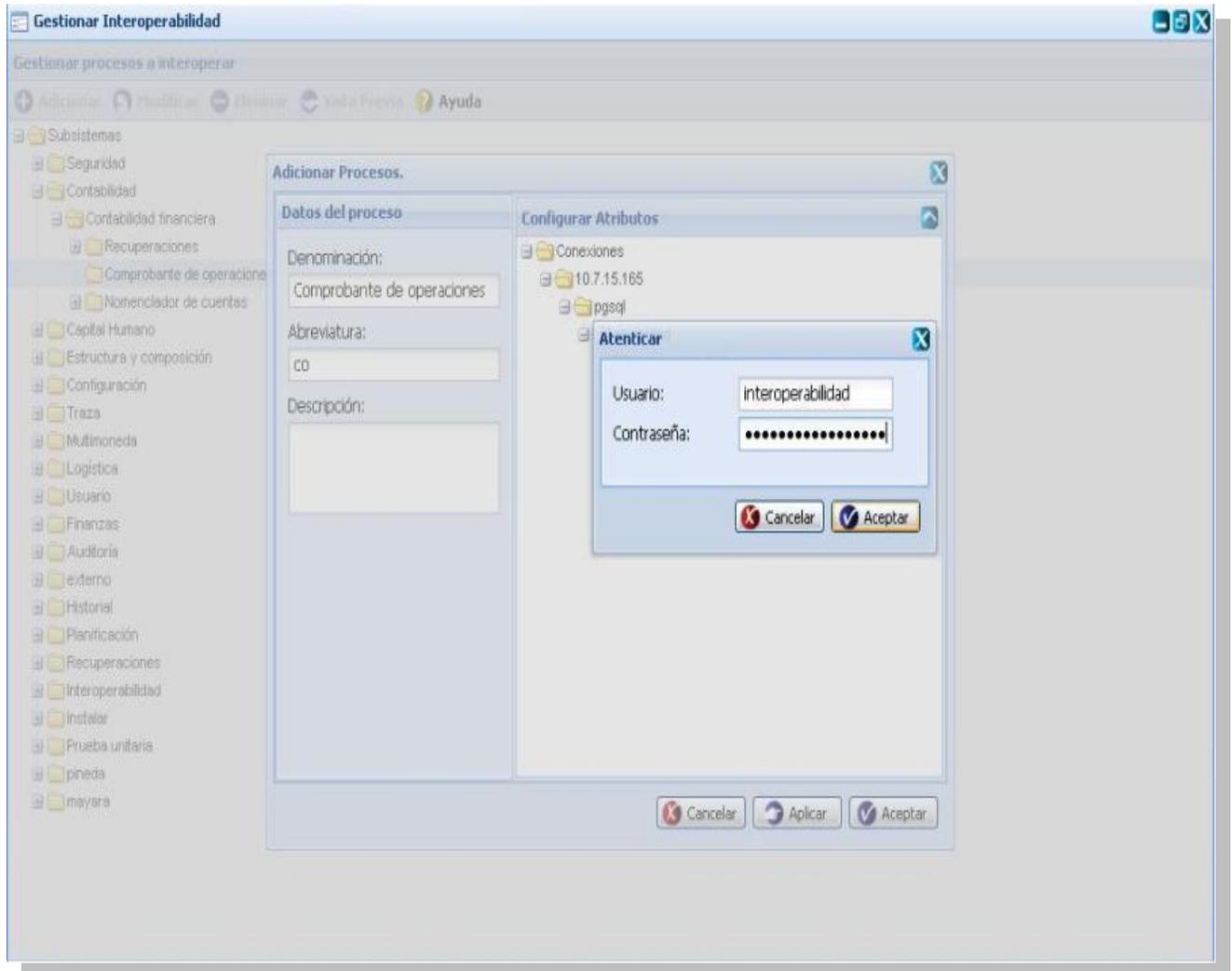


Figura 5. Gestionar Interoperabilidad (autenticación base de datos)

Las tablas son cargadas luego de consumirse el segundo servicio brindado por Seguridad nombrado **returnConex**. Estas tablas son: **Asiento**, **Pase**, **Cuenta** y **Anexo**; y se estructuran de la siguiente forma: cada **Asiento** cuenta con Pases, cada **Pase** tiene Cuentas y si la **Cuenta** es de gasto se genera un registro **Anexo**. Posteriormente, al seleccionarse los atributos de las tablas que conforman

el futuro XML a interoperar, se elige la opción **Aplicar y Aceptar**, y finalmente queda adicionado el proceso Comprobante de Operaciones. En la figura 6 aparece la ventana de confirmación correspondiente a dicha operación.

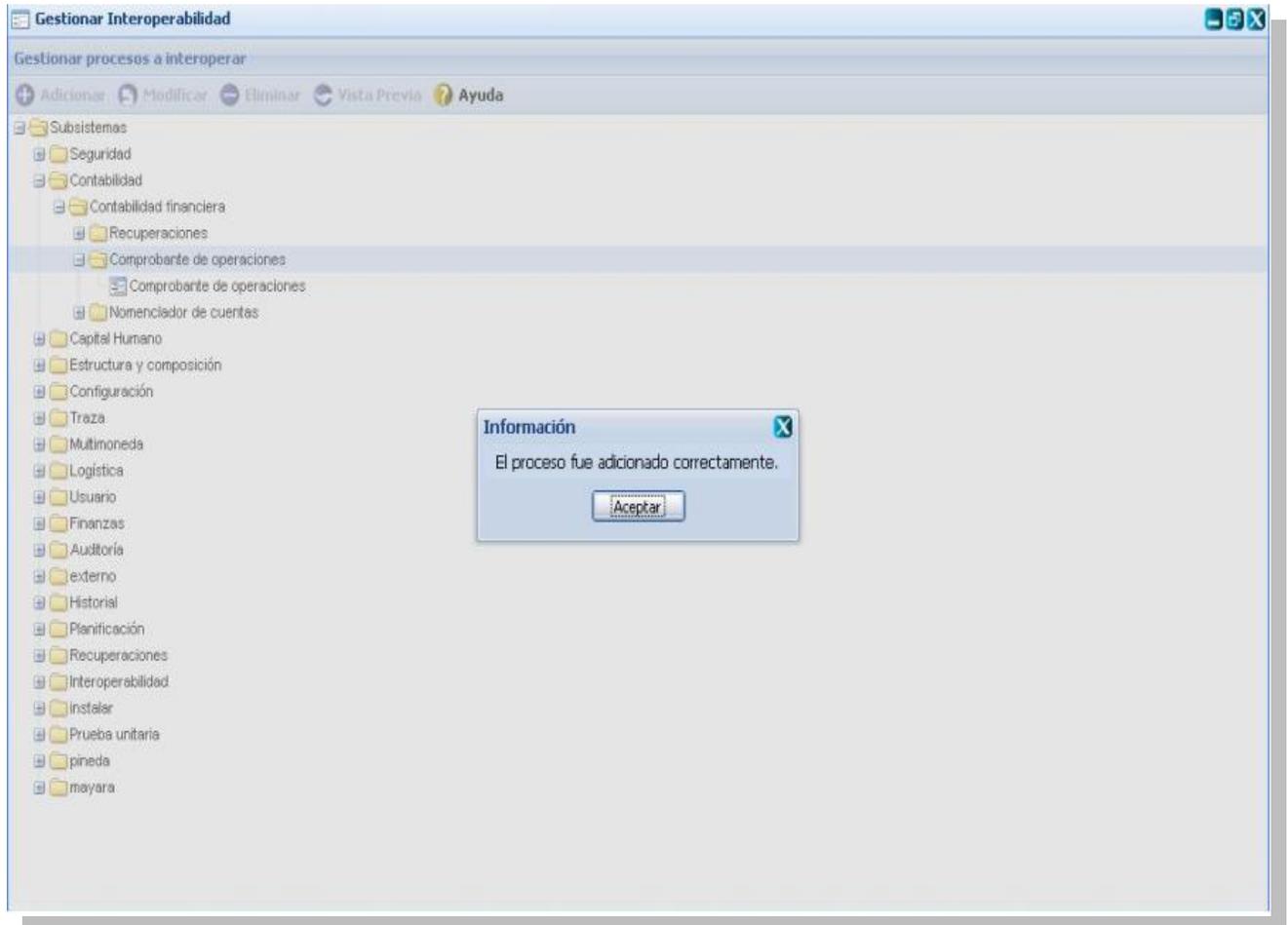


Figura 6. Gestionar Interoperabilidad (confirmación de adición)

Todos los pasos explicados anteriormente, detallan las acciones que realiza un administrador del Sistema Integral de Gestión Cedrux, con los permisos para ejecutarlas. Este además puede observar el XML del proceso creado, para valorar como queda finalmente su estructura. Solo tiene que ejecutar el botón **Vista Previa** y se carga en una nueva ventana con la información de este.

- **Ambiente de usuario**

El ambiente de usuario va a estar conformado por el consumo de tres servicios; dos de estos los brinda Estructura y Composición, y el otro Contabilidad. La interfaz de usuario es sencilla y sin complicación para su uso.

Después que el administrador del sistema crea el proceso Comprobante de Operaciones y selecciona los elementos que conforman el XML que va a interoperar, el usuario se encarga de importar o exportar dichos datos. Para realizar una de estas opciones es necesario que el usuario enmarque y seleccione el Comprobante de Operaciones que desea intercambiar.

En la interfaz que se muestra en la figura 7, se recogen los datos que el usuario selecciona para enmarcar el o los Comprobantes de Operaciones que desee interoperar. En esta se ilustran tres comboBox; el primero se nombra **Ejercicios** y sus datos van a ser cargados después de consumirse el tercer servicio **obtenerEjercicios** brindado por el equipo de desarrollo de Estructura y Composición. En este comboBox se selecciona el año o los años dentro de los que se encuentran los Comprobantes de Operaciones. El segundo comboBox se nombra **Período** y en este se van a elegir los Comprobantes de Operaciones que se encuentran dentro del mes o los meses señalados. Los datos de este comboBox, van a ser cargados después de consumirse el cuarto servicio **obtenerPeriodo**, brindado también por el equipo de Estructura y Composición. El otro comboBox es **Estado**, donde se elige uno de los tres estados que pueden presentar los Comprobantes de Operaciones: **Asentados**, **Terminados** o **Ambos Estados**.

Después de seleccionarse en los comboBoxes, todos los Comprobantes de Operaciones según su **Estado**, automáticamente se cargan en la interfaz todos aquellos que cumplen con esos parámetros. Es entonces cuando se consume el quinto y último servicio nombrado **interoperarComprobanteExportarGeneral**, el cual ha sido brindado por el equipo de desarrollo de Contabilidad. Su función está en suministrarle valor a los atributos que conforman las tablas de los Comprobantes de Operaciones que se van a mostrar en la interfaz. De estos Comprobantes de Operaciones, se selecciona el que el usuario desee interoperar y se ejecuta **Aceptar**.

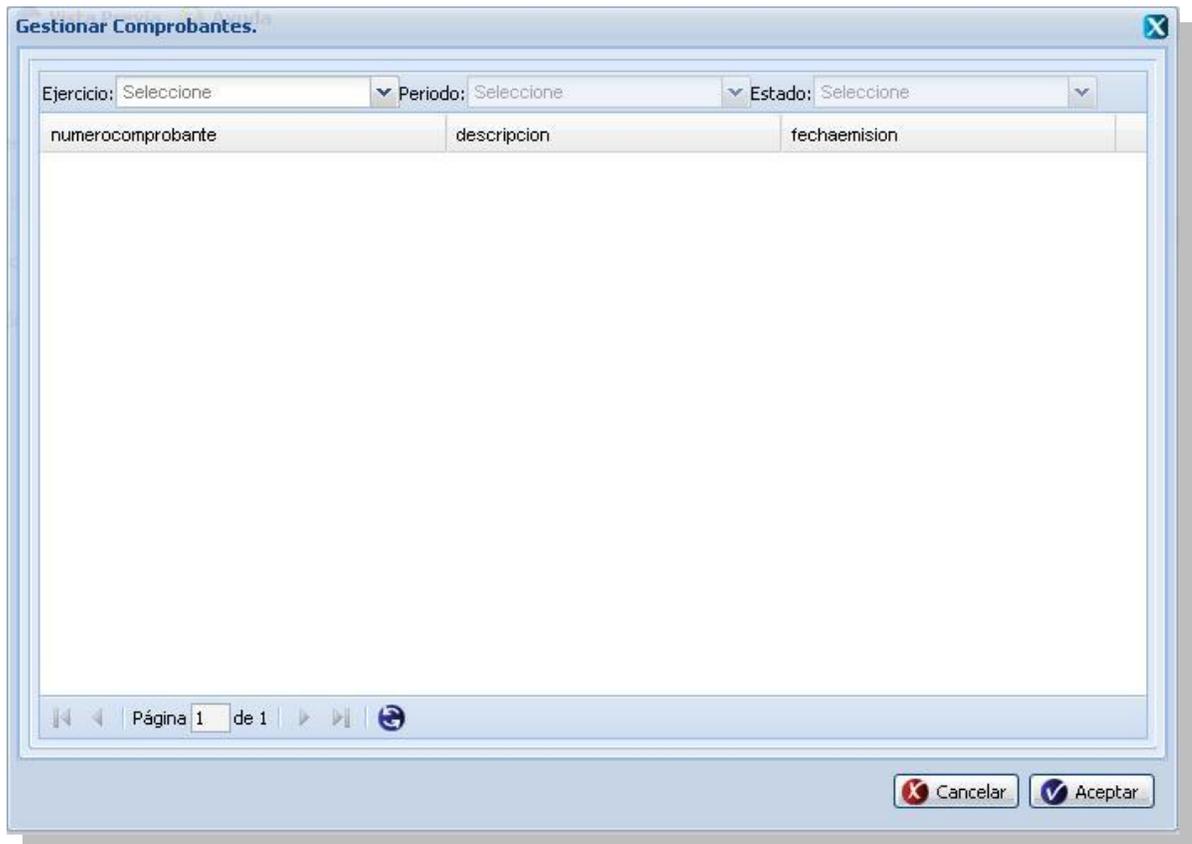


Figura 7. Gestionar Comprobantes

Por último, se carga en la interfaz principal de usuario mostrada en la figura 8 el Comprobante de Operaciones escogido anteriormente. El usuario selecciona el proceso en cuestión y ejecuta la opción **Exportar** o **Importar**.

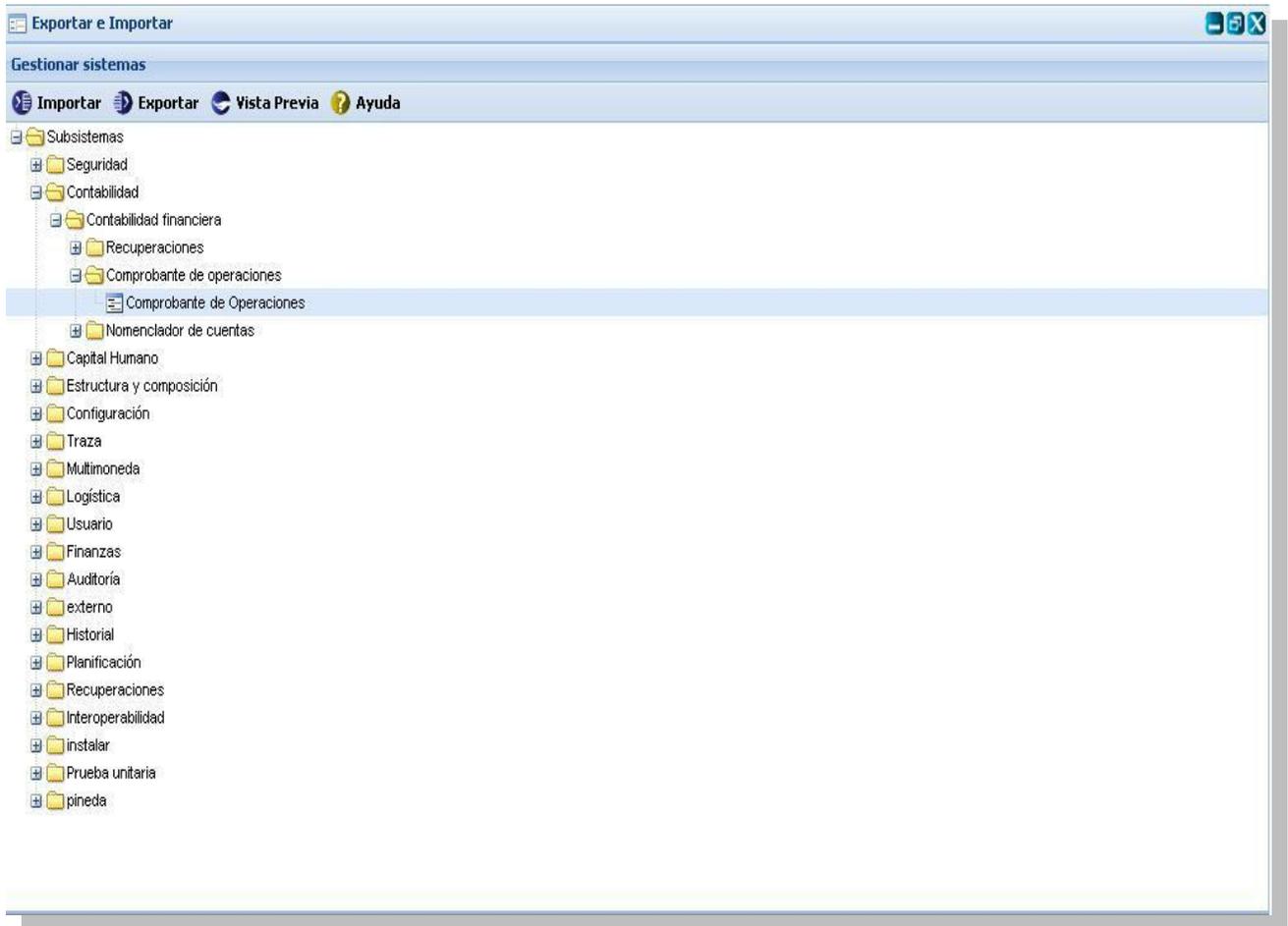


Figura 8. Exportar e Importar

- **Gestor de plantilla**

Otro aspecto fundamental que incide en la implementación de la plataforma, es el gestor de plantilla, que como su nombre indica es el encargado de gestionar o estructurar la plantilla XML. Es decir, internamente anida los elementos que componen el XML a interoperar. A continuación se muestra de forma general como se organizan las tablas del proceso Comprobante de Operaciones.

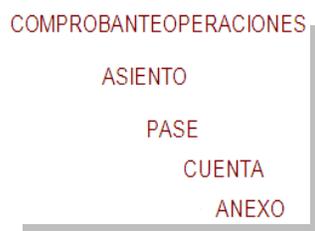


Figura 9. Elementos del proceso Comprobante de Operaciones

- **Publicar servicios**

Finalmente, después que se concluye con la implementación de la plataforma de interoperabilidad, se procede a publicar los servicios que brinda en la cantera de servicios del Sistema Integral de Gestión Cedrux. Estos son: **Importar datos** y **Exportar datos**.

### Herramientas

- **PHP**

En el sistema Cedrux, el lenguaje de programación web que se utiliza es PHP y de esta forma está definido en el marco de trabajo, por lo que ha sido adoptado para la implementación de la plataforma. Además, PHP brinda muchas ventajas y posee una clase llamada simple XML encargada de gestionar XML, muy útil para el desarrollo de la solución de interoperabilidad. Este es un lenguaje rico en características, que lo hace uno de los más usados y potentes en temas de programación web.

- **Zend Framework**

Para la construcción de la plataforma de interoperabilidad, se utilizó Zend Framework. Fue seleccionado por sus ventajas y experiencias de uso en el equipo de desarrollo. Se trata de un framework para el desarrollo de aplicaciones y servicios con PHP, que brinda soluciones para construir sitios web modernos, robustos y seguros. Es código abierto y trabaja con PHP5.

### 3.3 Conclusiones parciales

En el presente capítulo se evalúa la construcción de la guía metodológica de interoperabilidad mediante su aplicación al proceso Comprobante de Operaciones del subsistema Contabilidad en el Sistema Integral de Gestión Cedrux.

En el capítulo se especifica el trabajo realizado en la guía metodológica abordada en el capítulo anterior. Cada paso en el caso de estudio está enfocado de forma específica al desarrollo de la interoperabilidad del proceso en cuestión, luego de haberse seleccionado los aspectos esenciales para la construcción de su solución, a partir de las propuestas expuestas en los pasos generales de la guía metodológica.

Las principales herramientas, lenguajes y tecnologías utilizadas para la construcción de la plataforma de interoperabilidad; van desde la adopción de PHP como lenguaje de programación web, XBRL como estándar de interoperabilidad, HTTPS como protocolo de comunicación, el ofuscador de código como herramienta de seguridad, hasta la utilización de Zend Framework.

### CONCLUSIONES GENERALES

Con el desarrollo del presente trabajo de diploma se cumplió el objetivo principal de la investigación; crear una guía metodológica de interoperabilidad, que permita estandarizar el intercambio de información entre sistemas integrales de gestión en Cuba. Para obtener dicho resultado, se cumplieron los siguientes aspectos:

- Se realizó un estudio teórico donde se alcanzaron los conocimientos suficientes para el desarrollo de la guía metodológica, profundizándose en aspectos significativos relacionados con el desarrollo de la interoperabilidad entre sistemas integrales de gestión.
- Se identificó satisfactoriamente el orden de los procesos a interoperar dentro del Sistema Integral de Gestión Cedrux. El proceso Comprobante de Operaciones fue seleccionado como el primer objetivo a interoperar, a partir, de la prioridad definida por el especialista funcional encargado.
- Después de realizarse un análisis de las principales arquitecturas existentes, se adoptó la arquitectura orientada a servicios web como la más idónea para la construcción de la plataforma de interoperabilidad.
- Se estableció como base tecnológica para el desarrollo de la interoperabilidad, el estándar XML. XBRL fue seleccionado como el estándar de interoperabilidad para el proceso Comprobante de Operaciones, a partir, de una serie de estándares propuestos con características diferentes, especializados en el intercambio y transporte de datos.
- Se definieron los aspectos fundamentales que inciden en la construcción de la estructura metadatos para cada proceso a interoperar. Para el proceso Comprobante de Operaciones, se utilizó un gestor de plantilla como herramienta para organizar la información contenida en la estructura XML; obteniéndose resultados satisfactorios.
- Se propusieron los elementos fundamentales con los que debe contar el esquema de seguridad. Para proteger la información del proceso Comprobante de Operaciones, se definió el uso de HTTPS como protocolo de comunicación, una lista de chequeo como técnica de control, así como un ofuscador de código como herramienta para mantener la seguridad de la información.
- Se establecieron las reglas generales de interoperabilidad que deben ser adoptadas obligatoriamente para cada proceso. Además, se definió usar el ofuscador de código y el

estándar XBRL como reglas básicas para el desarrollo de la interoperabilidad del proceso Comprobante de Operaciones.

- Se definieron los aspectos esenciales para la implementación de la solución, desarrollándose un ambiente de configuración, un ambiente de usuario y un gestor de plantilla. A través, de estos aspectos se logró organizar y llevar a cabo con éxito la implementación de la plataforma de interoperabilidad.

### **RECOMENDACIONES**

Luego de haberse cumplido con los objetivos propuestos mediante la realización del trabajo de diploma, se recomienda:

- Aplicar la guía metodológica durante el desarrollo de la interoperabilidad del resto de los procesos que conforman el Sistema Integral de Gestión Cedrux.
- Promover la guía metodológica a todas las instituciones que desarrollan software de gestión, con el fin de estandarizar a nivel de país el proceso de interoperabilidad.

**BIBLIOGRAFÍA****Referenciadas**

- Acero, Fernando. 2009.** [En línea] 6 de 2009. [Citado el: 6 de 4 de 2010.] <http://www.kriptopolis.org/interoperabilidad-4>
- Álvarez, Miguel Angel. 2001.** [En línea] 13 de 6 de 2001. [Citado el: 8 de 4 de 2010.] <http://www.desarrolloweb.com/articulos/449.php>
- Álvarez, Miguel Angel. 2003.** *maestrosdelweb. maestrosdelweb.* [En línea] 3 de noviembre de 2003. [Citado el: 16 de febrero de 2010.] <http://www.maestrosdelweb.com/editorial/zendstudio/>
- Billy, Carlos. 2004.** *Introducción a la Arquitectura de Software.* 2004.
- Borges, Alejandro E. 2007.** *coit. coit.* [En línea] febrero-marzo de 2007. [Citado el: 12 de febrero de 2010.] <http://www.coit.es/publicaciones/bit/bit161/36-39.pdf>
- Caballero, Ismael. 2004.** *carixma. carixma.* [En línea] 2004. [Citado el: 29 de marzo de 2010.] <http://www.carixma.com/tutoriales/xbrl.htm>.
- Carriere, Tim Mallalieu y Jeromy. 2004.** *microsoft. microsoft.* [En línea] enero de 2004. [Citado el: 23 de marzo de 2010.] <http://www.microsoft.com/spain/interop/developers/dotnetinteroperability.aspx>.
- CAVSI. 2009.** [En línea] 2009. [Citado el: 8 de abril de 2010.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>
- Centelles, Miquel. 2005.** [En línea] 5 de 2005. [Citado el: 7 de abril de 2010.] <http://www.hipertext.net/web/pag264.htm>.
- CULIACAN, TECNOLOGICO DE. 2005.** *mitecnologico.* [En línea] 2005. [Citado el: 16 de febrero de 2010.] <http://www.mitecnologico.com/Main/EstandaresParaElManejoDeDatosEInformacion>.
- David Perry, Dewayne and Garlan. 1995.** *Introduction to the Special Issue on Software Architecture.* 1995.
- España, José María. 2007.** [En línea] 2007. <http://www.coit.es/publicaciones/bit/bit163/64-66.pdf>
- Faría, Daniela A. Torres. 2009.** *ldc. ldc.* [En línea] 11 de junio de 2009. [Citado el: 13 de febrero de 2010.] <http://www.ldc.usb.ve/~daniela/tcp-ip.pdf>.
- Garnet, Joan. 2007.** [En línea] 25 de 10 de 2007. [Citado el: 8 de 4 de 2010.] <http://www.joangarnet.com/blog/?p=415>

- Gastón Courtois, Pablo Telmo, Andrés Sosto, Maximiliano Bordón, Damián Gargiulo, Matías. 2006.** *ficcte. ficcte.* [En línea] 2006. [Citado el: 25 de marzo de 2010.] [http://ficcte.unimoron.edu.ar/wicc/Trabajos/III%20-%20isbd/718-Wicc\\_2006\\_Proyecto\\_Posadas.pdf](http://ficcte.unimoron.edu.ar/wicc/Trabajos/III%20-%20isbd/718-Wicc_2006_Proyecto_Posadas.pdf).
- González, M. (2008, 10 28).** Retrieved 2 23, 2010, from [http://programa.gobiernoenlinea.gov.co/apc-aa-files/5854534aee4eee4102f0bd5ca294791f/Documento\\_General\\_SECOP.pdf](http://programa.gobiernoenlinea.gov.co/apc-aa-files/5854534aee4eee4102f0bd5ca294791f/Documento_General_SECOP.pdf)
- Hernández, Lic. Anisleiby Fernández. 2007.** *scielo. scielo.* [En línea] mayo de 2007. [Citado el: 13 de abril de 2010.] [http://scielo.sld.cu/scielo.php?pid=S1024-94352007000500012&script=sci\\_arttext](http://scielo.sld.cu/scielo.php?pid=S1024-94352007000500012&script=sci_arttext)
- Isaza E., Gustavo A. 2007.** *ucaldas.* [En línea] 16 de noviembre de 2007. [Citado el: 31 de marzo de 2010.] [http://vector.ucaldas.edu.co/downloads/Vector2\\_6.pdf](http://vector.ucaldas.edu.co/downloads/Vector2_6.pdf)
- Juan Garbajosa Sopeña, F. J. (2008).** *madrimsd.* Retrieved from [http://www.madrimsd.org/informacionidi/biblioteca/publicacion/doc/VT/VT14\\_Tecnologias\\_software\\_orientadas\\_servicios.pdf](http://www.madrimsd.org/informacionidi/biblioteca/publicacion/doc/VT/VT14_Tecnologias_software_orientadas_servicios.pdf)
- Laboreo, Daniel Clemente. 2005.** *danielclemente.* [En línea] junio de 2005. [Citado el: 15 de febrero de 2010.] <http://www.danielclemente.com/html/estandares.html>
- Lebrún, Carlos Arturo Vega. 2007.** [En línea] 2007. [Citado el: 9 de 4 de 2010.] <http://www.eumed.net/tesis/2007/cavl/Beneficios%20de%20los%20servicios%20Web.htm>
- Marañón, Gonzalo Álvarez. 1998.** *iec. iec.* [En línea] 1998. [Citado el: 16 de marzo de 2010.] <http://www.iec.csic.es/cryptonomicon/java/ofuscacion.html>.
- Paul, Clements. 1996.** *A Survey of Architecture Description Languages.* 1996.
- Pérez, Francisco García. 2004.** *repositorio. repositorio.* [En línea] julio de 2004. [Citado el: 3 de mayo de 2010.] <http://repositorio.bib.upct.es/dspace/bitstream/10317/200/1/pfc1274.pdf>
- Perissé, Marcelo Claudio. 2009 .** *cyta. cyta.* [En línea] septiembre de 2009 . [Citado el: 15 de marzo de 2010.] [http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/semantic\\_web.pdf](http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/semantic_web.pdf).
- Poggi, Eduardo. 2006.** *siu. siu.* [En línea] 2006. [Citado el: 19 de noviembre de 2010.] <http://www.siu.edu.ar/infosiu/&edicion=19&nota=110?imprimir=1>.
- Reynoso, Carlos and Kiccillof, Nicolás. 2004.** *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft.* 2004.
- Ricardo Correa F., Daniella Caldana F., Daniel Santelices C., Omar Rebolledo M. 2009.** *Diccionario de Conceptos, Términos y Acrónimos relacionados con XBRL.* Santiago de Chile : s.n., 2009.

**Rico, Yajaira Dennise Limón. 2003.** [En línea] 2003. [Citado el: 7 de febrero de 2010.] <http://esinned7.iespana.es/descargas/1er%20REPORTE%20TT1.pdf>.

**Rivera, A. A. (julio de 2007).** Recuperado el 8 de junio de 2010, de <http://www.google.com/cu/url?sa=t&source=web&cd=5&ved=0CCgQFjAE&url=http%3A%2F%2Fwww.gobiernofacil.go.cr%2Fegob%2Fgobiernodigital%2Feventos%2FInteroperabilidad%2FImportanciaInteroperabilidad.ppt&rct=j&q=buenas+estrategias++interoperabilidad&ei=s-4MTKW6H8>

**Rodolfo Uribe, Piedad Castro, Sandra Barón e Iván Guarín. 2006.** *minhacienda. minhacienda.* [En línea] 24 de octubre de 2006. [Citado el: 30 de marzo de 2010.] <http://www.minhacienda.gov.co/portal/page/portal/MinHacienda/politicasapoyo/sectortecnologico/xbrl/PresentacionXBRLSectorPublico.pdf>

**XBRL España. (2005).** *Buenas Prácticas en proyectos XBRL.*

### Consultadas

**Acero, Fernando. 2009.** *Kriptopolis.* [Disponible en: <http://www.kriptopolis.org/interoperabilidad-4>]

**Álvarez, Miguel Angel. 2001.** *Desarrolloweb.* [Disponible en: <http://www.desarrolloweb.com/articulos/449.php>]

**Álvarez, Miguel Angel. 2003.** *maestrosdelweb.* [Disponible en: <http://www.maestrosdelweb.com/editorial/zendstudio/>]

**Billy, Carlos. 2004.** *Introducción a la Arquitectura de Software.*

**Borges, Alejandro E. 2007.** *La interoperabilidad y los estándares abiertos.* [Disponible en: <http://www.coit.es/publicaciones/bit/bit161/36-39.pdf>]

**Caballero, Ismael. 2004.** *carixma.* [Disponible en: <http://www.carixma.com/tutoriales/xbrl.htm>]

**Carriere, Tim Mallalieu y Jeromy. 2004.** *microsoft.* [Disponible en: <http://www.microsoft.com/spain/interop/developers/dotnetinteroperability.mspix>]

**CAVSI. 2009.** *cavsi.* [Disponible en: <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>]

**Centelles, Miquel. 2005.** *hipertext.* [Disponible en: <http://www.hipertext.net/web/pag264.htm>]

**CULIACAN, TECNOLÓGICO DE. 2005.** *mitecnologico.* [Disponible en: <http://www.mitecnologico.com/Main/EstandaresParaElManejoDeDatosEInformacion>]

- David Perry, Dewayne and Garlan. 1995.** *Introduction to the Special Issue on Software Architecture.*
- España, José María. 2007.** *Gestión Empresarial.* [Disponible en: <http://www.coit.es/publicaciones/bit/bit163/64-66.pdf>]
- Faría, Daniela A. Torres. 2009.** *Protocolos de Internet.* [Disponible en: <http://www ldc.usb.ve/~daniela/tcp-ip.pdf>]
- Garnet, Joan. 2007.** *joangarnet.* [Disponible en: <http://www.joangarnet.com/blog/?p=415>]
- Gastón Courtois, Pablo Telmo, Andrés Sosto, Maximiliano Bordón, Damián Gargiulo, Matías. 2006.** *Modelo de Sistema.* [Disponible en: [http://ficcte.unimoron.edu.ar/wicc/Trabajos/III%20-%20isbd/718-Wicc\\_2006\\_Proyecto\\_Posadas.pdf](http://ficcte.unimoron.edu.ar/wicc/Trabajos/III%20-%20isbd/718-Wicc_2006_Proyecto_Posadas.pdf)]
- González, M. 2008.** *SECOPI.* [Disponible en: [http://programa.gobiernoenlinea.gov.co/apc-aa-files/5854534aee4eee4102f0bd5ca294791f/Documento\\_General\\_SECOPI.pdf](http://programa.gobiernoenlinea.gov.co/apc-aa-files/5854534aee4eee4102f0bd5ca294791f/Documento_General_SECOPI.pdf)]
- Hernández, Eduardo Ochoa. Mapas conceptuales. 2009.** [Disponible en: <http://dieumsnh.qfb.umich.mx/ciees2009/Reforma%20PDF/Mapas%20conceptuales2009.pdf>]
- Hernández, Lic. Anisleiby Fernández. 2007.** *scielo.* [Disponible en: [http://scielo.sld.cu/scielo.php?pid=S1024-94352007000500012&script=sci\\_arttext](http://scielo.sld.cu/scielo.php?pid=S1024-94352007000500012&script=sci_arttext)]
- Isaza E., Gustavo A. 2007.** *Estándares de seguridad basados en XML para servicios web y web semántica.* [Disponible en: [http://vector.ucaldas.edu.co/downloads/Vector2\\_6.pdf](http://vector.ucaldas.edu.co/downloads/Vector2_6.pdf)]
- Juan Garbajosa Sopena, F. J. (2008).** *Tecnologías software orientadas servicios.* [Disponible en: <http://www.carixma.com/tutoriales/xbrl.htm>][http://www.madrimasd.org/informacionidi/biblioteca/publicacion/doc/VT/VT14\\_Tecnologias\\_software\\_orientadas\\_servicios.pdf](http://www.madrimasd.org/informacionidi/biblioteca/publicacion/doc/VT/VT14_Tecnologias_software_orientadas_servicios.pdf)]
- Laboreo, Daniel Clemente. 2005.** *danielclemente.* [Disponible en: <http://www.danielclemente.com/html/estandares.html>]
- Lebrún, Carlos Arturo Vega. 2007.** *eumed.* [Disponible en: <http://www.eumed.net/tesis/2007/cavl/Beneficios%20de%20los%20servicios%20Web.htm>]
- Marañón, Gonzalo Álvarez. 1998.** *iec.* [Disponible en: <http://www.iec.csic.es/criptonomicon/java/ofuscacion.html>]
- Paul, Clements. 1996.** *A Survey of Architecture Description Languages.*
- Pérez, Francisco García. 2004.** *Evaluación del rendimiento de arquitecturas software.* [Disponible en: <http://repositorio.bib.upct.es/dspace/bitstream/10317/200/1/pfc1274.pdf>]

- Perissé, Marcelo Claudio. 2009 .** *Gestión del conocimiento.* [Disponible en: [http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/semantic\\_web.pdf](http://www.cyta.com.ar/biblioteca/bddoc/bdlibros/semantic_web.pdf)]
- Perojo, Keilyn Rodríguez. 2006.** *Web Semántica.* [Disponible en: <http://www.bibliociencias.cu/gsd/collect/eventos/index/assoc/HASH01a3/6ff7e68d.dir/doc.pdf>]
- Poggi, Eduardo. 2006.** *siu.* [Disponible en: <http://www.siu.edu.ar/infosiu/&edicion=19&nota=110?imprimir=1>]
- Pressman, Roger S. 2002.** *Ingeniería de Software, un enfoque práctico.*
- Real Academia Española. DICCIONARIO DE LA LENGUA ESPAÑOLA - Vigésima segunda edición.** [Disponible en: <http://buscon.rae.es/drae/>]
- Reynoso, Carlos and Kiccillof, Nicolás. 2004.** *Estilos y Patrones en la Estrategia de Arquitectura de Microsoft.*
- Ricardo Correa F., Daniella Caldana F., Daniel Santelices C., Omar Rebolledo M. 2009.** *Diccionario de Conceptos, Términos y Acrónimos relacionados con XBRL.*
- Rico, Yajaira Dennise Limón. 2003.** *Sistema de modelado de procesos de negocio.* [Disponible en: <http://esinned7.iespana.es/descargas/1er%20REPORTE%20TT1.pdf>]
- Rivera, A. A. (julio de 2007).** *Proyecto de Interoperabilidad de e-Gobierno.* [Disponible en: <http://www.google.com/cu/url?sa=t&source=web&cd=5&ved=0CCgQFjAE&url=http%3A%2F%2Fwww.gobiernofacil.go.cr%2Fegob%2Fgobiernodigital%2Feventos%2FInteroperabilidad%2FImportanciaInteroperabilidad.ppt&rct=j&q=buenas+estrategias++interoperabilidad&ei=s-4MTKW6H8>]
- Rodolfo Uribe, Piedad Castro, Sandra Barón e Iván Guarín. 2006.** *XBRL: El idioma universal financiero y de negocios.* [Disponible en: <http://www.minhacienda.gov.co/portal/page/portal/MinHacienda/politicasapoyo/sectortecnologico/xbrl/PresentacionXBRLSectorPublico.pdf>]
- Tejada, Licenciada Eugenia Saenz de. 2005.** *paho.* [Disponible en: <http://www.paho.org/Spanish/AD/THS/EV/Blood-guiametodologica9.pdf>]
- XBRL España. (2005).** *Buenas Prácticas en proyectos XBRL.* [Disponible en: <http://xbrl.es/formacion/formacion.html>]

### GLOSARIO DE TÉRMINOS

**ARP:** Protocolo de Resolución de Direcciones.

**B2B:** Negocios para Negocios.

**BPM:** Gestión de Procesos de Negocio.

**CVS:** Sistema Concurrente de Versiones.

**EML:** Lenguaje de Metadatos Ecológicos.

**ERP:** Planificación de Recursos Empresariales.

**FRTA:** Arquitectura de Taxonomías para la Información Financiera.

**FTP:** Protocolo de Transferencia de Archivos.

**HTML:** Lenguaje de Marcado de Hipertexto.

**HTTP:** Protocolo de Transferencia de Hipertexto.

**HTTPS:** Protocolo Seguro de Transferencia de Hipertexto.

**IEEE:** Instituto de Ingenieros Eléctricos y Electrónicos.

**IETF:** Grupo de Tareas de Ingeniería de Internet

**IP:** Protocolo de Internet.

**ISO:** Organización Internacional de Estándares.

**MARC:** Catalogación Legible por la Máquina.

**MOF:** Manual de Organización y Funciones.

**MOM:** Software Intermediario (Middleware) Orientado a Mensaje.

**MVC:** Modelo Vista Controlador.

**ODBC:** Estándar de conectividad Abierta de Base de Datos.

**ORM:** Mapeo Objeto-Relacional.

**OSI:** Interconexión de Sistemas Abiertos.

**PHP:** Página de Inicio Personal.

**PKI:** Infraestructura de Clave Pública.

**RDF:** Marco de Descripción de Recursos.

**SECOPE:** Sistema Electrónico para la Contratación Pública.

**SGBD:** Sistema Gestor de Bases de Datos.

**SICE:** Sistema de Información para la Contratación Estatal.

**SIIF:** Sistema Integrado de Información Financiera.

**SMTP:** Protocolo Simple de Transferencia de Correo Electrónico.

**SOAP:** Protocolo de Acceso a Objetos Simples.

**SSL:** Capa de Conexión Segura.

**TCP:** Protocolo de Control de Transmisión.

**TIC:** Tecnologías de la Información y las Comunicaciones.

**TLS:** Seguridad de la Capa de Transporte.

**UCI:** Universidad de las Ciencias Informáticas.

**UML:** Lenguaje Unificado de Modelado.

**URL:** Localizador Uniforme de Recurso.

**WSDL:** Lenguaje de Descripción de Servicios Web.

**XBRL:** Lenguaje Extensible de Informes de Negocios.

**XMI:** XML Intercambio de Metadatos.

**XML:** Lenguaje Extensible de Marcado.

**Estructura metadatos:** Información altamente estructurada que describe características de los datos.

**Framework:** Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado.

**Guía Metodológica:** Se centra en desarrollar los fundamentos conceptuales y la metodología para un proceso, desde el punto de vista conceptual y normativo, proporcionando los elementos metodológicos necesarios para la planeación y el desarrollo del proceso mediante un conjunto de pasos a seguir, donde se describe en qué consiste cada uno.

**Interoperabilidad:** Característica que poseen los ordenadores que les permite su interconexión y funcionamiento en conjunto de manera compatible. Es la capacidad de los sistemas de comunicarse con otros externos para transmitir y recibir información, notificar o gestionar sucesos, así como para exportar e importar datos.

**Plataforma de Interoperabilidad:** Ofrece servicios a los actores involucrados en las transacciones y trámites, en la que se comprometen las entidades. Actúa como mediador entre sistemas, permitiendo la conexión y el intercambio de información de forma segura y confiable.

**Procesos:** Conjunto de tareas relacionadas lógicamente llevadas a cabo para lograr un resultado de negocio definido.

**Servicios web:** Son aplicaciones que utilizan estándares para el transporte, codificación y protocolo de intercambio de información.

**Subprocesos:** Parte de un proceso de mayor nivel que tiene su propia meta, propietario, entradas y salidas.

**Taxonomía:** Tipo de vocabulario controlado en que todos los términos están conectados mediante algún modelo estructural (jerárquico, arbóreo...) y especialmente orientado a los sistemas de navegación, organización y búsqueda de contenidos de los sitios web.