

Universidad de las Ciencias

Informáticas

Facultad 3



**Título: Propuesta de un lenguaje de ejecución para el
proceso de interoperabilidad en los sistemas que
emplean el marco de trabajo Sauxe.**

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

Autor: Rogelio Meadón Mecí

Tutores: Ing. Magdanis Galván Rey

Ing. Javier Ruiz Durán

Co-tutor: Ing. Pedro Manuel Nogales Cobas

La Habana, junio de 2012



“Hay en el mundo un lenguaje que todos comprenden: es el lenguaje del entusiasmo, de las cosas hechas con amor y con voluntad, en busca de aquello que se desea o en lo que se cree”.

Paulo Coelho.

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los 29 días del mes de junio del año 2012.

Rogelio Meadón Mecí

Ing. Magdanis Galván Rey

Ing. Javier Ruiz Duran

Ing. Pedro Manuel Nogales Cobas

Datos de Contacto

Tutor: Ing. Magdanis Galván Rey.

Formación académica: Ingeniero en Ciencias Informáticas.

Centro laboral: Universidad de las Ciencias Informáticas (UCI).

Correo electrónico: mgalvan@uci.cu

Dirección: Carretera San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba, Código postal 19370.

Tutor: Ing. Javier Ruiz Duran.

Formación académica: Ingeniero en Ciencias Informáticas.

Centro laboral: Universidad de las Ciencias Informáticas (UCI).

Correo electrónico: jduran@uci.cu

Dirección: Carretera San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba, Código postal 19370.

Tutor: Ing. Pedro Manuel Nogales Cobas.

Formación académica: Ingeniero en Ciencias Informáticas.

Centro laboral: Universidad de las Ciencias Informáticas (UCI).

Correo electrónico: pmnogales@uci.cu

Dirección: Carretera San Antonio de los Baños, Torrens, Municipio Boyeros, Ciudad de La Habana, Cuba, Código postal 19370.

A mi mamá por siempre estar presente y brindarme esa fuerza y el optimismo incluso cuando hasta yo creía que no podía. Por ser mi guía en estos 25 años, siempre te voy a estar agradecido por este regalo que me has hecho, las palabras no alcanzan para mostrarte cuanto te quiero y cuan agradecido estoy.

A mi hermanito por inspirarme a ser siempre un ejemplo para él y porque sé que realizará mi sueño de verlo hecho un gran oficial.

A mi abuelita Hilda que sé que este es su sueño y que no hay día que pase que no piense en ella.

A mi papá Abel por estar siempre atento de mí, por brindarme su apoyo y confianza todos estos años y por quererme como su hijo muchas gracias yo te quiero como mi padre.

A mi papá Rogelio que sé que me quiere mucho y yo lo quiero también.

A mis tías Violeta, Ana Rosa, Mercedes, Consuelo, Hildita, Adís, Concha, Marcia a todas.

A mis tutores el Javi y Pedro que vienen luchando conmigo desde tercero, gracias por su apoyo. Especialmente a Magdanis por ser mi segunda madre y ayudarme tanto en la realización de esta tesis.

El fruto de toda una carrera de estudio se los dedico a mi familia en especial a mi madre por brindarme su amor en todos estos años.

Este trabajo se lo quiero dedicar también a 3 personas muy importantes que sé que estarían muy orgullosas de mi como yo siempre lo estuve de ellos y que sepan que por más que pasen los años nunca los olvido.

A mis Abuelitos Bella, Eva y Virgilio por estar siempre presente en mis pensamientos y por cuidar de mí.

Resumen

Desde el surgimiento de la informática como ciencia, uno de los principales problemas enfrentados fue el intercambio de información entre sistemas. La informatización de varias esferas de la vida social aumentó la magnitud de la interoperabilidad a niveles considerables.

En el siguiente trabajo se propone un lenguaje de ejecución para la estandarización del proceso de interoperabilidad en los sistemas que utilizan el marco de trabajo Sauxe, y de esta forma garantizar que se pueda realizar viablemente la ejecución del proceso de interoperabilidad en dicho marco de trabajo.

Para lograr este objetivo se realizó un estudio de la interoperabilidad en las aplicaciones web de gestión como principales aplicaciones desarrolladas en el marco de trabajo Sauxe. Además se analizaron algunos de los lenguajes de ejecución de procesos de negocio, y su relación con los estándares de intercambio de información que soporta Sauxe.

Una vez analizado toda la bibliografía referente al tema, y las particularidades de Sauxe como marco de trabajo orientado a componentes, se analizaron las características particulares del proceso de interoperabilidad en cada una de las soluciones desarrolladas sobre Sauxe, con el objetivo de obtener rasgos comunes y elementos capaces de describir la ejecución de los procesos, para definir la estructura genérica del proceso de interoperabilidad.

La propuesta de solución se validó mediante el método: criterio de expertos arrojando resultados positivos en cuanto a dimensión de la problemática e idoneidad de la solución.

PALABRAS CLAVE: procesos, interoperabilidad, lenguaje de ejecución, Sauxe, estandarización.

Índice de ilustraciones

Figura 1: Arquitectura de las capas de Sauxe.....13

Figura 2: Mapa Conceptual de la solución de interoperabilidad de Sauxe.....22

Figura 3: Estructura del LEPI.....29

Figura 4: Declaración del elemento proceso.....30

Figura 5: Declaración del elemento exportar.....30

Figura 6: Declaración del elemento evento de descripción.....31

Figura 7: Declaración del elemento vista.....31

Figura 8: Declaración del elemento formulario.....32

Figura 9: Declaración de los botones.....32

Figura 10: Declaración del elemento servicio.....33

Figura 11: Declaración del elemento parámetro.....33

Figura 12: Declaración del elemento manejador de evento.....33

Figura 13: Declaración del elemento Excepción.....33

Figura 14: Declaración del elemento variable.....34

Figura 15: Declaración del elemento Grid.....35

Figura 16: Declaración del elemento descargar.....35

Figura 17: Declaración del elemento subir archivo.....35

Figura 18: Declaración del elemento importar.....36

Figura 19: Declaración del elemento marco.....36

Figura 20: Declaración del elemento Panel.....36

Figura 21: Declaración del elemento vista.....37

Figura 22: Declaración del elemento estructura.....37

Figura 23: Tabla de elementos del LEPI.....52

Figura 24: Tabla de competencias de los expertos.....55

Figura 25: Tabla de aspectos para confección de cuestionario.....58

Figura 26: Tabla de evaluación de los criterios por los especialistas.....59

Figura 27: Tabla de cálculo de la variación media.....60

Figura 28: Promedio de evaluación por especialistas.....63

Figura 29: Evaluación de la solución por los especialistas.....64

Figura 30: Encuesta realizada a expertos.....70

Figura 31: Encuesta realizada a expertos.....71

Figura 32 : Encuesta realizada a expertos.....72

Figura 33: Encuesta realizada a expertos.....73

Figura 34: Encuesta realizada a expertos.....74

Figura 35: Encuesta realizada a expertos.....75

Figura 36: Descripción de la ejecución del subproceso de exportación de comprobantes de operaciones.....76

Tabla de Contenido

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	5
INTRODUCCIÓN	5
1.1 Definiciones de interoperabilidad	5
1.2 Dimensiones de la interoperabilidad	6
1.3 Surgimiento y estado actual de la interoperabilidad	7
1.4 Formas de interoperar	8
1.5 Buenas prácticas de interoperabilidad	10
1.6 Ventajas de la interoperabilidad.....	11
1.7 Estándares de interoperabilidad	11
1.8 Marco de trabajo Sauxe.....	12
1.9 Lenguajes de referencia	15
CAPITULO 2: PROPUESTA DE SOLUCIÓN.....	21
INTRODUCCIÓN	21
2.1 Mapa conceptual de la solución de interoperabilidad en Sauxe	21
2.2 Descripción del proceso de interoperabilidad.....	23
2.3 Subproceso de exportación	24
2.3.1 Pasos para exportar datos usando LEPI.....	24
2.4 Subproceso de importación	25
2.4.1 Pasos para importar datos usando el LEPI.....	25
2.5 Requisitos del LEPI	26
2.6 Reglas sintácticas del LEPI.....	27
2.7 Estructura del LEPI.....	28
2.8 DTD asociado al LEPI.....	29
2.9 Simbología del LEPI	37
CONCLUSIONES PARCIALES	52
CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN	53
INTRODUCCIÓN.....	53
3.1 Proceso de validación.....	53
3.2 Elección de los especialistas	53
3.3 Elaboración del cuestionario.....	55
3.4 Desarrollo del proceso de validación	58

3.5 Solución aplicada a un proceso	64
CONCLUSIONES	65
RECOMENDACIONES	66
BIBLIOGRAFÍA	67
REFERENCIAS BIBLIOGRÁFICAS	69
ANEXOS	70

INTRODUCCIÓN

El desarrollo de la humanidad, la globalización mundial y la evolución de las tecnologías de la información y las comunicaciones (TIC), requieren que cada día sea necesaria la disponibilidad, confidencialidad e integridad de la información, así como la trasmisión puntual de datos entre distintas entidades. La acelerada carrera en la producción de software hace que las principales empresas vanguardias en esta esfera, en cumplimiento con sus requisitos específicos, elaboren un cúmulo de formatos digitales incompatibles entre sí, que luego hacen un tanto difícil el intercambio de información entre otras entidades o usuarios finales. Este fenómeno tiende a convertir el proceso de intercambio en una labor manual para profesionales que interactúan con dichos sistemas, además de causar grandes daños económicos en dependencia del escenario.

Cuba a partir de la década de los noventa encaminó sus esfuerzos hacia lograr la recuperación y desarrollo sostenible de su economía. Para ello se ha trazado una estrategia de digitalización y optimización de los procesos que se realizan en las empresas cubanas, punto de arranque en busca del desarrollo económico y social.

La Universidad de las Ciencias Informáticas (UCI) juega un papel primordial en los planes de desarrollo tanto económico como científico de la nación. Entre sus principales tareas se encuentra el desarrollo de un Sistema de Planificación de Recursos Empresariales (ERP) denominado Cedrux. Dicho sistema juega un papel de suma relevancia ya que su principal objetivo es el “de manejar muchas de las prácticas de los negocios asociados con las operaciones de producción, los aspectos de distribución de empresas además de la elaboración de bienes o servicios”. [1]

“Sauxe como marco de trabajo contiene un conjunto de componentes reutilizables que permiten lograr una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo” [2] Implementando una solución sobre Sauxe, se garantizan la seguridad, la integración, la administración de transacciones, la interoperabilidad, la concurrencia, entre otros. [2]

Como entorno de desarrollo, Sauxe brinda una serie de componentes, los cuales pueden ser utilizados por las aplicaciones que son desarrolladas sobre el mismo. Entre estos componentes se encuentra el componente de interoperabilidad, que al ser empleado por los subsistemas que integran el ERP, posibilita durante la implantación del mismo, la absorción gradual de la información contenida en el software que se

desea reemplazar, asegurando la retroalimentación entre sistemas, así como la capacidad de redistribuir los datos de forma rápida.

Actualmente el componente que garantiza la interoperabilidad no cuenta con la capacidad de entender una estructura común de los procesos de interoperabilidad de los sistemas que se implementan sobre el marco de trabajo Sauxe. Esto trae consigo la implementación de componentes independientes para cada sistema que desee intercambiar información, reutilización de código en muchos casos, y un gran esfuerzo por parte de los desarrolladores y demás personal involucrado.

Valorando todo lo previamente expuesto se plantea como **problema a resolver**: la solución de interoperabilidad en el marco de trabajo Sauxe no es flexible a la concepción de nuevos procesos para el intercambio de información.

Por lo que el **objeto de estudio** comprende el proceso de interoperabilidad de software en sistemas de gestión.

Sosteniendo como **campo de acción**: la interoperabilidad en el marco de trabajo Sauxe.

Para darle solución a dicha problemática se define como **objetivo general**: desarrollar un lenguaje que permita la ejecución del proceso de interoperabilidad en los sistemas que emplean el marco de trabajo Sauxe.

Objetivos específicos:

- Definir rasgos comunes que enmarcan los lenguajes de ejecución de procesos.
- Definir las características del proceso de interoperabilidad de todos los sistemas que emplean el marco de trabajo Sauxe.
- Definir un lenguaje de ejecución del proceso de interoperabilidad en los sistemas que utilizan el marco de trabajo Sauxe.
- Validar el lenguaje de ejecución del proceso de interoperabilidad en los sistemas que utilizan el marco de trabajo Sauxe.

Como **idea a defender** se plantea que: el desarrollo de un lenguaje común permitirá estandarizar la ejecución del proceso de interoperabilidad en los sistemas que se implementan sobre el marco de trabajo Sauxe.

Para el desarrollo del trabajo se utilizaron diferentes **métodos científicos de investigación**, conceptualmente representan la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones.[3]

Los métodos utilizados fueron:

Los **métodos empíricos** permitieron describir y explicar las características del fenómeno en estudio. Dentro de estos, se aplicaron los métodos particulares con el objetivo de recolectar los datos necesarios para identificar la problemática y las causas de esta, así como determinar la magnitud de su influencia.

- **La revisión de documentos:** para la determinación del estado del arte del objeto de investigación.
- **Las entrevistas:** fueron vitales en establecer los elementos necesarios para avalar los conceptos que se manejan en la investigación, medir el alcance y la importancia que tiene la temática. Captar la información cualitativa y cuantitativa del fenómeno, conocer los criterios sobre la forma en que se organiza y se lleva a cabo el proceso de interoperabilidad en el marco de trabajo, así como las posibles soluciones que se proponen en la investigación.
- **La observación:** para ampliar y corroborar información obtenida a través de las entrevistas realizadas.

Los **métodos teóricos** permitieron comprender el fenómeno que se estudia, su evolución y proponer las mejoras a los problemas que se identificaron.

- **Analítico – Sintético:** este método posibilitó la definición de términos que permitieron realizar el estudio de los lenguajes de ejecución de procesos y la interoperabilidad en las aplicaciones web de gestión, de manera que se pudo analizar y sintetizar toda esta información, además de tomar los elementos más relevantes teniendo en cuenta el objeto de estudio de la investigación.
- **Hipotético – Deductivo:** este método desempeñó un papel importante en el proceso de verificación de la idea a defender, permitió establecer conclusiones y predicciones a partir de la solución propuesta.
- **Histórico – Lógico:** posibilitó conocer los antecedentes históricos de estudios de interoperabilidad en aplicaciones web de gestión el mundo y en la UCI.

Este trabajo consta de: introducción, 3 capítulos, conclusiones, recomendaciones, bibliografía, referencias bibliográficas y anexos.

Capítulo 1: **Fundamentación teórica:** contiene un estudio de los conceptos generales que permiten comprender temas relacionados con los lenguajes de ejecución de procesos a nivel internacional. Así

como la arquitectura, los componentes del marco de trabajo Sauxe y el proceso de interoperabilidad. Además se hacen referencias conceptuales sobre la interoperabilidad en las aplicaciones web de gestión. También se analiza como influiría el uso de algunos de los lenguajes de ejecución de procesos de negocio en la estandarización de la ejecución del proceso de interoperabilidad en el marco de trabajo Sauxe.

Capítulo 2: **Propuesta de solución:** se propone el desarrollo de un lenguaje de ejecución, capaz de describir la interoperabilidad en Sauxe. Se definen los requisitos, la estructura, reglas sintácticas y simbología del mismo.

Capítulo 3: **Validación de la propuesta de solución:** se evaluará la capacidad descriptiva del lenguaje de ejecución, de acuerdo a la integración con Sauxe, facilidad de uso y correctitud de la solución. Además se aplica la solución a un caso de estudio previamente definido.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

INTRODUCCIÓN

En el presente capítulo se abordan los conceptos fundamentales para el entendimiento de la temática, que a su vez servirán de base para el desarrollo de la investigación. También se incluye el estudio de los lenguajes de ejecución de procesos, específicamente BPEL¹ y WS-BPEL² como referencia para la ejecución de procesos de negocio. Además se analiza la interoperabilidad en las aplicaciones web de gestión, formas de interoperar, tendencias actuales, dimensiones y buenas prácticas de interoperabilidad. Se estudia la viabilidad del empleo de los lenguajes de ejecución de procesos de negocio en el marco de trabajo Sauxe. Igualmente se hace un análisis bibliográfico de la tecnología a emplear.

1.1 Definiciones de interoperabilidad

Interoperabilidad no es solamente integración de sistemas. No hace referencia sólo al intercambio de datos ni contempla simplemente una definición de tecnología. Es un amplio término que puede ser interpretado de varias maneras en dependencia del contexto. A pesar de su variedad de ámbitos está muy ligada a las TIC, y ante la ausencia de una definición clara algunos de los más experimentados y prestigiosos autores la han definido como:

“La capacidad de los sistemas heterogéneos y de los procedimientos a los que éstos dan soporte, de compartir datos y posibilitar el intercambio de información y conocimiento entre ellos”. [4]

Algunas de las definiciones sin embargo vinculan la interoperabilidad a un ámbito específico, sin tener en cuenta que el término puede tomar matices diferentes, incluso puede adicionar argumentos importantes en la búsqueda de una definición capaz de abordar todos los ámbitos tecnológicos posibles.

Organizaciones reconocidas en la creación de estándares internacionales como la Organización Internacional de Normalización (ISO) definen la interoperabilidad como:

“La interoperabilidad es la capacidad que tiene un producto o un sistema, cuyas interfaces son totalmente conocidas, para funcionar con otros productos o sistemas existentes o futuros y eso sin restricción de acceso o de implementación”. [5]

¹ BPEL: Lenguaje de ejecución de procesos de interoperabilidad.

² WS-BPEL: Lenguaje de ejecución de procesos de interoperabilidad orientados a servicios web.

Unos de los conceptos más completos a consideración del autor teniendo en cuenta el ámbito de la investigación es el propuesto por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) quienes definen la interoperabilidad como:

“La capacidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada”. [6]

Añadiendo además tres vías diferentes por las cuales dos sistemas pueden lograr la interoperabilidad.

- Lenguaje común. Que ambos se comuniquen en un idioma común.
- Traducción directa. Que ambos tengan traductores de su idioma al idioma del otro.
- Lenguaje de referencia. Que exista un lenguaje de referencia y ambos tengan traductores ha dicho idioma de referencia.

Seleccionar una de estas formas para desarrollar un proceso de interoperabilidad no delimita la utilización de cualquiera de las otras existentes, aunque la no presencia de un lenguaje común, traducción directa o lenguaje de referencia significa que el sistema o proceso de negocio no es interoperable.

Después de analizar los criterios de algunos expertos en el tema y teniendo en cuenta el ámbito en el cual se desarrolla la investigación, el autor emite su propia definición de interoperabilidad:

“La habilidad, de dos o más sistemas y de los procesos de negocio a los que soportan, de intercambiar información, y utilizar dicha información en su ámbito de desarrollo.”

Esta definición será la referencia a la hora de abordar el término de interoperabilidad y su relación con la ejecución de procesos de negocio a lo largo de la investigación.

1.2 Dimensiones de la interoperabilidad

En el ámbito que comprende las aplicaciones web específicamente de gestión, la interoperabilidad constituye una condición necesaria y un requisito que deben cumplir para tener un completo acceso a la información disponible y una buena compatibilidad con el entorno donde son desplegadas. En muchas ocasiones la interoperabilidad se ve limitada al entorno y la problemática donde se desenvuelve, este fenómeno genera que el término se fragmente en busca de adaptarse y aun así mantener el objetivo común, intercambio de información. Según la problemática que se desea solucionar, la interoperabilidad puede tomar tres dimensiones:

- **La interoperabilidad organizacional.** Está relacionada con las metas que se desean conseguir, el modelado de los procesos y la necesaria colaboración entre los elementos participantes, que deben poder intercambiar información a pesar de tener distintas estructuras internas y procesos.[4]
- **La interoperabilidad semántica.** “Se preocupa de asegurar que el significado preciso de la información que se intercambia es entendido por otra aplicación que no fue diseñada inicialmente para ese propósito”. [4] Este tipo de interoperabilidad es la que permite que las aplicaciones recombinaen información de varias fuentes y la puedan procesar de una forma coherente.
- **La interoperabilidad técnica.** “Se preocupa de los problemas que existen para intercomunicar sistemas y servicios heterogéneos”. [4] En esta interoperabilidad existen aspectos claves como el uso de interfaces y estándares abiertos, servicios de interconexión, integración de datos, presentación de datos e intercambio de información, accesibilidad y la garantía de seguridad de los servicios.

Específicamente la interoperabilidad organizacional marca las dimensiones de la solución, ya que se deberá lograr el modelamiento de procesos de interoperabilidad y la colaboración entre los sistemas participantes, y todo con el objetivo que dichos procesos puedan lograr un eficiente intercambio de información aunque presenten estructuras distintas. También se moverá en el ámbito o las dimensiones de la interoperabilidad semántica debido a que varios sistemas deben ser capaces de adquirir información de varias fuentes y dicha información debe ser procesada coherentemente.

1.3 Surgimiento y estado actual de la interoperabilidad

Evolución de la interoperabilidad

El desarrollo de la interoperabilidad en aras de lograr el intercambio de información entre sistemas homólogos con tecnologías heterogéneas, ha sido una tarea difícil desde los primeros pasos de la informática como ciencia. A medida que se fusionaba la informática con el resto de las esferas (sociales y económicas) fue creciendo la cantidad de información que se procesaba y con ellos la necesidad de sistemas más eficientes, que además fueran capaces de intercambiar información entre ellos.

La llegada de TCP/IP (protocolo abierto de interconexión entre sistemas), permitió un gran avance a la interoperabilidad. Dichos protocolos, 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.[1]

El momento cumbre 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 un Lenguaje de Etiquetado Extensible (XML).[1]

Actualmente la interoperabilidad entre sistemas informáticos se ha incrementado considerablemente. Sin embargo, “incide también en niveles de comunicación e intercambios gubernamentales; y sobre estos aspectos los gobiernos enfrentan algunas dificultades al compartir e intercambiar información para sus procesos de negocio”.[1]

La interoperabilidad incurre directamente en los elementos que hacen a un estado moderno, en particular a los asociados a una mejor atención al ciudadano; esto significa, que el estado opere desde los requerimientos de estos y no desde su oferta.[1]

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.

Interoperabilidad en Cuba

Partiendo de un estudio realizado se puede afirmar que existe cierto avance en Cuba en el tema de la interoperabilidad debido a la importancia y relevancia que está asumiendo el intercambio de datos entre sistemas y específicamente aplicaciones web de gestión. Para el caso de los sistemas ERP usados en Cuba existe una problemática clara, y es la necesidad imperiosa de poder intercambiar información y usar esa información intercambiada.

A razón de esta problemática se ha venido dando pasos importantes en consolidar las estratégicas de interoperabilidad en Cuba, ejemplo de ello el “Primer taller nacional de Interoperabilidad en el año 2009, organizado por el equipo de desarrollo del sistema CEDRUX encargado del tema. En este taller se acordó el estándar de comunicación y las formas interoperabilidad entre las empresas participantes”.[7]

1.4 Formas de interoperar

El desarrollo de la interoperabilidad en aplicaciones web de gestión y específicamente en ERP desde el punto de vista funcional, dependen mucho de los mecanismos utilizados, los mismos constituyen estrategias nombradas “clases tecnológicas”, las cuales poseen particularidades 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.

Enlaces directos entre aplicaciones. Existen casos concretos, en los que se necesita priorizar el rendimiento sobre el volumen de información en la 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.[4]

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.[4]

Servicios web XML. 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 suponen una solución sencilla, independiente de la tecnología, que evita en su mayor parte las dependencias de plataforma y fabricante.[4]

Como se ha mencionado, una solución asentada en MOM puede brindar una infraestructura más apropiada, pero el enlace directo entre aplicaciones suele ser un poco más eficiente si se pretende cuidar aspectos como la integración y rendimiento de los sistemas. En cualquier caso, una estrategia de arquitectura orientada a servicios web, compuesta por servicios que cooperan en estrecha colaboración suele ser el modelo más flexible y ágil. La estrategia basada en estándares que representan los servicios web tiene el mayor potencial a largo plazo.

Debido a que la presente investigación está centrada en la estandarización del proceso de interoperabilidad, es relevante el conocimiento de estas estrategias o formas de interoperar, pues la solución propuesta contiene características de las estrategias *enlace directo entre aplicaciones* y *servicios web XML*.

1.5 Buenas prácticas de interoperabilidad

A nivel internacional muchas de las soluciones o sistemas en el mercado son un fracaso en cuanto a interoperabilidad refiere, debido a que no se toma la interoperabilidad como un todo sino como un problema de estandarización a la hora del intercambio de información. Este fenómeno se produce porque los implementadores se enfocan en los protocolos, taxonomía y formato de intercambio, pero no definen las dimensiones de sus soluciones. Para lograr un proceso de interoperabilidad se deben tener en cuenta sus dimensiones organizacionales, semánticas y tecnológicas, y sobre ellas aplicar un conjunto de estrategias o buenas prácticas relacionadas a la problemática existente. Autores experimentados en la temática proponen algunas de estas estrategias o buenas prácticas.

➤ Hacer uso de herramientas criptográficas: como los ofuscadores de código que se utilizan para cifrar el contenido de la información que viajará por la red.[8]

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”. [8] Evita que la información pueda ser modificada por un intruso, ocultándola de tal forma que es prácticamente imposible descifrarla, e incluso se dificulta enormemente interpretarla aún si se obtiene el código fuente.

➤ Siempre que sea posible, se debería optar por el uso de software libre. No en vano, el software libre suele usar preferentemente estándares abiertos. La disponibilidad del código fuente también tiene como consecuencia el debate abierto y democrático sobre sus especificaciones y estándares, lo que suele hacer que el software libre sea más robusto e interoperable que las aplicaciones privativas equivalentes.[4] No será muy complicado encontrar aplicaciones libres que cumplan con los criterios del marco de interoperabilidad, por lo que se debería considerar en igualdad de condiciones con sus contrapartidas propietarias en todo plan de adquisición.

➤ Se debe utilizar un lenguaje común para que los diferentes sistemas integrales de gestión puedan interactuar e intercambiar información de manera óptima y eficiente.[4]

➤ Se debe construir una plataforma como solución de interoperabilidad para el intercambio de información, independiente de hardware o software.[4]

Muchas de estas prácticas actualmente son implementadas por el marco de trabajo Sauxe, la solución que se propone cumplirá con las mismas.

1.6 Ventajas de la interoperabilidad

Desarrollar una plataforma de interoperabilidad, basada en el conocimiento de las dimensiones de la misma, trae grandes beneficios en cuanto a intercambio de información se trata. Entre las numerosas ventajas se pueden mencionar:[1]

- Los sistemas intercambian información de forma transparente, coherente y segura.
- Logra la mayor cooperación entre sistemas.
- Ignora la heterogeneidad de los sistemas.
- 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.

1.7 Estándares de interoperabilidad

La estandarización no es una tarea trivial, implica la revisión de todos los procesos y sistemas existentes y los que puedan surgir en un futuro, así como la detección de áreas donde es oportuna y beneficiosa la misma, además se necesitan conocimientos referentes a los estándares disponibles que cubren las áreas seleccionadas.

Uno de los beneficios de aplicar la estandarización de procesos, es la formalización de los mismos, lo que permite organizarlos, simplificarlos, medirlos y compararlos. En definitiva, la estandarización repercute positivamente en la capacidad de gestionar cualquier sistema y sobre todo los sistemas de gestión.

Para un mejor entendimiento de la temática se hace necesario conocer la definición de estándar y algunos de los estándares de interoperabilidad existentes.

Definición de estándar: “Un estándar es un conjunto de reglas normalizadas que describen los requisitos que deben ser cumplidos por un producto, proceso o servicio, con el objetivo de establecer un mecanismo base para permitir que distintos elementos hardware o software que lo utilicen, sean compatibles entre sí”. [9]

Una vez analizada la definición emitida por la W3C³, en cuanto a estándares se trata, el autor define el término de estandarización de procesos de interoperabilidad como: “La aplicación de un conjunto de reglas normalizadas para la realización de un proceso que cuente con la habilidad de intercambiar información entre sistemas homólogos”. Dicha definición tiene en cuenta la relación entre los estándares y los procesos de interoperabilidad.

Existen numerosos estándares a nivel internacional, todos marcados por la utilización que se le proporcione. Algunos de los estándares más utilizados a nivel mundial para el intercambio de información son:

- **XML.** “Es un lenguaje de marcas o etiquetas, utilizado en el intercambio de datos. Se propone como un estándar para el intercambio de información entre diferentes plataformas.”[10] Es un lenguaje de etiquetado de importancia en la actualidad, debido a que permite la compatibilidad entre sistemas de manera segura, fiable y fácil. Se puede usar en bases de datos, editores de texto, hojas de cálculo, entre otros.
- **XBRL.** Es el lenguaje de marcas heredero de XML, sirve de vínculo entre diferentes empresas para transmitir información de negocio, es un lenguaje para la comunicación electrónica de datos financieros y de negocio, que actualmente está revolucionando el reporte empresarial en el mundo. Aporta grandes beneficios en la preparación, análisis y comunicación de información de negocio.[11]

Ventajas de la estandarización

- Ayudan a minimizar la duplicación de esfuerzos en la elaboración, ejecución, recolección, procesamiento y distribución de la información o procesos que la manejan.[1]
- Los estándares permiten búsquedas y ejecuciones automatizadas. Cuando se utilizan los estándares, las computadoras se pueden programar permitiendo buscar y encontrar conjuntos de datos útiles.[1]

1.8 Marco de trabajo Sauxe

Los marcos de trabajo son aplicaciones muy utilizadas para la realización de sistemas complejos especialmente aplicaciones web de gestión, ya que automatizan muchos de los patrones de programación para orientarlos a un determinado propósito.[2]

³W3C: Word Wide Web Consortium.

“Un marco de trabajo es una aplicación o conjunto de bibliotecas orientadas a la reutilización a muy gran escala de componentes y software para el desarrollo rápido de aplicaciones”.[2]

El término marco de trabajo tiene una acepción más amplia. Además de incluir una biblioteca de componentes reutilizables, es toda una tecnología o modelo de programación que contiene máquinas virtuales, compiladores, bibliotecas de administración de recursos en tiempo de ejecución y especificaciones de lenguajes.

Sauxe como marco de trabajo además de cumplir con estas características tecnológicas contiene particulares que lo convierten en idóneo para la implementación de aplicaciones web de gestión. Cuenta con una arquitectura que le permiten acelerar el proceso de desarrollo, reutilizar código y promover buenas prácticas como el uso de patrones. Dicha arquitectura compuesta por capas, presenta en su capa superior un modelo vista controlador (MVC), posee un conjunto de componentes que provee la estructura genérica y el comportamiento para una familia de abstracciones, logrando una mayor estandarización, flexibilidad, integración y agilidad en el proceso de desarrollo.[2]

La figura 1 tiene como objetivo exponer algunos de los componentes que integran el marco de trabajo Sauxe. Además los elementos que componen las capas del mismo.

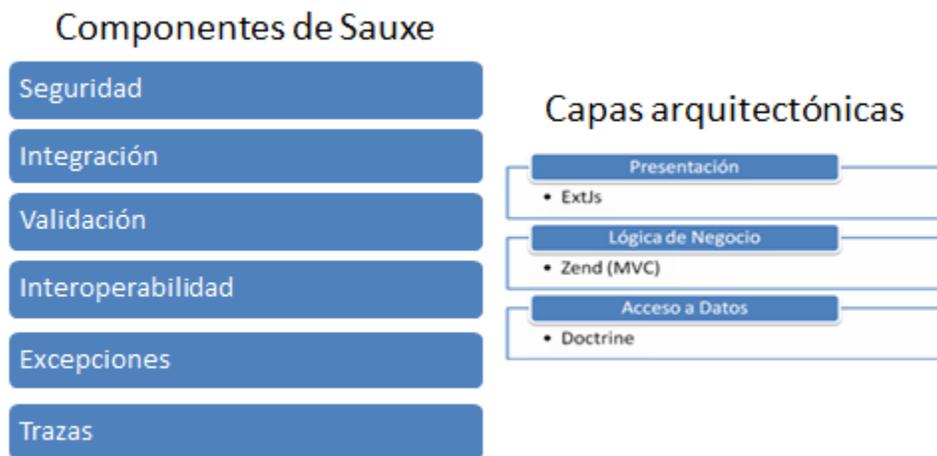


Figura 1: Arquitectura de las capas de Sauxe.

Capa de presentación: en la capa de presentación el marco de trabajo Sauxe utiliza la librería de JavaScript⁴ ExtJs⁵. La misma permite con pocas líneas de código, la realización de interfaces visuales amigables para los usuarios, ya que contiene componentes predefinidos, altamente configurables y fácil de usar. Entre las principales ventajas de dicha librería se encuentran: que es totalmente libre de licencia y compatible con muchos de los navegadores existentes.[2]

Capa de lógica de negocio: en la capa de la lógica de negocio Sauxe cuenta con una extensión de ZendFramework⁶, ZendExt⁷, implementa el patrón arquitectónico MVC. Esto permite la separación de la lógica del negocio, la presentación y el acceso a datos, permitiendo un mayor control y manejo de los elementos perteneciente a cada capa. Esta arquitectura proporciona mayor robustez y agilidad en el proceso de desarrollo de aplicaciones, específicamente aplicaciones web de gestión.[2]

Capa de acceso a datos: en la capa de acceso a datos Sauxe usa un mapeador de objetos relacionales (ORM) denominado Doctrine. El cual posee características que facilitan la realización de tareas complejas, tales como:[12]

- **Reutilización.** Permite invocar métodos de un objeto desde cualquier parte de una aplicación u otra aplicación sin temor a la pérdida de datos del objeto.
- **Encapsulación.** Doctrine encapsula la lógica de los datos, permitiendo realizar cambios que afectan a toda la aplicación únicamente modificando una función.
- **Seguridad.** Implementa mecanismos de seguridad que protegen las aplicaciones de los ataques más comunes, como la inyección de sentencias inválidas.
- **Mantenimiento del código.** Debido al buen ordenamiento de la capa de datos, permite realizar modificaciones sin provocar la pérdida de datos.

Componente de interoperabilidad de Sauxe.

⁴ *JavaScript* es un lenguaje interpretado que permite incluir macros en páginas web.

⁵ ExtJs: Librería de JavaScript utilizada en el uso de interfaces.

⁶ ZendFramework: Marco de trabajo para desarrollo de aplicaciones Web con PHP.

⁷ ZendExt: Extensión de un conjunto de librería de ZendFramework.

Entre los componentes inmersos en el marco de trabajo Sauxe se encuentra el de interoperabilidad, que permite que los sistemas puedan intercambiar información. Es el encargado de proveerle el servicio de interoperabilidad a todas las soluciones que son implementadas sobre dicho marco de trabajo.

Esta solución brinda la posibilidad de configurar los procesos que se desean interoperar, y dentro de los mismos sus atributos y localización. Cada proceso cuenta con un estándar que permite formalizar la estructura en la cual se exportan e importan los datos. Por lo que la principal actividad de la interoperabilidad en el marco de trabajo Sauxe, queda resumida a la exportación e importación de datos, haciendo uso de los estándares de intercambio definidos previamente por las soluciones.

Actualmente el componente de interoperabilidad no cuenta con la capacidad de ejecutar varios procesos de interoperabilidad, debido a que no es capaz de entender las particularidades de cada una de las soluciones implementadas sobre Sauxe.

Este fenómeno conlleva que para cada solución que desee interoperar sea necesaria la implementación de un componente propio, capaz de ajustarse a las particularidades de la misma.

A pesar de que cada proceso contiene características que lo distinguen de otros procesos e incluso comparten una estructura de intercambio de información bien distinta, siempre el proceso de interoperabilidad de cualquier solución va a ser el mismo, la exportación y la importación de la información, por lo que existe la forma de poder estandarizar la ejecución de dicho proceso para todas las soluciones.

Existen otros componentes que están muy relacionados al componente de interoperabilidad y que garantizan algunos de los aspectos a tener en cuenta a la hora del intercambio de datos como son la seguridad, el origen y destino de los datos.

En el caso de la interoperabilidad es importante la seguridad ya que no se podrá exportar e importar información o consumir algún servicio al cual no se tenga acceso. Esto es garantizado con Sauxe, el cual implementa un control de acceso basado en roles donde los privilegios se otorgan según los roles.

Para garantizar la integración Sauxe cuenta con un mecanismo de inversión de control (IoC) en el cual son publicados los servicios de integración de las soluciones implementadas sobre dicho marco de trabajo, facilitando la comunicación entre las mismas.

1.9 Lenguajes de referencia

Si fuese posible utilizar algunos de los lenguajes existentes que describa y ejecute el proceso de interoperabilidad, el componente de interoperabilidad del marco de trabajo Sauxe solo tendría que interpretar dicho lenguaje y de esa forma ejecutar cualquier proceso perteneciente a cualquier solución desarrollada sobre Sauxe.

El lenguaje a utilizar debe cumplir con las características siguientes:

- Debe ser de fácil interpretación.
- Debe contener mecanismos que regulen el flujo de ejecución del proceso.
- Debe ajustarse a la arquitectura de integración y de seguridad que contiene el marco de trabajo.

Análisis del lenguaje XML.

Entre los lenguajes más factibles para la interpretación, condicionado por su estructura y fácil implementación de intérpretes capaces de reconocerlo se encuentra XML. Cuenta con una gran aceptación por parte de entidades reconocidas en el desarrollo de aplicaciones, estas características lo hacen compatible con una gran variedad de estas aplicaciones.[10]

El XML es la solución que propuso la W3C al problema de hacer que la web fuera accesible desde cualquier medio (ordenador, impresora, televisor, teléfono); sin embargo, no tardó en descubrirse que XML era una herramienta muy potente, con posibilidades que iban mucho más allá de la web. De hecho fue pensado para que lo emplearan diseñadores que no tenían por qué tener conocimientos de informática. Donde XML ha demostrado ser más útil, y donde más se está aplicando actualmente, es en el intercambio de información entre aplicaciones.[10]

Como lenguaje de etiquetado extensible, XML resulta de baja complejidad a la hora de interpretarlo, esta característica hace que sea muy utilizado para intercambiar información y a su vez ser compatible con una gran variedad de aplicaciones.

Principales ventajas del lenguaje XML

XML es un lenguaje de intercambio de información abierto. Esta característica hace que muchas compañías en busca de lograr altos niveles de compatibilidad, adopten mecanismos para poder utilizarlo. Entre sus principales ventajas se encuentra:

- 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.[10]

Algunas de las extensiones del lenguaje XML

XML tiene la capacidad de extenderse según la necesidad o las particularidades del problema. Actualmente existen una gran variedad de extensiones del lenguaje XML muy conocidas como **metalenguajes**, basadas en satisfacer necesidades particulares a soluciones específicas. Esto ocurre porque el lenguaje cumple con características que posibilitan que mientras se utilice su sintaxis y se defina en algún documento la semántica de sus etiquetas se pueda utilizar a necesidad del creador. Entre las extensiones del lenguaje XML que más sobresalen se encuentran:[13]

- **XPath:** lenguaje de rutas XML, es un lenguaje para acceder a partes de un documento XML.[13]
También permite construir expresiones que recorren y procesan un documento XML. Posibilita buscar y seleccionar teniendo en cuenta la estructura jerárquica del XML. Fue creado para su uso en el estándar XSLT⁸, para seleccionar y examinar la estructura del documento de entrada de la transformación.
- **XLink:** lenguaje de enlace XML, permite insertar elementos en documentos XML para crear enlaces entre recursos.[10] De esta forma permite crear una relación de vínculos entre varios documentos y describir vínculos a documentos en multitud de ubicaciones.
- **XPointer:** lenguaje de direccionamiento XML, es un lenguaje que permite el acceso a la estructura interna de un documento XML, esto es, a sus elementos, atributos y contenido.[10]
- **XQL:** Lenguaje de Consulta XML, es un lenguaje que facilita la extracción de datos desde documentos XML.[10] Ofrece la posibilidad de realizar consultas flexibles para extraer datos de documentos XML en la Web.
- **XSL:** lenguaje extensible de hojas de estilo, cuyo objetivo principal es mostrar cómo debería estar estructurado el contenido, cómo debería ser diseñado el contenido de origen y cómo debería ser paginado en un medio de presentación.[14]

Ninguna de estas extensiones de XML cumple con las características necesarias para dar solución a la problemática planteada, ya que son para un uso muy delimitado y se alejan mucho de la ejecución de los

⁸ XSLT: Lenguaje extensible de hojas de estilos.

procesos en general. Aunque de ellos se pueden concluir que utilizan formas novedosas para personalizar la arquitectura del lenguaje que extienden.

Análisis del lenguaje BPEL

BPEL, siglas en inglés de (Business Process Execution Language), es un lenguaje de programación destinado para la ejecución de procesos empresariales, derivado de XML y persigue lograr un modelo de programación a gran escala.[15]

Un documento BPEL define el proceso, o la orquestación y la lógica de las acciones que serán ejecutadas por los motores de orquestación. El programa constituye en sí el código fuente de la aplicación que ejecuta el proceso, el motor de orquestación actúa como una máquina virtual capaz de ejecutar código BPEL.[15]

Objetivos del diseño de BPEL. El objetivo principal es definir una serie de conceptos de orquestación de servicios web que pretenden ser usados por vistas internas o externas de un proceso de negocio, aunque también puede utilizarse para otras cuestiones como son[15]:

Proveer sistemas de control jerárquicos y de estilo gráfico, que permitan que su uso sea lo más fusionado e íntegro posible. Esto reduciría la fragmentación del espacio del modelado de procesos.

- Proveer funciones de manipulación simple de datos, requeridas para definir datos de procesos y flujos de control.
- Soportar un método de identificación de instancias de procesos que permita la definición de identificadores de instancias a nivel de mensajes de aplicaciones. Los identificadores de instancias deben ser definidos por socios y pueden cambiar.
- Brindar la posibilidad de la creación y terminación implícitas de instancias de procesos, como un mecanismo básico de ciclo de vida. Operaciones avanzadas de ciclo de vida como por ejemplo "suspender" y "continuar" para mejorar el manejo del ciclo de vida del proceso.

BPEL le permite a los desarrolladores crear programas que automatizan las interacciones entre los servicios web, jugando un papel clave en las arquitecturas orientadas a servicios. Con los niveles de servicios desplegados, BPEL posibilita escribir programas de gran complejidad para lograr la automatización de las actividades de negocio. Esta automatización de la interacción entre los servicios web es comúnmente referida como orquestación de servicios web.

Análisis del lenguaje WS-BPEL.

WS-BPEL es un lenguaje basado en XML que permite especificar el comportamiento de un proceso de negocio basado en interacciones con servicios web. La estructura de un proceso WS-BPEL se divide en cuatro secciones: definición de relaciones con los socios externos, que son el cliente que utiliza el proceso de negocio y los servicios web a los que llama el proceso, definición de las variables que emplea el proceso, definición de los distintos tipos de manejadores que puede utilizar el proceso: manejadores de fallos y de eventos, y descripción del comportamiento del proceso de negocio; esto se logra a través de las actividades que proporciona el lenguaje.[15]

Los principales elementos constructivos de un proceso WS-BPEL son las actividades, que pueden ser de dos tipos: básicas y estructuradas. Las básicas son las que realizan una determinada labor, mientras que las estructuradas pueden contener otras actividades y definen la lógica de negocio. A las actividades pueden asociarse un conjunto de atributos y de contenedores. Estos últimos pueden incluir diferentes elementos, que a su vez pueden tener atributos asociados.[15]

Ventajas y desventajas de lenguajes estudiados.

El lenguaje de ejecución de procesos de negocio BPEL contiene elementos importantes que pueden ser de gran referencia para la implementación de una solución, como los manejadores de eventos. Además la estructura central de un proceso serían las actividades, y pueden clasificarse según el uso que se le vaya a dar a las mismas y a los eventos declarados. También se pueden declarar variables de uso general para cualquier elemento integrante del proceso, pero no cuenta con un mecanismo de configuración visual del proceso ya que no es su objetivo y como se plantea anteriormente, un proceso de interoperabilidad en el marco de trabajo Sauxe puede configurarse a nivel de interfaz, tanto los atributos del proceso como la localización del resultado del mismo. Además no cuenta con un mecanismo para definir dentro de un proceso un estándar de intercambio de datos o una estructura por la cual se medirá el proceso para configurar la salida a la hora de exportar datos y la entrada en tiempo de importación.

El elemento más importante que hace rechazar a BPEL, es que al igual que el WS-BPEL, su ejecución de procesos está basada en la orquestación de los servicios web descritos por algún WSDL⁹ y actualmente el marco de trabajo solo soporta el consumo de servicios mediante el mecanismo de inversión de control.

⁹ WSDL: Lenguaje de Descripción de Servicios Web.

Por otra parte XML como lenguaje de intercambio de datos es muy genérico para utilizarse sin especificar exactamente la sintaxis de un documento y cómo puede utilizarse cada etiqueta declarada en el mismo. Aunque, si se especifica la sintaxis de un documento y el uso asociado a cada elemento o etiqueta definida, es como si se creara un lenguaje de etiquetado propio para solucionar una problemática en particular.

Conclusiones parciales

La consulta y definición de los términos abordados a través de la investigación ofrecen mayor conocimiento de la problemática y de esta forma proponer una solución, teniendo en cuenta la dimensión y características del proceso de interoperabilidad en el marco de trabajo Sauxe. Además el estudio de algunos de los principales lenguajes de ejecución de procesos de negocio permitió conocer sus deficiencias, ventajas e idoneidad frente al problema existente, posibilitando la decisión de extender XML para obtener un lenguaje de ejecución que se ajuste las características deseadas.

CAPITULO 2: PROPUESTA DE SOLUCIÓN

INTRODUCCIÓN

El objetivo del presente capítulo es proponer un metalenguaje de ejecución de procesos de interoperabilidad para los sistemas que se implementan en Sauxe. El lenguaje diseñado no define un estándar de intercambio de datos, sino que estandariza la forma en que los sistemas realizan el proceso de interoperabilidad, lo que hace posible que el componente dinámico de interoperabilidad sea capaz de ejecutar procesos de cualquier solución, sin necesidad de la implementación de funciones capaces de satisfacer particularidades existentes en cada proceso. Para lograrlo, el lenguaje presenta un modelo donde el programa es quien rige el flujo de eventos a realizarse dentro de su ejecución, designándole al usuario el único rol de configuración de eventos. También presenta mecanismos para tratar los flujos alternos a cada evento a realizarse, en caso de no ser posible seguir con la secuencia de ejecución previamente programada. Además define un conjunto de elementos habilitados totalmente en la descripción; tanto a nivel de interfaz como en la manipulación de datos de cada actividad previa al intercambio. Para un mayor entendimiento de la solución se presenta un mapa conceptual que representa todos los elementos asociados a la interoperabilidad en el marco de trabajo Sauxe.

2.1 Mapa conceptual de la solución de interoperabilidad en Sauxe

La construcción del mapa conceptual permite una mejor comprensión del dominio del problema; no es más que una representación visual de los conceptos u objetos del mundo real que son significativos para el problema; representando las clases conceptuales y los componentes de software.[16]

La figura 2 muestra el mapa conceptual que relaciona todos los conceptos y relaciones del proceso de interoperabilidad y se hace una explicación de cada concepto y sus relaciones.

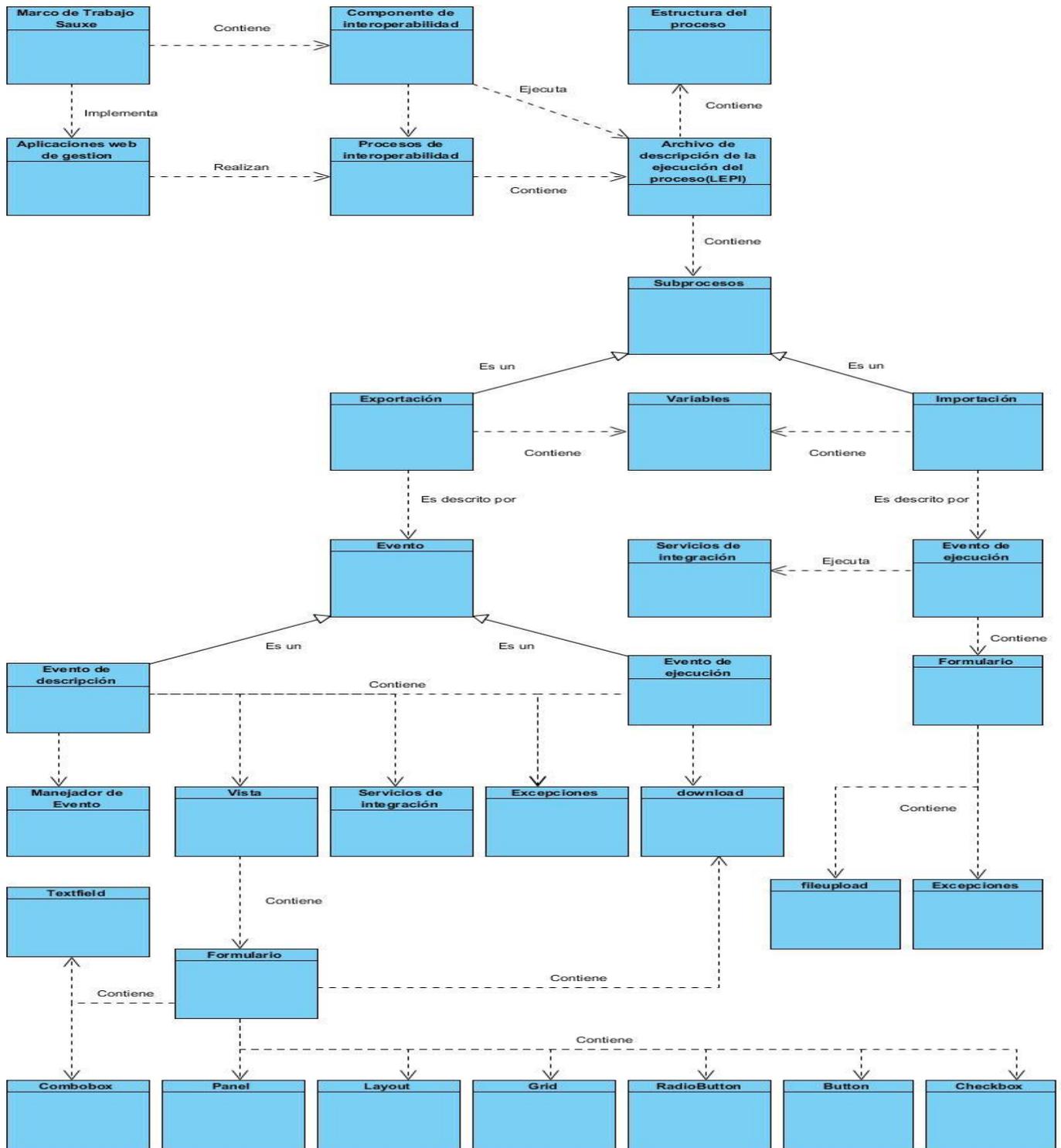


Figura 2: Mapa Conceptual de la solución de interoperabilidad de Sauxe.

En el mapa conceptual propuesto se muestra como primer concepto al marco de trabajo Sauxe, en el cual son implementadas muchas soluciones, especialmente los componentes que integran el Cedrux. Sauxe contiene un componente encargado de proveerles el servicio de interoperabilidad a todas las soluciones que se implementan sobre sí mismo. Para ello cada solución implementada en Sauxe debe definir el modo en que realizará su proceso de interoperabilidad, definir la estructura de sus datos y las configuraciones previas necesarias para la realización del proceso. Dicha estructura de datos tendrá mucha relación con el estándar de intercambio de información por el cual se regirá la solución para la importación y exportación de datos.

El proceso de interoperabilidad se descompone en dos subprocesos relevantes: la exportación y la importación de datos, ambos subprocesos deben ser descritos por el sistema que desea interoperar, pero dentro de un mismo archivo de descripción de procesos y bajo el mismo lenguaje, en este caso el LEPI propuesto. De esta forma el componente podrá ejecutar dinámica y automáticamente todos los procesos de interoperabilidad pertenecientes a las soluciones que se implementan en el marco de trabajo, sin tener que ajustarse a particularidades de cada proceso, ya que cada especificación estará embebida dentro del archivo que contiene la descripción del proceso y descrita en un lenguaje que es perfectamente entendible por el componente.

2.2 Descripción del proceso de interoperabilidad

La interoperabilidad en este ámbito refiere a exportación e importación de datos y su éxito depende principalmente de que se puedan ejecutar ambos subprocesos, es decir un proceso de interoperabilidad se considera completo si es posible exportar e importar datos siguiendo la descripción que plantea, además si le permite al usuario que está ejecutando dichos subprocesos, configurarlos según sus necesidades. Dicha configuración contribuye de manera significativa a la eficiente ejecución de cualquiera de los subprocesos. Un subproceso de interoperabilidad con un alto nivel de configuración, cuenta a su vez con una gran probabilidad de que mediante su ejecución, se logre una exportación o importación eficiente. Este indicador de eficiencia aumenta si además se hace un buen manejo de eventos, alternos a la ejecución de los subprocesos, en busca de minimizar posibles errores por parte de los usuarios. La configuración no solo puede existir a nivel de interfaz sino también a nivel de servicios de integración consumidos durante la ejecución y de estructura de datos del proceso, este último importante en la especificación de los datos por parte del usuario que realmente desee interoperar.

2.3 Subproceso de exportación

Premisas para la exportación: antes de realizar el proceso debe existir el archivo LEPI dentro del sistema que realizará el proceso, ya que define las pautas para llevar a cabo el mismo. Dentro del archivo deben estar especificados los eventos a realizarse para llevar a cabo la operación de exportar.

Resultado del proceso: este subproceso tendrá como resultado un compactado, el cual contendrá los registros con el estado del negocio que se desea exportar. Dicho compactado estará listo para ser transferido por cualquier vía escogida por las partes involucradas.

Beneficios hacia las partes: con la realización del subproceso el componente de interoperabilidad gana mucho a nivel estructural y de reutilización, ya que permite a muchas soluciones poder utilizar un mismo componente, quien atiende cada una de las particularidades de los procesos de interoperabilidad que realizan dichas soluciones. Pero quien más se beneficia es el marco de trabajo Sauxe ya que podrá concebir varios procesos de interoperabilidad en un solo componente sin necesidad de volver a implementarlo, permitiendo a todas las soluciones exportar sus datos y así intercambiar información con sistemas homólogos.

2.3.1 Pasos para exportar datos usando LEPI

➤ **Introducción al componente de interoperabilidad.**

Se deberá acceder al componente de interoperabilidad del marco de trabajo Sauxe.

➤ **Selección del subsistema que desea realizar el proceso.**

Se seleccionará el subsistema que se desea interoperar. Dentro del mismo se debe encontrar el archivo LEPI con la descripción del proceso. Una vez ejecutado comienza el flujo de eventos descritos en el lenguaje.

➤ **Selección de la funcionalidad exportar.**

Dentro de la interfaz principal del componente de interoperabilidad el usuario deberá seleccionar la opción exportar.

➤ **Configuración del subproceso de exportación.**

Una vez seleccionado el sistema que realizará el proceso y la funcionalidad exportar el usuario puede o no introducirse a la configuración del proceso que se realiza. Esto está en dependencia de como haya sido

descrita la ejecución del proceso en ejecución. Puede existir la ocasión en la cual la ejecución del subproceso de exportación se encuentre condicionada a la introducción o elección de datos por parte del usuario. Dichas configuraciones del subproceso serán actividades o eventos previos a la ejecución de los mismos y tendrán mucha relación con las salidas.

➤ **Especificación de la dirección local donde se desea guardar el resultado del proceso.**

Completado todo el proceso de interoperabilidad se procederá al archivo de los datos gestionado durante toda la operación, para ello es necesario contar con una dirección dentro de la máquina en la cual se esté ejecutando el proceso.

2.4 Subproceso de importación

➤ **Premisas para la importación.**

Debe existir el archivo LEPI dentro del sistema que desea realizar el proceso. Dentro del archivo debe estar descrito cómo se realizará el mismo. Teniendo en cuenta que la importación es estrictamente el proceso inverso de la exportación, en la descripción del proceso solo se debe especificar una forma para acceder a una dirección específica donde se encuentra el o los ficheros XML productos de la exportación.

➤ **Resultado del proceso.**

Este subproceso tendrá como resultado la ejecución de un servicio que posibilita archivar en base de datos los registros con el estado de negocio importado.

➤ **Beneficios hacia las partes.**

Este subproceso es muy importante ya que es la complementación del subproceso de exportación, permitiendo obtener la información que en cierto momento fue exportada. Debido la definición de estándar de comunicación entre sistemas este subproceso permite la importación de datos de otros sistemas siempre y cuando cumplan con dicho estándar. Además varias soluciones desarrolladas sobre el marco de trabajo Sauxe podrán utilizar el componente dinámico para importar datos, utilizando la solución propuesta para la ejecución de su proceso en general.

2.4.1 Pasos para importar datos usando el LEPI

La importación contará con los dos primeros pasos del subproceso de exportación. Luego el usuario procederá de la siguiente manera:

➤ **Selección de la funcionalidad importar.**

Ya seleccionado el subsistema, el usuario procederá a realizar el subproceso de importación. Los datos a importar deben estar ubicados en una dirección física a la cual el usuario pueda acceder. Esta operación internamente consultaría la descripción de la ejecución del proceso, exactamente el subproceso importar y la ejecutaría completándose de esta forma todo el proceso de interoperabilidad.

➤ **Selección de la dirección donde se encuentra el fichero a importar.**

En este paso se procederá a seleccionar los archivos a importar. Una vez importados los datos, el componente de interoperabilidad le brinda un servicio a la solución que realiza el proceso, el cual contiene los datos importados listos para guardar en su base de datos o para utilizarlos según convenga.

2.5 Requisitos del LEPI

A continuación se muestran los requisitos fundamentales que servirán de base para la creación del lenguaje de ejecución de proceso de interoperabilidad. Dichos requisitos fueron producto de un estudio de todos los sistemas que pueden realizar el proceso actualmente en el marco de trabajo Sauxe y principalmente los dos procesos que utilizan el componente de interoperabilidad existente. Dichas soluciones a pesar de contener rasgos comunes en cuanto al negocio refiere, se encuentran ejecutadas en componentes independientes.

➤ **Describe exactamente una entrada y una salida.**

El proceso de interoperabilidad en el marco de trabajo Sauxe tendrá como entrada un fichero LEPI, el mismo contendrá la descripción completa del proceso de interoperabilidad utilizando la sintaxis gramatical y la estructura definida para la utilización del lenguaje. Como salida obtendrá los datos estandarizados y listos para ser enviados o procesados según la operación o el subproceso de interoperabilidad ejecutado.

➤ **Capacidad de describir la estructura de cualquier solución.**

El lenguaje debe ser capaz de describir un flujo de acciones que define el programador de un sistema desarrollado sobre Sauxe, con la necesidad de consumir el servicio de interoperabilidad que provee dicho marco de trabajo. A su vez debe ser de fácil ejecución para el componente dinámico de interoperabilidad.

➤ **Define el flujo de eventos según el proceso y el subproceso a realizar.**

Permitirá definir mediante los eventos, el flujo de actividades a seguir para llevar a cabo el proceso de interoperabilidad y directamente la exportación o importación de datos.

➤ **Genera las diferentes vistas y componentes visuales según el flujo de evento.**

Cada actividad ya sea de configuración del proceso, como de ejecución deberá contar con interfaces visuales que hagan posibles dichas acciones. Esta característica está muy ligada a la capacidad que tienen los procesos de interoperabilidad de Sauxe de configuración a nivel visual tanto la ejecución como el resultado de dicho proceso.

➤ **Modela las relaciones de dependencia ejecutiva entre los flujos de eventos y sus componentes visuales.**

Se presenta un modelo donde al igual que WS-BPEL, las actividades serán descritas por eventos y cada evento realizado dependerá de la realización del anterior, que una vez terminada su ejecución lo invocará en caso de la no ocurrencia de un evento alterno. De ahí se presenta la necesidad de contar con algún mecanismo, que regule la ocurrencia de los eventos que describen las actividades a ejecutarse para la concepción del proceso de interoperabilidad.

➤ **Define una vía para la representación de los servicios a utilizar en cada flujo de eventos.**

Cada actividad utilizada para describir todo el proceso de interoperabilidad, puede consumir servicios de integración, necesarios para acceder a los datos vinculados al proceso. El lenguaje debe ser capaz de describir dichos servicios, pues son la única vía de acceso a la información por parte del componente dinámico con respecto al sistema que realiza el proceso de interoperabilidad.

2.6 Reglas sintácticas del LEPI

La medida en que el lenguaje extiende de XML, hace necesario que al mismo tiempo se vea forzado a cumplir sintácticamente las reglas definidas por la World Wide Web Consortium (W3C) para el diseño específico de etiquetas, independientemente del uso semántico que se le vaya a dar a las mismas. Eso implica que todas las extensiones de XML deberán cumplir con las reglas sintácticas definidas por la W3C, además de definir el significado y la semántica que tendrá cada elemento definido.

Las reglas de etiquetado definidas por XML y que cumple el lenguaje propuesto son:[10]

El documento sólo puede contener un elemento raíz: el elemento raíz de un documento es único y agrupa todo el contenido que se considera parte del propio documento. El elemento raíz es el primer elemento que aparece después de la sección de prólogo del documento. También se conoce al elemento raíz como el elemento de documento.

Todos los elementos deben contener etiquetas finales: todos los elementos de un documento deben tener una etiqueta final.

Los nombres de las etiquetas iniciales y finales del elemento deben ser idénticos: el nombre de una etiqueta final debe coincidir exactamente con el nombre de su etiqueta inicial correspondiente.

Los elementos no pueden superponerse: si la etiqueta inicial de un elemento aparece dentro de otro elemento, debe finalizar dentro del mismo elemento contenedor.

Todos los valores de atributo deben utilizar comillas: los valores de atributo deben estar entre comillas, ya sean sencillas o dobles.

Los caracteres <, >, & no se pueden utilizar dentro del texto de un documento: tienen un significado concreto para los analizadores del lenguaje.

El cumplimiento de estas reglas garantiza que el documento está bien formado, es decir, que utiliza la sintaxis establecida y recomendada por la W3C. Un documento bien formado no necesariamente puede ser válido, la validez está muy asociada al cumplimiento de las reglas especificadas en la definición de tipo de documento (DTD) por la que se debe regir. Los documentos XML que se ajustan a su DTD se denominan válidos pues define los tipos de elementos, atributos y entidades permitidas, además de representar todas las limitaciones para su uso.

2.7 Estructura del LEPI

El LEPI basado en las sintaxis de XML, se ajusta a una estructura según la necesidad de los procesos que describirá, teniendo en cuenta las reglas de etiquetado definidas por W3C para documentos XML. La estructura del lenguaje se basa en un proceso que presenta dos subprocesos únicos y fundamentales, cada subproceso (Exportar e Importar) presenta elementos dentro de sí que lo describen y lo ejecutan en dependencia de cuál de ellos se desee ejecutar. Además un proceso de interoperabilidad también presenta una estructura de datos asociada que no es más que la estructura de contenido que tendrán los datos que se intercambiarán. Cada subprocesos tendrá como base para su desempeño la ejecución de eventos programados que describirán cada actividad previa a realizar. Defínase como una actividad, una acción a realizar para complementar la ejecución del proceso de interoperabilidad en general. Dichas actividades generalmente son de configuración del proceso de interoperabilidad. Ver figura 3



Figura 3: Estructura del LEPI.

2.8 DTD asociado al LEPI

Crear una DTD equivale a crear un lenguaje derivado de XML para una aplicación específica. La definición de tipo de documento puede existir en un fichero externo al documento que se desea validar y puede ser compartido para varios documentos que deseen ser validados por dicha definición de tipos de documentos.[17]

Cada regla definida específicamente para el uso de cada etiqueta definida en el lenguaje debe contemplarse en la DTD, ya que la misma valida formalmente los elementos utilizados en la descripción de procesos usando el LEPI.

La primera regla a tener en cuenta a la hora de describir un proceso, es que el lenguaje propuesto a pesar de extender de XML y ser un metalenguaje muy utilizado para el intercambio de datos; no está diseñado con ese fin pues su estructura no lo permite. El lenguaje de ejecución de proceso de interoperabilidad describe los procesos realizados para lograr una interoperabilidad entre sistemas homólogos y además contiene elementos que hacen posible la ejecución lineal de dichos procesos.

Una característica muy importante en tiempo de ejecución sería la ejecución de todos los elementos según el subproceso a que pertenezcan de forma descendente, haciendo uso de los manejadores de eventos definidos.

Un proceso de interoperabilidad debe contener los subproceso de exportación e importación, además de un atributo nombre que podrá asignársele cualquier combinación de caracteres, también señala la denominación del proceso general, y su existencia es obligatoria. Es decir para que la declaración de una etiqueta proceso pueda ser válida tiene que contener las etiquetas de declaración de los dos subprocesos y un nombre.

```
<!ELEMENT process (export, import, structure)>
<!ATTLIST process
  name CDATA #REQUIRED
>
```

Figura 4: Declaración del elemento proceso.

La declaración de la etiqueta que describe el subproceso de exportación. Es única y obligatoria en el documento, podrá contener varias o ninguna etiqueta eventos de descripción y una sola que describe el evento de ejecución. Además podrá contener variables para el manejo de datos y una o varias estructuras que son las que representan la forma en que serán exportados los mismos. Incluye un nombre que lo identifica y dicho nombre podrá ser la combinación de cualquier combinación de caracteres válidos.

```
<!ELEMENT export (event_description*, event_execution, variable+)>
<!ATTLIST export
  name CDATA #REQUIRED
>
```

Figura 5: Declaración del elemento exportar.

Los eventos de descripción podrán contener un elemento vista, al menos uno o varias etiquetas servicios, un manejador de eventos y una etiqueta excepción para el control de flujos alternos asociados a cada evento. Además deberá contar con los atributos nombre e id.

```
<!ELEMENT event_description (view, service+, event_handler, exception)>
<!ATTLIST event_description
  name CDATA #REQUIRED
  id ID #REQUIRED
>
```

Figura 6: Declaración del elemento evento de descripción.

El elemento vista tiene que contener al menos un formulario, y entre sus atributos se encontrará el nombre de la vista, el mismo puede ser cualquier combinación de caracteres válidos.

```
<!ELEMENT view (form+)>
<!ATTLIST view
  name CDATA #REQUIRED
>
```

Figura 7: Declaración del elemento vista.

Los elementos grid, textfield, checkbox, radiobutton, tree, comboBox, y radioGroups son muy similares en cuanto a funcionamiento. Se utilizan fundamentalmente en la introducción y selección de datos por parte del usuario. Dichos datos pueden ser gestionados mediante servicios de integración, variables creadas con valores predeterminados o estructuras implementadas en el mismo archivo de descripción de la ejecución del proceso. Hay que tener en cuenta que estos elementos son diseñados para manejar datos de tipo unitarios (cadenas de caracteres, números reales) o más complejos como objetos. En el caso de los grid pueden llegar a contener otros elementos.

Una etiqueta formulario podrá contener los elementos grid, textfield, checkbox, radiobutton, tree, comboBox, layouts, radioGroups varias veces o puede no contenerlos. También incluye dos botones, uno para poder resetear de forma completa los datos accionados en los elementos de formulario, y otro de tipo enviar, el cual asignará a las variables creadas los datos seleccionados en los elementos contenidos dentro del formulario.

```

<ELEMENT form (Grid*, Textfield*, Checkbox*, Radiobutton*, Tree*, ComboBox*,
Layout*, RadioGroups*, reset_button?, submit_button)>
<!ATTLIST form
  id ID #REQUIRED
  name CDATA #REQUIRED
>

```

Figura 8: Declaración del elemento formulario.

Los elementos botones pueden ser de dos tipos y son especificados a través de un nombre que incluye cualquier combinación de caracteres válidos. Además los botones de tipo enviar serán los encargados de ejecutar la acción de asignación de los valores de los elementos seleccionados dentro del formulario a sus respectivas variables. Opcionalmente en dependencia del tipo de evento, mediante el atributo de referencia, podrían también invocar al manejador de eventos para proceder con la ejecución del próximo evento.

```

<ELEMENT reset_button EMPTY>
<!ATTLIST reset_button
  name CDATA #REQUIRED
>
<ELEMENT submit_button EMPTY>
<!ATTLIST submit_button
  reference IDREF #IMPLIED
  name CDATA #REQUIRED
>

```

Figura 9: Declaración de los botones.

Este elemento será el encargado de describir los servicios que se invocarán en el proceso. Un elemento servicio será contenedor de elementos parámetros que serían los parámetros necesarios para su ejecución en caso de necesitarlos. Como atributos de dicha etiquetas se encontrará el nombre del servicio y nombre del componente que lo provee, podrán ser cualquier combinación de caracteres válidos. Tendrá un atributo id para el caso de necesitar guardar el valor resultante en una variable esta pueda referenciarlo a través de su id.

```

<ELEMENT service (parameter)*>
<!ATTLIST service
  name CDATA #REQUIRED
  id ID #REQUIRED
  subsystem CDATA #REQUIRED
>

```

Figura 10: Declaración del elemento servicio.

La etiqueta parámetro tendrá un atributo de tipo referencia que en caso de no ser vacío hará referencia al id de una variable. Dicha variable contiene el valor necesario para la ejecución del servicio. También contará con un nombre específico que se le dominará al parámetro en el servicio, dicho nombre será cualquier combinación de caracteres válidos.

```
<ELEMENT parameter EMPTY>
<!ATTLIST parameter
  reference IDREF #IMPLIED
  name CDATA #IMPLIED
>
```

Figura 11: Declaración del elemento parámetro.

Los manejadores de eventos serán las etiquetas que le darán el curso a la ejecución del proceso ya que se encargarán de invocar al siguiente evento que contenga como id su referencia.

```
<ELEMENT event_handler EMPTY>
<!ATTLIST event_handler
  reference IDREF #REQUIRED
>
```

Figura 12: Declaración del elemento manejador de evento.

Las excepciones controlarán el flujo alterno de la ejecución de los eventos. El código de la excepción especificará la excepción a capturar ya que solo podrá capturarse una sola, aunque existan varios flujos alternos o escenarios imprevistos en la ejecución.

```
<ELEMENT exepcion EMPTY>
<!ATTLIST exepcion
  type CDATA #REQUIRED
  code NMTOKEN #REQUIRED
>
```

Figura 13: Declaración del elemento Excepción.

Una vez que se ejecute un evento y se pase a invocar a otro, resulta importante guardar en variables los datos contenidos en los componentes visuales o los servicios invocados desde el evento ejecutado. Esto ocurre por si fuese necesario el trabajo con datos desde otro evento. Es por ello que pueden declararse y

utilizarse las etiquetas variables con el fin de un uso posterior. Las mismas presentan atributos como son el id la referencia y el valor, este último puede ser desde un entero hasta uno o varios objetos.

```
<ELEMENT variable EMPTY>
<ATTLIST variable
  id ID #REQUIRED
  reference IDREF #IMPLIED
  values CDATA #REQUIRED
>
```

Figura 14: Declaración del elemento variable.

Etiqueta contenida en los formularios y puede ser utilizada para el manejo de grandes volúmenes de datos visualmente. Su funcionamiento es muy parecido al de la librería ExtJs. Dicha etiqueta puede mostrar y permitir trabajar con datos preferentemente con el mismo formato. Además puede contener otros elementos visuales dentro de sí misma. Para su mejor funcionamiento posee una etiqueta denominada store la cual referencia a una variable que contiene los datos que serán cargados dentro del elemento grid. Además podrá contener al menos una o varias columnas en la cual se situarán ordenadamente los datos cargados por la etiqueta store.

Como atributos la etiqueta Grid tendrá un puntero de tipo referencia que referenciaría el id de la variable donde guarda la selección o configuración de los datos dentro de sí mismo por parte del usuario, este atributo deberá ser de existencia obligatoria. Otro atributo contenido por el elemento grid será el valor, este atributo almacena la información que brinda el elemento a respuesta de la acción del usuario.

```
<ELEMENT Grid (store, column+)>
<ATTLIST Grid
  pointer ID #REQUIRED
  values CDATA #REQUIRED
  name CDATA #REQUIRED
>
<ELEMENT store EMPTY>
<ATTLIST store
  reference IDREF #REQUIRED
>
<ELEMENT column EMPTY>
<ATTLIST column
  name CDATA #REQUIRED
>
```

Figura 15: Declaración del elemento Grid.

Este elemento solo estará presente en los formularios de las vistas de los eventos de ejecución, dentro del subproceso de exportación, ya que el último paso para exportar de cualquier solución es la descarga de los datos a una dirección específica. Datos que luego serán transmitidos mediante cualquier vía o forma de intercambio de datos.

Dicho elemento o etiqueta obligatoriamente contiene los atributos nombre y dirección. En el caso de la referencia se utilizará para referenciar la variable que tiene los datos a guardar y la dirección contendrá la dirección señalada por el usuario en el mismo elemento download, dirección que será el destino de dichos datos.

```
<!ELEMENT download EMPTY>  
<!ATTLIST download  
  reference IDREF #REQUIRED  
  id ID #REQUIRED  
  name CDATA #REQUIRED  
>
```

Figura 16: Declaración del elemento descargar.

Este elemento solo estará inmerso en la vista de los eventos asociados al subproceso de importación. A través de este elemento que requiere una dirección introducida por el usuario que se encuentre ejecutando el proceso se procederá a cargar los archivos.

```
<!ELEMENT Fileupload EMPTY>  
<!ATTLIST Fileupload  
  pointer IDREF #REQUIRED  
  reference IDREF #REQUIRED  
  dir CDATA #REQUIRED  
  values CDATA #REQUIRED  
  name CDATA #REQUIRED  
>
```

Figura 17: Declaración del elemento subir archivo.

Este elemento es uno de los dos subprocesos que contiene un proceso de ejecución y su función es la realización del proceso inverso de exportación. En este caso se cargarán los datos exportados anteriormente de algún sistema homólogo.

El componente de interoperabilidad deberá proveer un servicio en el cual se envía el flujo de datos extraídos de los archivos importados. Para ello es necesario que el proceso cuente con un elemento especial de las vistas diseñado solo para este caso y es el fileupload.

```
<!ELEMENT import (Fileupload)>
<!ATTLIST import
  name CDATA #REQUIRED
>
```

Figura 18: Declaración del elemento importar.

Este elemento o etiqueta será utilizada para la organización correcta de los elementos integrantes de una vista. Dentro de esta etiqueta podrán estar los componentes grid, textfield, checkbox, radiobutton, tree, comboBox y radioGroups. Permite según su tipo, cambiar la forma en que aparecen los componentes dentro de la vista. Como atributo contará con un nombre que puede ser cualquier combinación de caracteres válidos y es obligatorio, además cuenta con un atributo tipo que puede ser border, card, fit o window.

```
<!ELEMENT Layout (Grid*, Textfield*, Checkbox*, Radiobutton*, Tree*, ComboBox*,
RadioGroups*)>
<!ATTLIST Layout
  name CDATA #REQUIRED
  type (border | window | fit | card) #REQUIRED
>
```

Figura 19: Declaración del elemento marco.

Los contenedores Panel, admiten diversos layouts, es decir, configuraciones visuales para incluir los distintos elementos como grid, textfield, checkbox, radiobutton, tree, comboBox y radioGroups aunque también cada uno de los elementos mencionados puede aparecer de forma independiente es decir no necesariamente en un Layout dentro del Panel. El atributo será un nombre que especifique al panel.

```
<!ELEMENT Panel (Grid*, Textfield*, Checkbox*, Layout, Radiobutton*, Tree*,
ComboBox*, RadioGroups*)>
<!ATTLIST Panel
  name CDATA #REQUIRED
>
```

Figura 20: Declaración del elemento Panel.

Como bien se muestra en el modelo conceptual un evento de ejecución presenta las mismas características que uno de descripción, excepto en las vistas que adiciona un componente llamado fileupload, elemento en el cual se especificará la dirección física a guardar los datos o el resultado del proceso general.

```
<!ELEMENT event_execution (view, structure*, service+, event_handler, exception)>
<!ATTLIST event_execution
  name CDATA #REQUIRED
  id ID #REQUIRED
>
```

Figura 21: Declaración del elemento vista.

Una estructura en el marco de la investigación se refiere a la arquitectura de los datos con que contará cada proceso de interoperabilidad. Cada subproceso de exportación debe contar con una estructura de datos. De esta forma se asegura que los archivos exportados bajo el estándar XML estén organizados y sea válido. La estructura además será contenedora de un atributo hijo, que este a su vez podrá contener otras estructuras dentro y así recursivamente hasta que se defina una estructura sin hijos, solo con atributos.

```
<!ELEMENT structure (attribute+, child*)>
<!ATTLIST structure
  id ID #REQUIRED
  name CDATA #REQUIRED
>
<!ELEMENT attribute EMPTY>
<!ATTLIST attribute
  name CDATA #REQUIRED
  value CDATA #REQUIRED
>
<!ELEMENT child (structure)*>
```

Figura 22: Declaración del elemento estructura.

2.9 Simbología del LEPI

Se definieron una serie de etiquetas, de las cuales son de suma importancia, el conocimiento de sus funciones dentro del lenguaje así como su significado. Las mismas fueron definidas bajo las reglas sintácticas definidas por la W3C para etiquetado de documentos XML. Muy lejos de contener datos, están

diseñadas con el fin de describir tanto desde nivel de secuencia en la ejecución de los procesos hasta la forma visual en que se muestra cada elemento contenido en el proceso de interoperabilidad. En este epígrafe se muestra la sintaxis de cada elemento, y como están exactamente escrito todos los elementos dentro del documento de descripción de la ejecución de un proceso. Además se especifica el uso de cada elemento y el significado de cada atributo contenido.

Etiquetas terminales

➤ `<elemento atributo 1="valor1" atributo 2="valor2" atributo n="valor n" />`

No serán contenedoras de otras etiquetas y al igual que todas las etiquetas definidas en el lenguaje, será obligatorio su análisis en tiempo de ejecución por parte del componente de interoperabilidad del marco de trabajo Sauxe.

Etiquetas contenedoras

➤ `<elemento atributo 1="valor1" atributo 2="valor2" atributo n="valor n"> </elemento>`.

Esta etiqueta podrá contener al menos una etiqueta de tipo terminales y varias contenedoras.

Debido a que la arquitectura de la capa de presentación del marco de trabajo actualmente se gestiona mediante la librería de JavaScript ExtJs, la misma permite que con pocas líneas de código sea posible realizar interfaces amigables para los usuarios. Es una de las más avanzadas librerías para el desarrollo rápido de aplicaciones con una apariencia totalmente novedosa y una arquitectura flexible. Se hizo necesario y pensando en la utilización del lenguaje por el componente de interoperabilidad, nombrar elementos pertenecientes a las vistas, de la misma manera que se hace en el marco de trabajo utilizando la librería de JavaScript. Además de esto para mantener una uniformidad en cuanto a lenguaje refiere se hizo necesario adoptar el idioma con la cual se nombra los elementos de ExtJs. Esto trae como ventaja que el componente de interoperabilidad asocie dichos elementos del lenguaje con los componentes visuales que le brinda la librería.

La siguiente tabla muestra en que funciones pueden ser utilizados los elementos del lenguaje y que relación puede existir entre los mismos, además del significado de cada atributo perteneciente al elemento.

Elemento	Sintaxis	Especificación	Atributos
Proceso	<pre><process> </process></pre>	Etiqueta que define todo el proceso de interoperabilidad. Su declaración dentro del lenguaje es obligatoria. La ejecución de todos sus elementos dará lugar a que sea posible la interoperabilidad tanto para exportar como para importar datos.	name: especifica el nombre del proceso. Y podrá contener cualquier combinación de caracteres válidos respecto al tipo de datos de dicho atributo.
Subproceso de Exportación	<pre><export> </export></pre>	Dentro de esta etiqueta se describirá todo el subproceso de exportación, debido a que dentro de la misma se encontrarán todos los elementos utilizados en la descripción de dicho subproceso.	name: cada subproceso deberá contar con un nombre. En este caso a pesar de que la existencia del elemento exportar es única es necesario que contenga nombre para identificarlo y mostrar dicho nombre en la interfaz de configuración del subproceso.
Eventos de descripción	<pre><event_description> </event_description></pre>	Base del proceso de ejecución. Cada evento describe un escenario específico del proceso de interoperabilidad y a su vez la ejecución	id: identificador único en todo el LEPI. name: señala el nombre del evento dentro LEPI y de ahí también el nombre de la

		continúa de todos los eventos, describe el proceso de interoperabilidad de forma genérica.	actividad que describe dicho evento.
Eventos de ejecución	<pre><event_execution> </event_execution></pre>	Este elemento solo podrá aparecer una vez en todo LEPI y al igual que el evento descripción contará con elementos necesarios para describir un escenario perteneciente a un proceso de interoperabilidad, además contará con la capacidad de describir y ejecutar un escenario final.	id: identificador único en todo el LEPI. name: señala el nombre del evento dentro LEPI y el nombre de la actividad que describe dicho evento.
Vista	<pre><view> </view></pre>	Elemento en el cual se especifican las características visuales con las que contará el evento que serán las mismas de la actividad que describe dicho evento.	name: atributo que señala el nombre de la vista que describe el evento.
Formulario	<pre><form></pre>	Contendrá todos los componentes visuales	id: identificador único en todo

	<code></form></code>	necesarios que a su vez contendrá la vista y será capaz de cargarlo con los respectivos datos obtenidos de los servicios que contiene el evento donde se ubica.	el LEPI. name: señala el nombre formulario dentro de la vista ya que la misma puede contener varios.
Grid	<code><grid></code> <code></grid></code>	Componente visual muy similar en funciones al de la librería de JQuery ExtJs. Puede contener otros componentes, es muy útil para organizar Componentes agrupados dentro de una misma vista.	values: este atributo guarda los datos seleccionado o introducidos por el usuario. name: nombre que recibirá el elemento dentro de la vista. pointer: su valor referencia al Id de la variable donde se guardarán los datos.
Store	<code><store/></code>	Este elemento contendrá los datos a mostrar en el elemento grid. Es decisivo en la creación dinámica del grid y está muy relacionado con las columnas ya que se mostrarán los datos en las columnas definidas en el grid.	reference: este atributo le señalará al componente de donde deberá cargar los datos es decir el id de la variable que contiene los valores a mostrar.

<p>Columnas</p>	<p><column/></p>	<p>Mediante este elemento se puede señalar la configuración en cuanto a nombre y secuencia de las columnas donde se mostrarán los datos. Muy importante que los nombres de las columnas se relacionen con los datos que van a estar contenido en ellas.</p>	<p>name: cualquier combinación de caracteres válidos. Nombre que sugiere el contenido de las columnas.</p>
<p>Botón enviar</p>	<p><submit_button/></p>	<p>Este elemento se puede utilizar en cada formulario para una vez seleccionados o introducidos los datos por el usuario hacer las asignaciones de dichos datos a sus respectivas variables y en caso de haberse completado el evento referenciar al manejador de eventos para dar curso a la ejecución del proceso.</p>	<p>reference: este atributo puede o no estar presente y es la forma de especificar el id del manejador de eventos en caso de haberse configurado o consultados los datos en el evento en ejecución.</p> <p>name: el texto que visualmente mostrará el elemento.</p>
<p>Botón resetear</p>	<p><reset_button/></p>	<p>Se puede utilizar para cambiar algunos de los datos seleccionados o</p>	<p>name: el texto que visualmente mostrará el elemento.</p>

		introducidos en el formulario.	
ComboBox	<ComboBox/>	Muy utilizable cuando la información que manejamos está agrupada en categorías, ya que permite representarla en un solo componente de forma aislada.	<p>Id: identifica al elemento dentro de la vista y muy útil para acceder al valor del elemento dentro del evento.</p> <p>reference: forma de crear hipervínculos y tener acceso a otros elementos de dentro y fuera del mismo documento.</p> <p>name: nombre que recibirá el elemento dentro de la vista.</p> <p>values: se guardará el valor de la acción del usuario sobre el componente.</p> <p>pointer: id de la variable donde se guardarán los datos.</p>
Layout	<layout> </layout>	Permite agrupar varios componentes visuales dentro de un formulario.	<p>type: este atributo brinda la posibilidad de configurar aún más la forma en que el usuario interacciona con los datos. Puede adoptar los valores (border, card, fit o window) aunque sino se especifica su valor por defecto será window por ser más viable su configuración y</p>

			<p>ejecución.</p> <p>name: nombre que recibirá el elemento dentro de la vista.</p>
RadioGroups	<RadioGroups/>	<p>Elemento utilizado para contener varios valores preferentemente de un mismo objeto y aun así poder utilizar uno solo.</p>	<p>id: identifica al elemento dentro de la vista y muy útil para acceder al valor del elemento dentro del evento.</p> <p>reference: forma de crear hipervínculos y tener acceso a otros elementos de dentro y fuera del mismo documento.</p> <p>name: nombre que recibirá el elemento dentro de la vista.</p> <p>values: se guardará el valor de la acción del usuario sobre el componente.</p> <p>pointer: id de la variable donde se guardarán los datos.</p>
Panel	<panel> </panel>	<p>Este elemento puede contener varios componentes visuales también. Se puede usar para dar una mayor</p>	<p>id: identifica al elemento dentro de la vista.</p> <p>reference: forma de crear hipervínculos y tener acceso</p>

		organización a los mismos.	a otros elementos de dentro y fuera del mismo documento. name: nombre que recibirá el elemento dentro de la vista.
Download	<download/>	Este elemento solo se encuentra en las vistas de los eventos de ejecución ya que en él se almacena la dirección donde se guardará los archivos XML productos de todo el proceso.	id: identifica al elemento dentro de la vista. reference: forma de crear hipervínculos y tener acceso a otros elementos de dentro y fuera del mismo documento. name: nombre que recibirá el elemento dentro de la vista.
Campo de texto	<textfield/>	Elemento utilizado para almacenar y visualizar texto mediante dos vías, introducido por el usuario manualmente o de forma automática mediante algún servicio de acceso a datos.	reference: forma de crear hipervínculos y tener acceso a otros elementos de dentro y fuera del mismo documento. name: nombre que recibirá el elemento dentro de la vista.
Casilla de verificación	<Checkbox/>	Elemento de verificación. Muy utilizado para referenciar características de tipo booleanas, es decir características de tipo	reference: forma de crear hipervínculos y tener acceso a otros elementos de dentro y fuera del mismo documento. name: nombre que recibirá el elemento dentro de la vista.

		verdadera o falsa.	
Radiobutton	<Radiobutton/>	Este elemento le permitirá al usuario seleccionar una única opción entre un grupo de elecciones.	reference : forma de crear hipervínculos y tener acceso a otros elementos de dentro y fuera del mismo documento. name : nombre que recibirá el elemento dentro de la vista.
Árbol	<Tree/>	Muy útil en el trabajo visual con elementos relacionados de forma jerárquica o anidados, similar a la estructura de árbol de otros lenguajes.	reference : forma de crear hipervínculos y tener acceso a otros elementos de dentro y fuera del mismo documento. name : nombre que recibirá el elemento dentro de la vista.
Servicios	<service> </service>	Esta etiqueta es de suma importancia en el lenguaje ya que define la manera de gestionar los datos. Como no se permitirán realizar operaciones en el proceso es muy necesario que cada dato que se gestione provenga de un servicio de integración previamente declarado	subsystem : es necesario contar con el nombre del subsistema que brinda el servicio ya que la sintaxis en la cual se invoca lo requiere. name : puede ser cualquier combinación de caracteres válidos y debe coincidir con el nombre indicado por quien lo provee. Id : mediante el atributo id se podrán guardar los datos en las variables declaradas en

		por el sistema que realiza el proceso.	función de utilizar este valor en otros elementos.
Parámetros	<code><parameter/></code>	Mediante esta etiqueta se podrá consumir servicios que requieran parámetros. La definición de varias de esta etiqueta podrán simular todos los parámetros necesarios por un servicio para su ejecución.	reference: la referencia en esta etiqueta indicará el id de la variable donde está guardado el valor que debe tomar el parámetro y además debe coincidir el tipo de datos propuesto en el cuerpo del servicio. name: cualquier combinación de caracteres válidos que identifique el nombre del parámetro especificado por el servicio.
Manejador de eventos	<code><event_handler/></code>	Mediante este elemento se podrá dar una secuencia a la ejecución del proceso	reference: el atributo referencia nos señala el id del evento que le sigue en la secuencia de la ejecución.

		<p>de interoperabilidad. Cada evento contará con un elemento de este tipo de donde se podrán invocar a otros eventos una vez terminada la ejecución del evento donde se encuentra.</p>	
<p>Excepciones</p>	<p><code><exepcion/></code></p>	<p>La etiqueta excepciones ayuda al control de los flujos alternos a cada evento, en caso de la ocurrencia de algún evento adverso a la ejecución normal del ciclo propuesto en la estructura del lenguaje las excepciones son las encargadas de lanzar los mensajes de acorde con el error ocurrido. El marco de trabajo contiene un mecanismo de tratamiento de excepciones en la cual son declaradas por componentes las</p>	<p>code: para el tratamiento de un tipo de excepciones solo hay que conocer el código que le ha asignado el sistema donde se trabaja a la excepción que se desea tratar o ir al fichero de configuración de excepciones que es donde se encuentran todas las excepciones tratadas por el marco de trabajo.</p>

		<p>excepciones que pueden ocurrir en dicho componente.</p>	
<p>Variables</p>	<p><variable/></p>	<p>La definición de una variable indica la utilización posterior de la misma por otro elemento. Este elemento surge por la necesidad de usar valores asociados a elementos de cierto evento en un evento foráneo ya que una vez ejecutado un evento se puede proceder a la destrucción de sus elementos y con ello a</p>	<p>Id: mediante ese atributo se podrá referenciar el elemento y así obtener el valor que guarda.</p> <p>values: en este atributo se guardará el valor que tendrá la variable.</p> <p>reference: mediante este atributo podrá obtener valores de eventos.</p>

		<p>la pérdida de los valores de los elementos dentro del evento. Las variables guardarían los datos y solo tendrían que referenciarlas a través de su id para obtener su valor.</p>	
<p>Comentario</p>	<p><--comment--></p>	<p>Los comentarios son una vía para el usuario que este describiendo un proceso de interoperabilidad encuentre una vía para escribir anotaciones que le sirvan de guía dentro del documento. Esta etiqueta no deberá ser interpretada por el componente.</p>	

<p>Estructura</p>	<p><structure> </structure></p>	<p>Este elemento especifica la estructura de los datos a exportar pues cada proceso puede definir su propia estructura. La misma tiene en cuenta el formato definido en el marco de trabajo y la propuesta concebida puede coexistir con un porcentaje elevado de compatibilidad con XML.</p>	<p>Id: atributo utilizado en la referencia a este elemento desde un elemento o evento foráneo. name: cualquier combinación de caracteres válidos usada para nombrar la estructura.</p>
<p>Subproceso de importación</p>	<p><import> </import></p>	<p>Define el proceso de importación del proceso general.</p>	<p>name: cualquier combinación de caracteres válidos propuestos por XML.</p>
<p>Fileupload</p>	<p><fileupload/></p>	<p>Elemento muy utilizado especialmente por el subproceso de importación ya que mediante él se cargarán los archivos previamente exportados por un sistema homólogo.</p>	<p>name: Este atributo hace referencia al nombre que mostrara en la interfaz el elemento a la hora de importar ciertos archivos. dir: Este atributo guarda la dirección física de donde se cargarán los datos a importar. pointer: Atributo del elemento</p>

			que referencia el id de la variable donde se guardará los datos importados.
--	--	--	---

Figura 23: Tabla de elementos del LEPI.

CONCLUSIONES PARCIALES

Se diseñó el lenguaje de ejecución de proceso de interoperabilidad, teniendo en cuenta que cada elemento declarado debe ser implementado por el componente dinámico de interoperabilidad. Se hizo necesario añadirle características que facilitarían la implementación, manteniendo las peculiaridades del proceso y sus subprocesos. La principal ventaja obtenida con la realización de este lenguaje, se encuentra en evitar la realización de componentes individuales a cada solución implementada en Sauxe. Dicho lenguaje es capaz de describir las particularidades de cada proceso y aun así poder estandarizarlos. Además se obtuvo un DTD, que sirve para validar la buena formación y validez de la descripción de cada proceso.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

INTRODUCCIÓN

En el presente capítulo se realiza una validación teórica del LEPI propuesto. Esta se basa en el criterio de expertos, los cuales fueron seleccionados por su gran experiencia y habilidades en los diferentes temas abordados en la investigación. Con la aplicación de este método se obtienen sugerencias que contribuyen a mejorar la ejecución de la interoperabilidad implementada en el marco de trabajo Sauxe.

3.1 Proceso de validación

El criterio de expertos es un instrumento rápido y bastante eficaz por el potencial que contiene para conformar, valorar y enriquecer criterios, concepciones, modelos, estrategias, metodologías, entre otros. Este método utiliza técnicas como: encuestas, cuestionarios, entrevistas, estados de opinión, y Positivo – Negativo -Interesante. Para valorar la eficacia de la solución propuesta se utilizó la técnica de cuestionarios.[18]

Para validar el trabajo realizado por los expertos se emplea el coeficiente de concordancia de Kendall y el estadígrafo Chi cuadrado (X^2). [18]

La dificultad de este método radica fundamentalmente en que es basado en el criterio de un grupo de personas que por su experiencia pueden realizar esta evaluación, por lo que los resultados siempre pueden ser cuestionados. Este margen de error es reducido considerablemente si se hace una buena elaboración del cuestionario y sobre todo una buena elección de los expertos.

3.2 Elección de los especialistas

Elección de los especialistas.

Se entiende por experto a “una persona experimentada, que posea vastos conocimientos, experiencia y habilidad en determinada actividad, debe ser capaz de analizar el problema planteado y valorar si la solución propuesta es efectiva, siendo capaz de hacer recomendaciones al respecto”. [18]

Las características de los expertos influyen decisivamente en la confiabilidad de los resultados obtenidos. Estas características son: calificación técnica, capacidad de emitir una decisión al respecto, conocimientos específicos sobre el tema a evaluar, disposición a participar. De los especialistas candidatos fueron

escogidos siete, atendiendo a su disposición y nivel de competencia necesario para la realización del cuestionario.

Competencias necesarias para la selección de expertos.

- **Estándares de interoperabilidad:** El experto deberá tener experiencia y conocimiento acerca de estándares de intercambio de datos específicamente XML.
- **Interoperabilidad en aplicaciones web de gestión:** Deberá tener experiencia en el intercambio de datos entre aplicaciones web de gestión.
- **Implementación de soluciones en Sauxe:** El experto deberá tener experiencia en el desarrollo de aplicaciones sobre el marco de trabajo Sauxe.
- **Interoperabilidad en Sauxe:** Deberá tener conocimiento sobre la ejecución de procesos de negocio y de cómo se lleva a cabo la interoperabilidad en Sauxe.

Se determinó que para la competencia entre los expertos se tuvieron en cuenta los criterios recogidos en la siguiente tabla.

A continuación se relacionan por cada experto las características definidas anteriormente.

Especialista	Grado científico	Dominio de estándares de interoperabilidad	Interoperabilidad en aplicaciones web de gestión	Implementación de soluciones en Sauxe	Interoperabilidad en Sauxe
E1	Licenciado en Ciencias de la Computación	Alto	Alto	Medio	Medio
E2	Ingeniero en Ciencias Informáticas	Alto	Medio	Alto	Medio
E3	Ingeniero en Ciencias	Medio	Medio	Alto	Alto

	Informáticas				
E4	Ingeniero en Ciencias Informáticas	Alto	Alto	Medio	Alto
E5	Ingeniero en Ciencias Informáticas	Alto	Alto	Alto	Medio
E6	Ingeniero en Ciencias Informáticas	Medio	Medio	Alto	Medio
E7	Ingeniero en Ciencias Informáticas	Alto	Alto	Medio	Alto

Figura 24: Tabla de competencias de los expertos.

Se realiza el cálculo del coeficiente de competencia del experto a través de la siguiente expresión:

$$K = \frac{1}{2} (Kc. + Ka)$$

Donde:

Kc. – Coeficiente de conocimiento del experto sobre el criterio, el cual debe estar en un rango de 0 - 1.

Ka – Coeficiente de argumentación del experto sobre el criterio, tiene que estar en un rango de 0 - 1.

Con el resultado obtenido en K.

Si $0,8 < K < 1,0$ coeficiente de competencia **alto**.

Si $0,5 < K < 0,8$ coeficiente de competencia **medio**.

Si $K < 0,5$ coeficiente de competencia **bajo**.

Como se planteaba anteriormente, aunque la cantidad de expertos es proporcional en la eficiencia del método de validación, es de suma importancia que los expertos contengan un alto nivel de competencia para realizar el cuestionario. Debido a ello a pesar de tener un gran número de expertos solo se seleccionaron los que contaban con un nivel de competencia mayor de 0,5 según los cálculos realizados.

3.3 Elaboración del cuestionario

El cuestionario se elaboró de acuerdo al cumplimiento de tres elementos que resultan necesarios para la descripción y desarrollo de un lenguaje de ejecución. Estos se relacionan a continuación:

- **Fácil integración y ejecución.**
- **Usabilidad y calidad.**
- **Objetividad.**

Teniendo en cuenta estos aspectos se conformaron las preguntas que atribuyen a la evaluación de los mismos. Dado el cumplimiento de estos aspectos es decir, una evaluación positiva por parte de los expertos sobre las preguntas que atribuye a cada aspecto, se puede sostener la **hipótesis**, que la solución planteada cumple con los objetivos previstos.

Teniendo en cuenta estos aspectos se conformaron las preguntas a realizar por cada experto. Las preguntas con corte evaluativo hacia **fácil integración y ejecución** de la solución propuesta tienen como objetivo evaluar la capacidad del LEPI para integrarse en Sauxe y ejecutar los procesos de interoperabilidad de las distintas soluciones. En este aspecto se deben tener en cuenta el criterio de los programadores, comprobantes de operaciones e inventario.

1. **Facilidad de implementación:** Criterio dirigido a conocer si una vez descrita la ejecución de un proceso utilizando la solución propuesta, puede implementarse un componente de interoperabilidad.
2. **Coherencia entre los elementos integrantes:** Criterio que indicará el nivel de concordancia y la relación que existe entre los elementos que conforman la solución y que a su vez puedan simular la correspondencia que existe entre las diversas actividades que conforman un proceso.
3. **Ajuste a las tecnologías:** Se trata de evaluar si la implementación de un componente de interoperabilidad que utilice la solución propuesta para ejecutar los procesos de interoperabilidad, se ajusta a las tecnologías de desarrollo que requiere el marco de trabajo Sauxe.
4. **Integración con Sauxe:** Si es posible integrar la solución con Sauxe de acuerdo a la arquitectura y la tecnología que utiliza.
5. **Alternativa al manejo de errores:** Criterio que evalúa la posibilidad de manejar eventos alternos al ciclo normal de ejecución de cualquier proceso.
6. **Mecanismo para la ejecución, control y configuración:** Este criterio evalúa la accesibilidad y la forma con que cuenta el usuario para configurar el proceso de interoperabilidad.

7. **Mecanismo para el manejo de datos:** Trata de evaluar la forma en que se obtiene la información proveniente de los servicios de integración que se consumen.
 8. **Mecanismo para la configuración de estructura:** Este criterio trata de evaluar cuantitativamente la manera en que se trabaja con la estructura que tendrán los datos que se obtienen de cada subproceso. Las preguntas que atribuyen a comprobar el **grado de usabilidad**, tienen como objetivo conocer si es posible el uso del lenguaje propuesto en la ejecución del proceso de interoperabilidad.
 9. **Correcta visión de la solución atendiendo al problema:** en esta pregunta se trata de evaluar si la solución es la más apropiada a pesar de que puedan existir varias formas de solucionar la problemática.
 10. **Facilidad de uso:** evalúa la habilidad que puede tener el usuario en la utilización del LEPI para la ejecución del proceso de interoperabilidad.
 11. **Descripción de cada elemento del proceso:** Una vez identificadas las actividades que conforman un proceso de interoperabilidad, se pueden describir cada elemento, secuencia, acción o atributo de dichas actividades con los diferentes mecanismos o elementos que provee el lenguaje.
 12. **Capacidad para describir varios procesos:** Evalúa si es posible describir la ejecución de todos los procesos pertenecientes a todas las soluciones desarrolladas actualmente en Sauxe.
 13. **Accesibilidad:** Evalúa si es posible acceder a la definición del tipo de documento (DTD).
- Las preguntas relacionadas con la **objetividad** permiten medir el nivel de integridad de la solución y cumplimiento de los objetivos
14. **Necesidad e Importancia:** De forma cualitativa la importancia que le atribuyen los especialistas a la correcta ejecución un proceso de interoperabilidad.

Tipo de preguntas	Descripción	Cantidad
Fácil integración y ejecución	Dirigidas a evaluar la integración con el marco de trabajo Sauxe.	8
Grado de usabilidad	Dirigidas a la evaluación por parte de los usuarios que	5

Objetividad de la solución	usarán el LEPI propuesto.	
	Evalúa si es correcta o no la solución y si cumple con los objetivos propuestos.	1

Figura 25: Tabla de aspectos para confección de cuestionario.

3.4 Desarrollo del proceso de validación

En la encuesta se solicita que las preguntas sean evaluadas mediante 5 niveles, a cada nivel se le dio una calificación cuantitativa en aras de la correcta ejecución y realización de los cálculos pertinentes. Los niveles fueron: Excelente (5), Muy bueno (4), Bueno (3), Regular (2), Malo (1). Cada nivel tiene un valor numérico asociado.

Una vez seleccionados los especialistas, confeccionado el cuestionario y regulado el proceso de evaluación de cada pregunta dada por los expertos, fueron entregados a los expertos las encuestas con las 14 preguntas seleccionados para la evaluación, ellos valoraron la importancia que presenta cada aspecto relacionado con las preguntas que debieron responder para la interoperabilidad en las aplicaciones web de gestión y específicamente las aplicaciones desarrolladas en Sauxe, así como el grado de significación que se le confiere al estudio de dicha temática.

En dicha encuesta se obtuvieron opiniones y consejos, los cuales sirvieron de mucha ayuda en la ejecución de los disimiles proceso de interoperabilidad en Sauxe.

	E1	E2	E3	E4	E5	E6	E7
P1	5	4	5	5	5	5	4
P2	4	4	4	3	4	4	4
P3	5	5	4	4	3	5	4
P4	5	5	5	5	5	5	5
P5	5	5	4	5	5	5	5
P6	4	4	4	3	5	4	4

P7	5	4	4	4	4	5	4
P8	5	5	4	4	4	4	5
P9	5	4	4	4	3	5	4
P10	5	5	5	5	5	4	5
P11	5	3	4	3	4	3	5
P12	5	4	4	5	4	5	5
P13	5	4	5	4	4	5	5
P14	5	4	4	4	5	5	5

Figura 26: Tabla de evaluación de los criterios por los especialistas.

Calcular la concordancia de los expertos

Si bien fue importante determinar el nivel de competencia de los expertos para saber que se escogió a la persona correcta, es de igual importancia calcular la concordancia entre ellos, porque puede que este sea un buen experto pero su criterio puede ser desigual al del grupo, ya que todos tienen la posibilidad de plantear opiniones únicas que no siempre resultan ser las adecuadas.

Después de obtenida una proposición del peso para los criterios en la consulta a los expertos se necesitó demostrar la fiabilidad de sus respuestas comprobando el nivel de acuerdo entre estos para otorgar mayor confiabilidad. Se puede utilizar entonces el coeficiente de concordancia de Kendall. [18]

Para obtener el coeficiente de Kendall lo primero que conviene tener en cuenta es identificar las variables que se van a utilizar, para esto, en este caso se tendría que:

m = números de expertos.

n = número de criterios.

VALIDACIÓN DE LA PROPUESTA DE SOLUCIÓN

	E1	E2	E3	E4	E5	E6	E7	Sj
P1	5	4	5	5	5	5	4	33
P2	4	4	4	3	4	4	4	27
P3	5	5	4	4	3	5	4	30
P4	5	5	5	5	5	5	5	35
P5	5	5	4	5	5	5	5	34
P6	4	4	4	3	5	4	4	28
P7	5	4	4	4	4	5	4	30
P8	5	5	4	4	4	4	5	31
P9	5	4	4	4	4	5	4	30
P10	5	5	5	5	5	4	5	34
P11	5	3	4	3	4	3	5	27
P12	5	4	4	5	4	5	5	32
P13	5	4	5	4	4	5	5	32
P14	5	4	4	4	5	5	5	32
ΣE	68	60	60	58	61	64	64	435
T	143.5	52.66	87.5	28	45.5	65.75	59.5	ΣT=170.55

Figura 27: Tabla de cálculo de la variación media.

Sj = Refleja la suma de rangos correspondientes a la evaluación realizada por los expertos a la pregunta j y se hallaría por la fórmula:

$$S_j = \sum_{i=1}^m R_{ij}$$

(R_{ij} es el rango asociado a la evaluación del experto "i" a la pregunta "j"). [18]

Se define también la media de la suma de rangos de cada pregunta j.

$$\bar{S} = \frac{\sum_{j=1}^n S_j}{n}$$

S media= $435/14 = 31.071428 = 31.07$

Con estos valores es posible entonces obtener el resultado del coeficiente de Kendall (k).[19]

$$K = \frac{S}{\frac{1}{12} m^2 (n^3 - n) - m \sum_{i=1}^m T_i}$$

$K = 0.48$

Donde:

T_i es el resultado de los rangos iguales, llamados también ligas, que ofreció el experto i para las preguntas y se calcula:[19]

$$T_i = \frac{\sum_{t=1}^i (t^3 - t)}{12}$$

Donde:

i : número de grupos con rangos iguales para el experto i .

t : número de observaciones dentro de cada uno de los grupos para el experto i .

“Los valores del coeficiente “ k ” deben oscilar entre 0 y 1 ($0 < k < 1$), si k alcanza el valor uno ($k = 1$) entonces existe una concordancia total de criterios, mientras mayor sea el valor de k , es decir, cuanto más se acerque a uno, mayor será la concordancia entre los expertos.” [18]

Validación de la hipótesis propuesta.

Luego se aplica la prueba de significación de hipótesis para comprobar el grado de significación de Kendall, planteándose la hipótesis nula y la alternativa de la siguiente forma:

H_0 : no existe concordancia entre los expertos, $k = 0$.

H_1 : existe concordancia entre los expertos, $k \neq 0$.

Se determina Chi-cuadrado calculado como: $X^2_{cal} = m (n - 1) k$

$X^2_{cal} = 7 (14-1) * 0.48 = \mathbf{43.68}$

“Por otra parte, se busca el Chi-cuadrado tabulado en la tabla del percentil de la distribución Chi-cuadrado con un nivel de significación $n - 1$ grados de libertad, representado por

$X^2_{tabla} = X^2_{\alpha; n-1}$.” [18]

$X^2_{tabla} = X^2_{\alpha; n-1} = \mathbf{23.7}$

$43.68 > 23.7$

$X^2_{cal} > X^2_{tabla}$

Por lo que se procede a rechazar la hipótesis nula y se acepta la hipótesis alternativa concluyéndose así que existe concordancia entre los expertos.

Si no se obtuvo un consenso entre las respuestas de los expertos se procede a eliminar el experto que más variación introduce en el estudio, respetando siempre que el número total de estos sea ≥ 7 .

Una vez calculado el nivel de concordancia entre los especialistas se procede a conocer el promedio de evaluación por las preguntas y los aspectos seleccionados a evaluar.

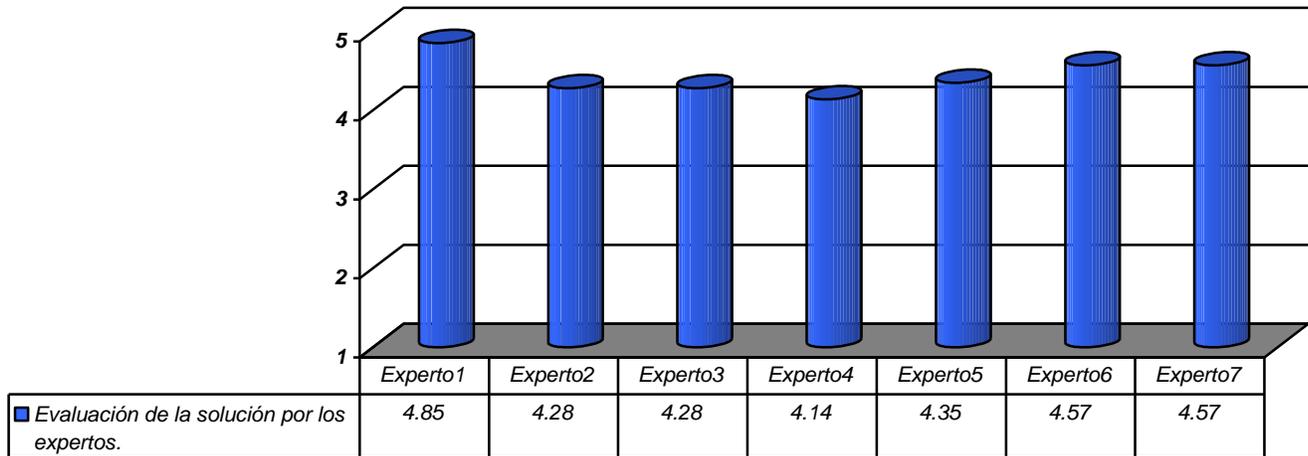


Figura 28: Promedio de evaluación por especialistas.

La siguiente gráfica muestra el promedio de evaluación que obtuvo cada uno de los aspectos enumerados según corresponde en el cuestionario, por parte de los especialistas encuestados. La evaluación de cada uno de estos aspectos unido a los cálculos realizados anteriormente ratifica que la utilización del LEPI podría solucionar la problemática existente dotando al componente de interoperabilidad del marco de trabajo Sauxe las herramientas necesarias para la ejecución de cualquier proceso de interoperabilidad concebido por cualquier solución implementada sobre Sauxe.

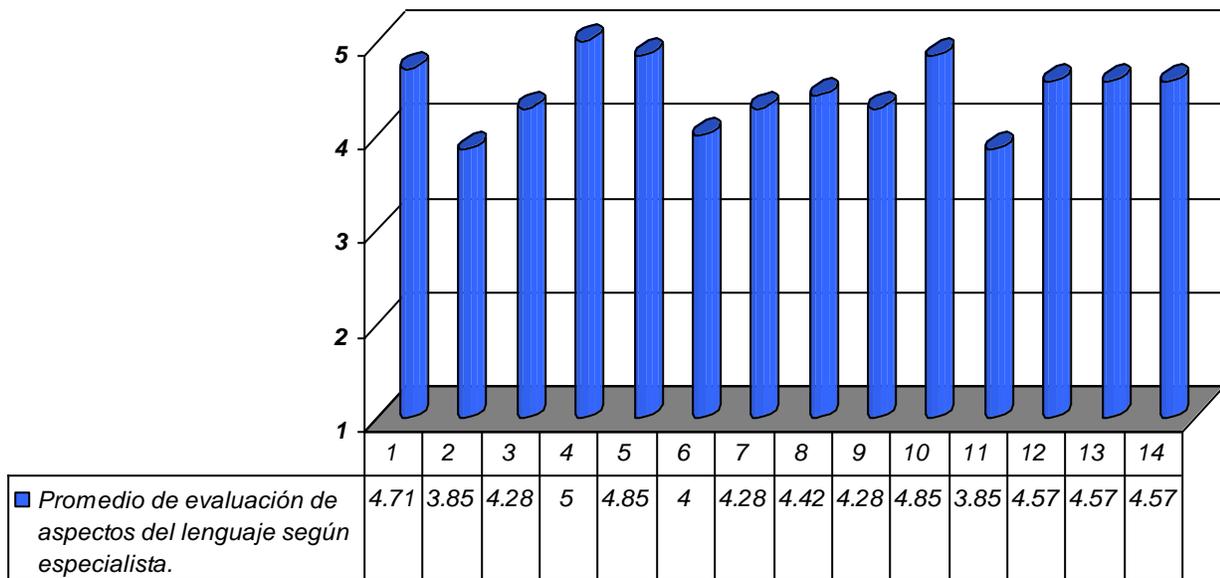


Figura 29: Evaluación de la solución por los especialistas.

3.5 Solución aplicada a un proceso

El uso estricto de cada uno de los elementos garantiza que se logre describir la ejecución del proceso de manera controlada. Para un mejor entendimiento de lo anteriormente expuesto, se efectuó la descripción de la ejecución del proceso de interoperabilidad de contabilidad (Anexo 40); específicamente comprobante de operaciones; utilizando LEPI. El componente de interoperabilidad de comprobante de operación es uno de los componentes independientes que brinda dicho servicio, solo a la solución de contabilidad.

Conclusiones parciales

El análisis de los resultados anteriores permite afirmar que es posible utilizar LEPI en el desarrollo de la interoperabilidad utilizando Sauxe pues las encuestas realizadas a los expertos arrojaron resultados positivos en función de los indicadores establecidos. La propuesta fue aplicada en un caso de estudio real, demostrando su capacidad para la descripción y ejecución de la interoperabilidad en Sauxe.

CONCLUSIONES

Al concluir el presente trabajo, se dio cumplimiento al objetivo general propuesto en la investigación. Se propuso un LEPI en los sistemas que utilizan el marco de trabajo Sauxe. Para lograr dicho objetivo se cumplieron las siguientes tareas investigativas.

➤ **Realización de un estudio bibliográfico sobre los diferentes lenguajes de ejecución de procesos.**

Dicho estudio propicio los elementos necesarios para conocer el funcionamiento y características de los lenguajes de ejecución de procesos existentes a nivel internacional, específicamente BPEL y WS-BPEL.

➤ **Realización de un estudio minucioso acerca del proceso de interoperabilidad en los sistemas que emplean el marco de trabajo Sauxe.**

Una vez analizado los lenguajes de ejecución de procesos de negocio, se realizó un estudio sobre la ejecución del proceso de interoperabilidad en el marco de trabajo Sauxe. Este estudio arrojó las bases para lograr definir una solución que cumpliera con las características arquitectónicas de Sauxe, y que a su vez fuera capaz de lograr estandarizar la ejecución del proceso de interoperabilidad en dicho marco de trabajo.

Definición de un lenguaje de ejecución de proceso de interoperabilidad en los sistemas que utilizan el marco de trabajo Sauxe.

Una vez concluido que no existía ningún lenguaje de ejecución de procesos que lograra cumplir con las características necesarias para integrarse con Sauxe, se definió un lenguaje de ejecución de procesos de interoperabilidad. De dicho lenguaje se especificó la sintaxis, semántica, estructura y el uso que se le puede dar a los elementos que lo conforman, así como el DTD que valida un archivo de descripción del proceso de interoperabilidad.

➤ **Validación del lenguaje de ejecución de procesos de interoperabilidad en los sistemas que utilizan el marco de trabajo Sauxe.**

La propuesta de solución se realizó utilizando el método criterio de experto, quienes contienen un alto nivel de experiencia y conocimiento en competencias relacionadas con la ejecución de procesos de negocio, implementación de aplicaciones web de gestión y ejecución del proceso de interoperabilidad en Sauxe.

RECOMENDACIONES

Una vez que se cumplieron los objetivos de la investigación el autor recomienda el uso de todos los elementos definidos en la misma, para la implementación del componente de interoperabilidad. Además se hace énfasis en, lograr la implementación de una solución que permita la descripción de los procesos de interoperabilidad usando el LEPI de manera automatizada, y que a su vez sea compatible con el marco de trabajo Sauxe. Ya que dicha solución solo permite la descripción de la ejecución del proceso de interoperabilidad en el marco de trabajo de forma manual. Esta solución eliminaría los errores humanos a la hora de describir la ejecución de dicho proceso.

BIBLIOGRAFÍA

1. **Acero, Fernando.** *kriptopolis.org/interoperabilidad-4. kriptopolis.org/interoperabilidad-4.* [Online] 2009. [Cited: diciembre 9, 2011.] <http://www.kriptopolis.org/interoperabilidad-4..>
2. **Fingermann, Hilda.** El lenguaje, concepto y definición. *lengua.laguia2000.com.* [Online] Octubre 2009. [http://lengua.laguia2000.com/.](http://lengua.laguia2000.com/)
3. **EllisLab .** *www.Codelgniter.com. www.Codelgniter.com.* [Online] 2009. Codelgniter.com.
4. **Barros, Dr. Oscar.** *Reingeniería de Procesos de Negocio.Un Enfoque Metodologico.*
5. **Davenport, Thomas.** *Process Innovation.* USA : Harvard Business School Press, 1993.
6. **ISO.** Norma Internacional ISO 9001 . *Sistemas de gestión de la calidad .* Ginebra : Secretaría Central de ISO en Ginebra, 2000. 9001.
7. **J. C. Álvarez, M. Mateos, M. N. Moreno.** *Metodología de reingeniería del software para la remodelación de aplicaciones científicas heredadas . .* Salamanca : Universidad de Salamanca, 2004. DPTOIA-IT.
8. **Software Engineering Institute.** <http://www.sei.cmu.edu/reengineering/lsysree.pdf>. *Perspectives on Legacy System Reengineerin.* [Online] Carnegie Mellon University. <http://www.sei.cmu.edu/reengineering/lsysree.pdf>.
9. **Codorniu, Ing. Cesar Lage.** *Guía de referencia para el desarrollo de marcos de trabajo tecnológicos de sistemas en dominios de gestión.* Cuba : UCI, 2009.
10. **Baryolo, Oiner Gomes.** *Libro de Ayuda de la Arquitectura Tecnológica del ERP-Cuba.* Cuba : UCI, 2009.
11. **W3C-XML.** "World Wide Web Consortium. Extensible Markup Language (XML) 1.0". [Online] World Wide Web Consortium, Mayo 2009. <http://www.w3.org/TR/1998/REC-xml-19980210.html>.
12. **W3C-BPEL.** WS-BPEL. [Online] 2009. <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>.
13. **Microsoft.** <http://www.asia.microsoft.com/latam/msdn/articulos/2000/04/art02/#top>. [Online] Microsoft, 2009. <http://www.asia.microsoft.com/latam/msdn/articulos/2000/04/art02/#top>.
14. **W3C-XSL.** XSL Transformation (XSL). [Online] World Wide Web Consortium. <http://www.w3.org/TR/1999/WD-xsl-19990421.html>.

15. **María, Msc. Frida Ortiz García.** *Metodología de la Investigación*. Mexico : Limosa, 2005.
16. **W3C-DTD.** W3C-DTD. *W3C.org*. [Online] <http://www.w3.org/TR/html4/sqml/dtd.html>.
17. **Borges, Alejandro.** La interoperabilidad y los estándares abiertos, Base de desarrollo de la Sociedad de la Información. [Online] 2007. <http://www.coit.es/publicaciones/bit/bit161/36-39.pdf>.
18. **CEPAL, Unión Europea.** *Libro blanco de interoperabilidad de gobierno electrónico para América Latina y el Caribe*. 2008.
19. **Gobierno Brasileño Comitê Ejecutivo de Gobierno Electrónico.** *Estándares de Interoperabilidad de Gobierno Electrónico*. . 2008.
20. **Model LISI.** Levels of Information Systems Interoperability. [Online] 2008. <http://www.bmpcoe.org/library/books/lisi%20model>.
21. **Menezes, Carlos.** ERPS-Planificación de Recursos Empresariales. *ERPS-Planificación de Recursos Empresariales*. [Online] 2010. <http://cmenez.wordpress.com/2007/02/19/erps-planificacion-de-recursos-empresariales/>.
22. **Munk, Sandor.** *INTEROPERABILITY IN THE INFOSPHERE – CHALLENGES, PROBLEMS, SOLUTIONS*. . 2009.
23. *La interoperabilidad perspectiva de Microsoft*. 2009.
24. **Cobas, Pedro Manuel Nogales.** *Estrategia para la interoperabilidad en sistemas ERP en Cuba*. Ciudad Habana : UCI, 2010.
25. **Española, Asociación de Academias de la Lengua.** Real Academia Española. *Real Academia Española*. [Online] Asociación de Academias de la Lengua Española, 2001. [Cited: Junio 8, 2012.] <http://buscon.rae.es/drael/>.
26. **Visconti, Marcello and Astudillo, Hernán.** *Fundamentos de Ingeniería de Software*. Santa María : Universidad Técnica Federico Santa María.
27. **Uribe, Rodolfo.** *XBRL:El idioma universal financiero y de negocios*. 2009. Vol. II.
28. **DMI,Data Mining Institute** . *Diccionario estadístico*. Madrid : s.n., 2009.

REFERENCIAS BIBLIOGRÁFICAS.

1. Osoria, J.R.M.R.-E.R., *Guía metodológica de interoperabilidad para sistemas integrales de gestión en Cuba*, in UCI. 2010, UCI: Ciudad de la Habana.
2. Baryolo, O.G., *Libro de Ayuda de la Arquitectura Tecnológica del ERP-Cuba.* , UCI, Editor. 2009.
3. Donald B. Lemke, T.S., Al Milgrom, *Investigating the Scientific Method*, ed. CAPSTONE-USA. 2008.
4. Acero, F. 2009; Available from: <http://www.kriptopolis.org/interoperabilidad-4>.
5. ISO. 1990.
6. *IEEE Standard Computer Dictionary*. 1990: New York.
7. Cobas, P.M.N. 20012, UCI: Ciudad de la Habana.
8. Nogales, P.M. [Estrategia de interoperabilidad para la transferencia de datos entre sistemas ERP en Cuba.] 2011; Available from: <http://www.cyta.com.ar/ta1002/v10n2a3.htm>.
9. W3C. *Standards*. 2009; Available from: <http://www.w3.org/standards/>.
10. W3C. *W3C-XML*. 2009; Available from: <http://www.w3.org/XML/>.
11. W3C-XBRL, *XBRL and the Semantic Web*. 2009.
12. Tejo, O.G.B.y.D.G., *Plantilla Registro de la Propiedad intelectual (Sauxe)*. 2009.
13. Electrónicos, I.d.I.E.y., *IEEE P1484.12.3*. 2009.
14. W3C. *XSL Transformation (XSL)*. Available from: <http://www.w3.org/TR/1999/WD-xsl-19990421.html>.
15. W3C-BPEL. *W3C-BPEL*. 2009; Available from: <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.pdf>.
16. Calvo, J.V.y.R.A., *Concept Maps as Cognitive Visualizations of Writing Assignments*. University of Sydney, Australia, 2008.
17. W3C-DTD. 2011; Available from: <http://www.w3.org/TR/html4/sgml/dtd.html> .
18. Kitchenham, B., *Principles of Survey Research*. Keele University, 2003.
19. Sheskin, D.J., *Kendall 's Coefficient of Concordance*. 2008.

ANEXOS

En la encuesta se solicita que las preguntas sean evaluadas mediante 5 niveles, a cada nivel se le dio una calificación cuantitativa en aras de la correcta ejecución y realización de los cálculos pertinentes. Los niveles fueron: Excelente (5), Muy bueno (4), Bueno (3), Regular (2), Malo (1). Cada nivel tiene un valor numérico asociado.

ASPECTOS	PREGUNTAS	EVALUACIÓN
Objetividad	Necesidad e Importancia	5
	Facilidad de implementación.	4
Fácil integración y ejecución	Coherencia entre los elementos integrantes	5
	Ajuste a las tecnologías	5
	Integración con Sauxe	5
	Alternativa al manejo de errores.	4
	Mecanismo para la ejecución, control y configuración del proceso	5
	Mecanismo para el manejo de datos	5
	Mecanismo para la configuración de estructura	5
Usabilidad	Correcta visión de la solución atendiendo al problema.	5
	Facilidad de uso	5
	Descripción de cada elemento del proceso	5
	Capacidad para describir varios procesos.	5
	Accesibilidad	5

Nombre y apellido del encuestado:

Ing. Maglanes Galvan Rey

Grado Científico:

Ingeniero

Centro y Departamento y Rol dentro de ellos:

CETGE, Tecnología, Analista Principal

Figura 30: Encuesta realizada a expertos.

En la encuesta se solicita que las preguntas sean evaluadas mediante 5 niveles, a cada nivel se le dio una calificación cuantitativa en aras de la correcta ejecución y realización de los cálculos pertinentes. Los niveles fueron: Excelente (5), Muy bueno (4), Bueno (3), Regular (2), Malo (1). Cada nivel tiene un valor numérico asociado.

ASPECTOS	PREGUNTAS	EVALUACIÓN
Objetividad	Necesidad e Importancia.	4
Facil integración y ejecución	Facilidad de implementación.	4
	Coherencia entre los elementos integrantes	5
	Ajuste a las tecnologías	5
	Integración con Sauxe	5
	Alternativa al manejo de errores.	4
	Mecanismo para la ejecución, control y configuración del proceso.	4
	Mecanismo para el manejo de datos	5
Usabilidad	Mecanismo para la configuración de estructura	4
	Correcta visión de la solución atendiendo al problema.	5
	Facilidad de uso	5
	Descripción de cada elemento del proceso.	4
	Capacidad para describir varios procesos.	4
	Accesibilidad	4

Nombre y apellido del encuestado:

Javier Ruiz Durán

Grado Científico:

Centro y Departamento y Rol dentro de ellos:

CEIGE - Tecnología - Arquitecto

Figura 31: Encuesta realizada a expertos.

En la encuesta se solicita que las preguntas sean evaluadas mediante 5 niveles, a cada nivel se le da una calificación cuantitativa en aras de la correcta ejecución y realización de los alcances definidos. Los niveles fueron: Excelente (5), Muy bueno (4), Bueno (3), Regular (2), Malo (1). Cada nivel tiene un valor numérico asociado.

ASPECTOS	PREGUNTAS	EVALUACION
Objetividad	Necesidad e Importancia.	5
	Facilidad de implementación.	4
	Coherencia entre los elementos integrantes.	5
	Ajuste a las tecnologías.	5
Facil integración y ejecución	Integración con Saude.	4
	Alternativa al manejo de errores.	5
	Mecanismo para la ejecución, control y configuración del proceso.	5
	Mecanismo para el manejo de datos.	5
Usabilidad	Mecanismo para la configuración de estructura.	5
	Conecta visión de la solución atendiendo al problema.	5
	Facilidad de uso.	5
	Descripción de cada elemento del proceso.	5
	Capacidad para describir varios procesos.	5
	Accesibilidad.	5

Nombre y apellido del encuestado

Samuel Luis Rojas

Grado Científico

-

Centro y Departamento y Rol dentro de ellos

CE-E Tecnología Asistida

Figura 32 : Encuesta realizada a expertos.

En la encuesta se solicita que las preguntas sean evaluadas mediante 5 niveles, a cada nivel se le dio una calificación cuantitativa en aras de la correcta ejecución y realización de los cálculos pertinentes. Los niveles fueron: Excelente (5), Muy bueno (4), Bueno (3), Regular (2), Malo (1). Cada nivel tiene un valor numérico asociado.

ASPECTO	PREGUNTAS	EVALUACIÓN
Objetividad	Necesidad e Importancia	5
	Facilidad de implementación.	4
Facil integración y ejecución	Coherencia entre los elementos integrantes	5
	Ajuste a las tecnologías.	5
	Integración con Sauxe	5
	Alternativa al manejo de errores.	5
	Mecanismo para la ejecución, control y configuración del proceso.	5
	Mecanismo para el manejo de datos.	5
	Mecanismo para la configuración de estructura	5
Usabilidad	Correcta visión de la solución atendiendo al problema	5
	Facilidad de uso.	5
	Descripción de cada elemento del proceso	5
	Capacidad para describir varios procesos	5
	Accesibilidad.	5

Nombre y apellido del encuestado

Rosé Banta

Grado Científico

-

Centro y Departamento y Rol dentro de ellos.

CEPSE - Depto. Tecnología - J' Projects.

Figura 33: Encuesta realizada a expertos.

En la encuesta se solicita que las preguntas sean evaluadas mediante 5 niveles, a cada nivel se le dio una calificación cuantitativa en aras de la correcta ejecución y realización de los cálculos pertinentes. Los niveles fueron: Excelente (5), Muy bueno (4), Bueno (3), Regular (2), Malo (1). Cada nivel tiene un valor numérico asociado

ASPECTOS	PREGUNTAS	EVALUACIÓN
Objetividad	Necesidad e Importancia.	5
	Facilidad de implementación.	4
	Coherencia entre los elementos integrantes.	4
Facil integración y ejecución.	Ajuste a las tecnologías.	5
	Integración con Sauxe.	4
	Alternativa al manejo de errores.	4
	Mecanismo para la ejecución, control y configuración del proceso.	4
	Mecanismo para el manejo de datos.	4
Usabilidad	Mecanismo para la configuración de estructura.	4
	Correcta visión de la solución atendiendo al problema.	5
	Facilidad de uso.	4
	Descripción de cada elemento del proceso.	4
	Capacidad para describir varios procesos.	4
	Accesibilidad	4

Nombre y apellido del encuestado:

Pedro Manuel Nagales Cobos

Grado Científico:

Ingeniero en Ciencias Informáticas

Centro y Departamento y Rol dentro de ellos:

CEIBE Tecnología. Jefe de proyecto.

Figura 34: Encuesta realizada a expertos.

En la encuesta se solicita que las preguntas sean evaluadas mediante 5 niveles, a cada nivel se le dio una calificación cuantitativa en aras de la correcta ejecución y realización de los cálculos pertinentes. Los niveles fueron: Excelente (5), Muy bueno (4), Bueno (3), Regular (2), Malo (1). Cada nivel tiene un valor numérico asociado.

ASPECTOS	PREGUNTAS	EVALUACION
Objetividad	Necesidad e importancia.	5
	Facilidad de implementación.	3
Fácil integración y ejecución	Coherencia entre los elementos integrantes.	5
	Ajuste a las tecnologías.	4
	Integración con Sauxe	5
	Alternativa al manejo de errores.	5
Usabilidad	Mecanismo para la ejecución, control y configuración del proceso.	5
	Mecanismo para el manejo de datos.	5
	Mecanismo para la configuración de estructura.	5
	Correcta visión de la solución atendiendo al problema.	3
Usabilidad	Facilidad de uso.	5
	Descripción de cada elemento del proceso.	5
	Capacidad para describir varios procesos.	5
	Accesibilidad.	5

Nombre y apellido del encuestado:

Gilberto Arias Navarrete

Grado Científico:

Centro y Departamento y Rol dentro de ellos:

Fac 2: Departamento de Innovación

Figura 35: Encuesta realizada a expertos.

```

<?xml version="1.0" encoding="iso-8859-1"?>
<process name="Interoperar_Contabilidad_Financiera">
  <export name="Exportar_comprobantes_de_operaciones">
    <variable id="" reference="service o acción sobre components internos" name="" valor="" />
    <!-- Esta variable guardara el json del grid que contendra los comprobantes seleccionados-->
    <variable id="224" reference="222" valor="comprobantes.json" name="comprobantes"/>
    <variable id="279" reference="512" valor="Comprobantes.rar" name="compactado"/>
    <!-- Aqui empieza el evento de seleccionar los comprobantes segun el ejercicio el periodo y el estado-->
    <evento_descripcion id="001" name="Obtener_Comprobantes">
      <view name="Comprobantes disponibles ">
        <form id="0011" name="name del formulario">
          <comboBox name="Selección de Ejercicio" idreference="104" pointer="id de variable"/>
          <comboBox name="Selección de Periodo" idreference="21" pointer="id de variable"/>
          <comboBox name="Selección de Estado" idreference="23" pointer="id de variable"/>
          <Grid id="222" reference="223" valor="Datos del servicio 223"/>
          <reset_button name=" " tipo="resetear"/>
          <submit_button name=" " reference="Solo a un manejador de evento"/>
        </form>
      </view>
      <!--Este servicio se ejecuta sin parameter por lo que la etiqueta parameter no tiene atributos-->
      <service id="104" name="Obtener_Ejercicios" component="Contabilidad_Financiera">
        <parameter/>
      </service>
      <!--Este servicio dado un ejercicio seleccionado brindara todos los periodos asociados-->
      <service id="21" name="Obtener_Periodo" component="Contabilidad_Financiera">
        <parameter name="ejercicio" reference="19"/>
      </service>
      <!--Este servicio dado un ejercicio y un periodo seleccionado brindara todos los estados asociados-->
      <service id="23" name="Obtener_Estado" component="Contabilidad_Financiera">
        <parameter name="ejercicio" reference="19"/>
        <parameter name="periodo" reference="22"/>
      </service>

      <service id="223" name="Obtener_Comprobantes" component="Contabilidad_Financiera">
        <parameter name="ejercicio" reference="19"/>
        <parameter name="periodo" reference="22"/>
        <parameter name="estado" reference="105"/>
      </service>
      <!--el code especifica el name de la etiqueta que contiene la exception que se quiere lanzar este c
      <!--Este es la etiqueta que invoca el proximo evento-->
      <manejador_de_eventos reference="002"/>
      <exception code="CF059" tipo="LP"/>
    </evento_descripcion>
    <evento_execution id="002" name="Obtener Comprobantes">
      <vista name=" Configurar atributos de los comprobantes">
        <form>
          <tree id="369" reference="0002" valor="" />
        </form>
        <form>
          <Fileupload id="888" reference="279" dir="../../../../../../Archivos del Sistema/Interoperabilidad"/>
        </form>
      </vista>
      <service id="512" name="Obtener Datos Especificos de un grupo de comprobantes dados ">
        <Integrator>
          <component name="Contabilidad_Financiera">
            <parameter name="comprobantes" reference="224"/>
            <parameter name="Datos especificos" reference="369"/>
          </component>
        </Integrator>
        <!--Por si existe algun flujo alterno en caso de la ejecucion del evento-->
        <exception code="CF059" tipo="LP"/>
      </service>
      <manejador_de_eventos ref="888"/>
    </evento_execution>
  </export>

```

Figura 36: Descripción de la ejecución del subproceso de exportación de comprobantes de operaciones.