

Universidad de las Ciencias Informáticas
Facultad 5



*Título: Diseño de una guía de referencia para la integración
de datos en una Arquitectura Orientada a Servicios.*

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

Autor: Lester Leonardo González Rodríguez

Tutor: Ing. Leevan Abón Cepeda

Co-Tutor: Yudisbel Rojas Cruz

La Habana
Junio de 2012

DECLARACIÓN DE AUTORÍA

Declaro ser el autor de la tesis “Diseño de una guía de referencia para la integración de datos en una Arquitectura Orientada a Servicios” y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Lester Leonardo González Rodríguez

Ing. Leevan Abón Cepeda

AGRADECIMIENTOS

Quiero agradecer a un conjunto de personas que sin ellos no hubiera sido posible la realización de la presente investigación. Por tanto se agradece explícitamente a:

Mis padres por apoyarme siempre a pesar de las circunstancias, por sus valiosos consejos, por estar siempre pendientes de mí, por convertirme en la persona que soy hoy, por luchar incansablemente por sus hijos y por permitir que hoy les pueda agradecer todo lo que ustedes han hecho por mí.

Mi hermana Lumey por ser la mejor hermana del mundo, por todo su amor y por darme la confianza para seguir adelante. A mi sobrina que la quiero como a una hija aunque no haya podido verla crecer como quisiera.

Mi hermano Leiser por ser más que un hermano y amigo. Por su apreciada compañía a lo largo de toda mi vida, así como su apoyo incondicional. Por apoyarme siempre, confiar en mí y estar conmigo en las buenas y en las malas.

Mi prima Naily y su novio Sandy, por su paciencia, por guiarme y orientarme, por todo su apoyo y ayudarme siempre cuando me ha hecho falta. Por sus consejos y darme fuerza cuando más las he necesitado. Sin ustedes no hubiese sido posible este trabajo.

Mi primo Freddy, por ser un ejemplo en el plano profesional y estar siempre dispuesto a ayudar.

Mi familia, abuelos, tíos y primos que siempre estuvieron pendientes de mí.

Mis amigos Yuniesky Cárdenas, Erick Dévora y Javier Lafita, más que amigos los considero como hermanos, porque a lo largo de estos años siempre me han dado confianza, apoyado y creído en mí.

A mis compañeros de los diferentes grupos por los que he pasado, por compartir tantos momentos buenos y malos.

Mi tutor, el Ing. Leevan Abón Cepeda y mi co-tutora Yudisbel Rojas Cruz, por encontrar el tiempo libre para atenderme, por las indicaciones y correcciones, así como el apoyo brindado.

A todas las personas que de una forma u otra contribuyeron a la realización de la presente investigación, los mencionados y a otros que no están porque harían la lista interminable. Por todo su apoyo y confianza en mí, les estaré agradecido eternamente.

DEDICATORIA

Quiero dedicar la presente investigación a mis padres, por la educación que me han inculcado, sin los cuales, nada de lo que soy hubiera sido posible. A mi hermana y sobrina por su apoyo incondicional. A mi hermano por ser mi eterno amigo en las buenas y en las malas. A mi prima Naily y a su novio Sandy, por su preocupación constante, que con su ejemplo, me han ayudado a ser mejor profesionalmente. A mi familia, abuelos, tíos, primos, amigos y a todos los que de una forma u otra me apoyaron y confiaron en mí.

RESUMEN

Como parte del desarrollo de las Tecnologías de la Informática y las Comunicaciones, la Arquitectura Orientada a Servicios (SOA, por sus siglas del inglés Services Oriented Architecture), ha propiciado un creciente auge en cuanto a desarrollo de software. Facilitar una infraestructura que permite proveer la integración de diferentes sistemas y aplicaciones, es una de las características que la hacen popular entre las tecnologías emergentes. Un aspecto importante para conseguir la integración de diferentes sistemas, es lograr que los sistemas puedan interactuar a nivel de datos. Por tal motivo, surge la necesidad de desarrollar una guía que permita facilitar la integración a nivel de datos en una SOA. En la presente investigación se propone un conjunto de pasos que incluyen técnicas, arquitectura, consejos y patrones cuyo objetivo se centra en obtener una guía de referencia que pueda ser utilizada en futuros proyectos relacionados con la integración de datos en una SOA. Con el objetivo de retroalimentar y validar la guía propuesta, fue aplicada una variante del Método Delphi, utilizando como elementos de certificación los criterios y experiencias de un grupo de expertos. La guía resultó ser apropiada, pues el 100% de los expertos consideran que puede ser efectiva si se aplicara, y de forma general fue categorizada como Bastante Adecuada para realizar la integración de diferentes aplicaciones a nivel de datos en una SOA.

Abstract

As part of the development of Information Technologies and Communications, Service Oriented Architecture has led a growing boom in terms of software development. To facilitate an infrastructure for providing the integration of different systems and applications is one of the features that make it popular with emerging technologies. An important aspect for achieving integration of different systems is to ensure that systems can interact at the data level. For this reason, the need to develop a guide that will facilitate the data level integration in an SOA. In the present study proposes a set of steps that include technical, architecture, tips and patterns whose focus is on obtaining a reference guide that can be used in future projects related to data integration in an SOA. In order to feedback and validate the proposed guide was applied a variant of the Delphi method, using as certification elements, the criteria and experiences of a group of experts. The guide turned out to be appropriate, since 100% of the experts believe it may be effective if applied, and generally was categorized as quite adequate for the integration of different data level applications in an SOA.

ÍNDICE

INTRODUCCIÓN	10
CAPÍTULO 1 “FUNDAMENTACIÓN TEÓRICA”	15
1.1. INTRODUCCIÓN	15
1.2. CONCEPTOS FUNDAMENTALES.....	15
1.3. ARQUITECTURA ORIENTADA A SERVICIOS	16
1.3.1. <i>Antecedentes de SOA</i>	16
1.3.2. <i>Razones para adoptar SOA</i>	17
1.3.3. <i>Estándares de SOA</i>	18
1.3.4. <i>Bus de Servicios Empresariales</i>	19
1.4. INTEGRACIÓN DE APLICACIONES	20
1.4.1. <i>Patrones de Transformación de mensajes</i>	21
1.4.2. <i>Patrones de acceso a datos</i>	23
1.4.3. <i>Almacenamiento de datos</i>	26
1.4.4. <i>Otras tecnologías de almacenamiento</i>	27
1.4.5. <i>Herramientas ETL</i>	28
1.4.6. <i>Interoperabilidad</i>	30
1.4.6.1. <i>Interoperabilidad Semántica</i>	31
1.5. TECNOLOGÍAS DE INTEGRACIÓN DE DATOS	31
1.5.1. <i>Tecnologías de Informatica Corporation</i>	32
1.5.2. <i>Tecnologías de IBM</i>	34
1.6. OTRAS TECNOLOGÍAS	35
1.6.1. <i>Oracle Data Integration Suite</i>	35
1.6.2. <i>Suite WSO2</i>	36
1.7. CONCLUSIONES	37
CAPÍTULO 2 “PROPUESTA DE SOLUCIÓN”	38
2.1. INTRODUCCIÓN	38
2.2. PROPÓSITO DE LA GUÍA	38
2.3. DESCRIPCIÓN DE LA GUÍA.....	38
2.3.1. <i>Revisar el proceso empresarial para confirmar la necesidad de la integración</i>	39
2.3.2. <i>Identificar las fuentes de datos</i>	40
2.3.3. <i>Identificar la información que tributará a cada fuente de datos</i>	41
2.3.4. <i>Precisar las características técnicas de cada sistema operacional</i>	42
2.3.5. <i>Conformar equipo especializado en las fuentes a integrar</i>	42
2.3.6. <i>Establecer prioridad para la integración</i>	43
2.3.7. <i>Realizar la integración por orden de prioridad</i>	43
2.3.7.1. <i>Adopción de un Modelo de Datos Canónicos</i>	45
2.3.7.1.1. <i>Aspectos a tener en cuenta en el diseño de un MDC</i>	48
2.3.7.1.2. <i>Patrones a tener en cuenta</i>	55
2.3.7.2. <i>Definir una arquitectura a nivel de datos</i>	59
2.3.7.3. <i>Selección de Herramientas</i>	61
2.3.7.4. <i>Integrar los datos</i>	62
2.4. CONCLUSIONES	62

CAPÍTULO 3 “VALIDACIÓN DE LA PROPUESTA”	63
3.1. INTRODUCCIÓN	63
3.2. MÉTODO DELPHI	63
3.3. APLICACIÓN DEL MÉTODO DELPHI	65
3.3.1. <i>Selección de los expertos</i>	65
3.3.2. <i>Elaboración del cuestionario para validar la propuesta</i>	70
3.3.3. <i>Desarrollo práctico y explotación de los resultados</i>	70
3.4. CONCLUSIONES	76
CONCLUSIONES GENERALES	77
RECOMENDACIONES.....	78
REFERENCIAS BIBLIOGRÁFICAS	79
ANEXOS	¡ERROR! MARCADOR NO DEFINIDO.

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TIC) han experimentado significativos avances, convirtiéndose en una herramienta poderosa para la automatización de los procesos sociales, destacando esencialmente la gestión empresarial. Las empresas reciben soporte tecnológico para lograr mayor eficiencia en sus procesos operativos, flexibilidad para adaptarse con rapidez a los cambios que impone el entorno y agilidad para innovar. Mundialmente se ha desatado una enorme competencia en el mercado en cuanto al desarrollo de software, exigiendo a las empresas un gran reto en la producción de sistemas y aplicaciones que le permitan ser más flexibles, que funcionen de manera más eficaz y mejoren sus procesos de negocios.

Ha sido solo cuestión de tiempo para que las empresas hayan tomado conciencia de que las decisiones de negocios, dependen en gran medida de la precisión, consistencia y transparencia de los datos. Permitiéndole acceder, mejorar la calidad y suministrar sus datos, incluyendo el tiempo real para numerosas aplicaciones. Un elemento importante para las empresas u organizaciones lo constituye el contar con una visión única y estandarizada de los datos. Pues en ocasiones existen varios sistemas que manejan información relevante y se encuentran distribuidos por las diferentes áreas de una empresa con el objetivo de prestar disímiles servicios.

Por tal motivo es que grandes empresas reconocidas mundialmente que se dedican al desarrollo de software se han enfocado en la integración de datos. Han dirigido parte de sus acciones a desarrollar plataformas globales con este fin, con el propósito de recopilar y combinar información de varias fuentes.

La tecnología de la información aplicada a las organizaciones ha vivido dos grandes hitos: el desarrollo del modelo relacional de bases de datos y la llegada de las soluciones de planificación de recursos. Antes del modelo relacional las aplicaciones definían y gestionaban su propio modelo de datos almacenando la información en ficheros externos o en soluciones más sofisticadas que utilizaban modelos de datos diversos como los jerárquicos o en red, estas circunstancias provocaban que distintas aplicaciones dentro de la misma organización tuvieran información replicada, trayendo consigo problemas de inconsistencias, consumo de recursos, repetición de tareas, entre otros. La llegada del modelo relacional y los sistemas de gestión de bases de datos relacionales, inició un proceso de extracción de datos de las aplicaciones a las bases de datos.

Tal situación marcó un gran avance tanto para la gestión de los datos de las organizaciones como para el desarrollo de aplicaciones informáticas. Pues permitía una mayor seguridad de los datos, mejor eficiencia en su tratamiento y la eliminación de inconsistencias, entre otras ventajas, mediante un punto central de gestión de los datos. Así las aplicaciones eran más fáciles de diseñar y más ligeras al no ser necesarios muchos módulos encargados de la gestión de datos.

Estas aplicaciones se relacionaban y se relacionan actualmente con la base de datos por razón de un lenguaje de consulta y de definición de datos estandarizado, el SQL (Structured Query Language). Esto trajo consigo las arquitecturas de software de dos capas, y posteriormente, surgieron los modelos de tres capas. Luego surgieron tecnologías como el Data Warehouse y la minería de datos que tenían como objetivo explotar las grandes cantidades de datos que tenían las organizaciones.

Así las organizaciones fueron creando sus modelos de datos relacionales y aplicaciones para nutrir sus almacenes de datos, pero esto provocaba falta de integración de datos dentro de una misma organización, debido a que estas aplicaciones se diseñaban y construían por áreas de negocio.

Actualmente las organizaciones están viviendo un cambio en el desarrollo de sus sistemas, de los datos a los procesos, con el objetivo de obtener arquitecturas más ágiles y flexibles que se adapten a los continuos cambios que se generan en el mercado. Son muchas las iniciativas de las TIC orientadas a proyectos de integración de datos, los cuales representan un activo crítico dentro de cualquier proceso de negocio.

La adopción de una Arquitectura Orientada a Servicios (SOA), constituye hoy una necesidad para la mayoría de las empresas, pues permite alinear las Tecnologías de la Información (TI) con los objetivos de negocio y proveer una infraestructura que soporte la integración de sistemas y responder velozmente a los cambios de requerimientos. Es una arquitectura que aporta beneficios en cuanto a rapidez para habilitar soluciones innovadoras y adaptarse a cambios en el mercado. Flexibilidad para reducir los tiempos y costos de implantación, y para contar con una arquitectura ágil que permita la evolución, cambio y crecimiento del negocio.

Actualmente el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE), ofrece soluciones basadas en una Arquitectura Orientada a Servicios, proporcionándole a las empresas una infraestructura técnica bastante ágil que permite la integración de sistemas en un menor tiempo y costo. Sin embargo, en sus soluciones resulta complicado el tema relacionado con la integración de sistemas a

nivel de datos, pues generalmente, en una organización existen varios sistemas o aplicaciones de uso interno que responden a las diferentes áreas de la misma, con el objetivo de prestar diferentes servicios a sus empleados. Pero la mayoría fueron desarrollados de manera independiente o por otras empresas, por lo que cada sistema recopila información en distintos formatos y ubicaciones. Como consecuencia se generan problemas como por ejemplo, no existe una visión unificada de la información, existen incompatibilidades entre los datos, ya sea por su estructura, por el modelo que emplean o por el formato en que se encuentran. Además poseer en las empresas variedad de entornos y diferentes tipos de estándares de información, conllevan inevitablemente a varias incongruencias a nivel de datos tornándose complicado la integración, y por tanto las empresas sufren de pérdidas monetarias.

A raíz de dichas situaciones, el centro no se encuentra completamente capacitado para enfrentar soluciones relacionadas con la integración de datos en una Arquitectura orientada a Servicios. Provocando que no exista la organización adecuada a la hora de llevar a cabo un proyecto de tal envergadura. Además no se tienen en cuenta algunos elementos necesarios y consiguientemente se realizan determinadas acciones innecesariamente al no disponer de una guía que permita facilitar la integración de datos en el proceso de adopción de una SOA.

Por tales razones la investigación se plantea como **problema de la investigación** la siguiente interrogante ¿Cómo facilitar el desarrollo de soluciones de integración a nivel de datos para sistemas informáticos basadas en una Arquitectura Orientada a Servicios?

A partir del problema planteado, el **objeto de estudio** se centra en la integración de Sistemas Informáticos, teniendo como **campo de acción** la integración a nivel de datos para una Arquitectura Orientada a Servicios.

El **objetivo general** de la presente investigación consiste en diseñar una guía de integración de datos para una arquitectura SOA. Derivándose del objetivo general los **objetivos específicos** son:

- Elaborar el marco teórico sobre el diseño de una guía de integración de datos para una arquitectura SOA.
- Identificar los principales tecnologías relacionadas con la integración de sistemas a nivel de datos en el mundo.
- Diseñar una guía que permita a las empresas mayor facilidad para integrar sus sistemas

informáticos a nivel de datos.

- Validar la propuesta realizada mediante la aplicación del método Delphi.

Los **resultados esperados** con esta investigación es la obtención de una guía que facilite a las empresas la integración de sus sistemas informáticos a nivel de datos. Que se utilice como modelo de referencia para la realización de futuros proyectos ya sean de carácter nacional o internacional relacionados con la integración de datos en una SOA.

La **idea a defender** con la presente investigación es la obtención de una guía que facilitará al CDAE el desarrollo de soluciones de integración a nivel de datos basadas en una Arquitectura Orientada a Servicios y por tanto se podrán minimizar el número de errores durante el desarrollo de proyectos relacionados con dicha temática.

Para alcanzar los objetivos propuestos se hacen uso de diferentes **Métodos de Investigación** como son:

Métodos Teóricos:

Análisis histórico – lógico: Se utiliza con el objetivo de analizar e investigar los orígenes y evolución de la integración de datos, para lograr una mejor comprensión de las especificidades del tema.

Analítico – sintético: Investigar los principales problemas existentes en la integración de datos, para darle solución a los mismos.

Métodos empíricos:

Consulta a expertos: Permitirá obtener información necesaria para la investigación, mediante la experiencia que poseen los expertos.

Entrevista: consistirá en realizar diferentes entrevistas a personal calificado en la temática, pertenecientes al Centro de Consultoría y Desarrollo de Arquitecturas Empresariales u otros centros, con el objetivo de obtener información calificada que contribuya a la realización de la guía.

El trabajo de diploma estará estructurado en 3 capítulos.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA: Se establece el diseño teórico de la investigación, donde se examinan los principales conceptos que serán manejados durante el desarrollo de la tesis. Se realiza el

estado del arte a nivel nacional e internacional para identificar los principales problemas que presentan las aplicaciones a nivel de datos al efectuarse la integración de distintos sistemas informáticos.

CAPÍTULO 2. ELABORACIÓN DE LA GUÍA PROPUESTA: Se definen pasos para conformar una guía que permita facilitar la integración de datos para una arquitectura SOA, que se deberán aplicar para solucionar los diferentes problemas que van apareciendo durante la integración, y consiguiente a esto se realiza la propuesta de solución donde se describe dicha guía.

CAPÍTULO 3. VALIDACIÓN DE LA GUÍA: Se enfoca en la validación de la guía propuesta donde se aplica un método de expertos que permita evaluar la investigación, obteniéndose los resultados de la evaluación efectuada a través de encuestas realizadas.

CAPÍTULO 1 “FUNDAMENTACIÓN TEÓRICA”

1.1. Introducción

En el presente capítulo se expone una visión general de los conceptos más significativos y aspectos fundamentales concernientes a la elaboración de una guía para facilitar la integración de los datos en una Arquitectura Orientada a Servicios. Se analizará el estado actual de la temática en el mundo y sus principales tecnologías.

1.2. Conceptos fundamentales

Teniendo en cuenta que la presente investigación incluye la elaboración de una guía para facilitar la integración de diferentes aplicaciones a nivel de datos en una Arquitectura Orientada a Servicios, se hace necesario conocer los principales elementos relacionados con la misma.

Guía: Se basa en indicar, enseñar, conducir, dirigir, orientar, aconsejar y encaminar con el objetivo de listar u organizar datos o informaciones referentes a determinada materia. Constituye una herramienta analítica que tiene como fin facilitar información y garantizar un fácil aprendizaje sobre una actividad específica, ayudando a organizar el tiempo y a encontrar lo esencial de cada tema.

Integración de datos: La integración de datos consiste en el proceso de almacenar y combinar los datos de diversas fuentes con el fin de proporcionar una visión única y estandarizada de los datos. Puede tener diferentes escenarios o gran variedad de situaciones, por ejemplo cuando dos compañías necesitan combinar sus bases de datos, o cuando se necesitan integrar diferentes aplicaciones dentro de una misma empresa.

Arquitectura Orientada a Servicios: La Arquitectura SOA establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios. La forma más habitual de implementarla es mediante Servicios Web, una tecnología basada en estándares e independiente de la plataforma, con la que SOA puede descomponer aplicaciones monolíticas en un conjunto de servicios e implementar esta funcionalidad en forma modular". [1]

1.3. Arquitectura Orientada a Servicios

Es una arquitectura de software que tiene mucho futuro en dicha industria y a lo que tecnología de la Información se refiere, ya que permitirá realizar lo que antes se veía como algo imposible, como es la integración de sistemas. Muchas empresas han hecho grandes inversiones para mejorar la gestión de sus recursos a lo largo de muchos años, debido a que tienen una enorme cantidad de datos almacenados en sus sistemas. En este sentido, la Arquitectura Orientada a Servicios permite ayudar a las organizaciones, agilizando los procesos para que puedan hacer negocios de manera más eficiente y en menor tiempo, permitiendo adaptarse a las cambiantes necesidades y la competencia del mercado actual.

De tal forma se puede definir SOA como:

“Una arquitectura que básicamente está orientada a un conjunto de servicios tanto de negocio como tecnológicos que interactuando entre sí nos proporcionan la lógica necesaria para construir aplicaciones de una manera rápida y cumpliendo con los principios de la Orientación a Servicios. Además proporciona una serie de guías y recomendaciones para conseguir los objetivos que se impone una organización a la hora de desarrollar aplicaciones. Surge de la necesidad de mejorar la integración e interoperabilidad entre aplicaciones; cuyo fin o propósito es la convergencia entre el desarrollo e integración que minimice las imitaciones entre aplicaciones”. [2]

A partir del análisis de varios conceptos y un estudio realizado se puede definir SOA como una arquitectura de software que permite a las instituciones integrar diferentes sistemas o aplicaciones, mediante la utilización de servicios y un conjunto de estándares.

Un servicio es una función de aplicación empaquetada como un componente reutilizable para ser usado en un proceso de negocio [3]. Una forma de implementar una SOA es mediante los servicios web, los cuales son aplicaciones modulares que pueden ser publicadas, localizadas e invocadas desde cualquier ubicación en la web o en una red local. El suministrador y el consumidor del servicio web no tienen que preocuparse por el Sistema Operativo, idioma o modelo de componentes usado para crear o acceder a los servicios, ya que están basados en estándares omnipresentes de internet, como XML, HTTP y SMTP. [4]

1.3.1. Antecedentes de SOA

Durante años las empresas han tenido la necesidad de gestionar sus recursos, lo que ha incidido en que

busquen diferentes soluciones para lograrlo. Una de las primeras vías utilizadas fueron las llamadas infraestructuras, las cuáles eran eficaces en la gestión de recursos, propiciando la creación de numerosas aplicaciones en las empresas. Sin embargo, las aplicaciones no eran diseñadas con el fin de integrarse, trayendo consigo problemas de interoperabilidad.

El desarrollo de las TIC ha obligado a las instituciones a crear aplicaciones de mayor complejidad, utilizando menos tiempo y presupuesto del que estaban acostumbrados a emplear, implicando en la mayoría de los casos el uso de funcionalidades que ya se encontraban implementadas como parte de otros sistemas, o sea reutilizando la funcionalidad requerida, lo cual no era algo tan fácil debido a que estas aplicaciones no habían sido diseñadas con el fin de integrarse. Se optó entonces por implementar la funcionalidad que se necesitaba, desde el principio como si esta no existiera lo que implicaba mayor tiempo de desarrollo y algunos problemas como funcionalidades replicadas en diferentes sistemas.

Es por ello que surge la Arquitectura Orientada a Servicios, la cual garantiza a las organizaciones de hoy en día gran agilidad de los negocios, considerándose un aspecto de fundamental importancia para el éxito en el presente mercado mundial.

1.3.2. Razones para adoptar SOA

La Arquitectura Orientada a Servicios permite solucionar numerosos inconvenientes de software que hoy en día presentan las aplicaciones, facilitando y estandarizando la integración de sistemas, mediante la interoperabilidad de la información, aplicaciones y requerimientos de los procesos de negocio, permitiendo mayor flexibilidad y reutilización de estos procesos, proporcionando mayor rapidez de desarrollo y un menor presupuesto. Por lo que constituye un marco de trabajo que permite a las organizaciones alinear los objetivos de negocio con la infraestructura de las tecnologías de la información, integrando los datos y la lógica de negocio de diferentes sistemas.

Existen diversas razones para que una organización adopte un enfoque SOA, entre ellas se pueden destacar:

Reutilización: Es el factor fundamental en el cambio a SOA. Las funciones de negocio, dentro de una empresa, pueden ser expuestas como servicios web y ser reutilizadas para cubrir nuevas necesidades de negocio.

Escalabilidad: Como los servicios de SOA están débilmente acoplados, las aplicaciones que usan esos servicios escalan fácilmente, debido a que existe muy poca dependencia entre las aplicaciones clientes y los servicios que usan.

Flexibilidad: Es otra de las características que proporciona el acoplamiento débil entre los servicios. Cualquier cambio en la implementación de uno de ellos no afectaría al resto siempre que se mantenga la interfaz.

Eficiencia de coste: Las arquitecturas SOA se basan en la exposición de servicios ya existentes para ser reutilizados. Al usar servicios web para exponer estos servicios, se reutiliza la infraestructura web existente en virtualmente todas las organizaciones por lo que se limita considerablemente el coste. [5]

1.3.3. Estándares de SOA

Uno de los principios fundamentales de una Arquitectura Orientada a Servicios, es que es basado en estándares, a continuación se presentan los mismos.

UDDI (del inglés Universal Description Discovery and Integration): Define los mecanismos para publicar servicios en un registro. Permite a los consumidores comprobar que servicios están disponibles, así como encontrar su localización física. [6]

Define tres elementos:

- Una especificación técnica para la construcción de un directorio distribuido de servicios.
- Un lenguaje XML para el almacenamiento de los datos.
- Un API estándar (basada en SOAP) para guardar y recuperar información de los registros UDDI.

WSDL (del inglés Web Service Description Language): Es un lenguaje XML para describir los servicios en la arquitectura SOA [6]. Permite establecer una comunicación con los servicios web.

SOAP (del inglés Single Object Access Protocol): Es un protocolo de mensajería para el intercambio de información entre sistemas [6]. Además define como dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

XML (del inglés Extensible Markup Language): Lenguaje para establecer el formato estándar para los datos que se van a intercambiar.

1.3.4. Bus de Servicios Empresariales

Dentro del ambiente SOA, uno de los componentes de mayor importancia es el Bus de Servicios Empresariales (ESB). A continuación se mostrarán algunos conceptos asociados a dicho componente.

“El Bus de Servicios Empresariales es una infraestructura de software que facilita la integración de aplicaciones. Una arquitectura de software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones brindando brinda servicios fundamentales para las arquitecturas más complejas”. [7]

“Constituye un elemento de software, un middleware, una infraestructura basada en estándares, que proporciona servicios para la construcción de arquitecturas más complejas basadas en eventos y en un motor de mensajería. Permite la integración de aplicaciones de forma rápida, directa y basada en estándares. Es una suite de productos independientes de la infraestructura que facilita el procesado, la transformación de datos, el enrutamiento y la orquestación de procesos usando servicios web”. [8]

“Consiste en una solución de integración distribuida, basada en los mensajes y en estándares abiertos. La función de un ESB es proporcionar una comunicación fiable entre los distintos recursos tecnológicos tales como aplicaciones, plataformas y servicios, que están distribuidos en múltiples sistemas por toda la empresa”. [9]

Considerando lo anteriormente abordado, básicamente el ESB es una infraestructura de software basada en estándares, que permite la integración de aplicaciones de forma rápida a través de la gestión de servicios.

Su creación permite satisfacer la demanda y evitar los problemas comunes que generaban las plataformas de Integración de Aplicaciones Empresariales (EAI). Todas las aplicaciones integradas debían trabajar a través de un corredor, creando un punto único de fallo, proporcionado un riesgo para los sistemas de negocios complejos. Sin embargo, el ESB cuenta con numerosos corredores, evitando así este riesgo. Se puede implementar de forma rápida y poco costosa, por lo que es a menudo favorecido por los desarrolladores que trabajan con plazos muy ajustados y presupuestos limitados. El mismo, puede tener una visión a largo plazo y soportar una amplia estrategia SOA. [7]

A medida que los departamentos de TI se centran cada vez más en el diseño de SOA para reducir los costes de desarrollo y para aumentar la agilidad del negocio, los ESB se están convirtiendo en un paso clave para el establecimiento de una SOA empresarial. Constituyen los cimientos de una SOA y pueden

complementarse con capacidades de productividad adicionales, como la orquestación de servicios y los registros. [9]

Desde una perspectiva SOA, un ESB constituye un elemento esencial para lograr una infraestructura fundamentada en servicios SOA.

1.4. Integración de aplicaciones

Una solución de integración necesita que los sistemas transmitan, compartan información entre ellas. A pesar de las diferentes plataformas operativas, lenguajes de programación, y formatos de datos, dicha solución debe ser capaz de interactuar con las diversas tecnologías asociadas a los sistemas involucrados. [10]

Existen varios estilos para realizar la integración de aplicaciones, los mismos son:

Transferencia de archivos: Ocurre cuando una aplicación exhibe un archivo y luego otra lo consume. Con tal propósito, las aplicaciones necesitan establecer los mismos parámetros en cuanto al nombre del archivo, la ubicación, el formato, la fecha de cuándo será expuesto, consumido y eliminado. [10]

Base de datos compartida: Varias aplicaciones comparten el mismo esquema de base de datos, a partir de la utilización de una base de datos común. Al utilizar el mismo esquema, no hay almacenamiento de datos duplicados, por lo que no necesita transferirse información de una aplicación a otra. [10]

Invocación a procedimiento remoto: Una aplicación expone algunas de sus funcionalidades para que otras puedan acceder remotamente y consuman de las mismas. La comunicación se produce en tiempo real y de forma sincrónica. [10]

Mensajería: Una de las aplicaciones publica un mensaje en un canal de mensajes común. Posteriormente otras aplicaciones pueden utilizarlo, utilizando el mismo canal y formato del mensaje. La comunicación ocurre asincrónicamente. [10]

De los estilos expuestos, la mensajería requiere gran valor para la presente investigación. Su importancia radica en que a partir de un conjunto de patrones permite establecer la comunicación asincrónica y el intercambio entre varias aplicaciones de manera flexible y eficiente. Es una tecnología que permite alta velocidad y la entrega confiable de paquetes de datos, mediante el uso de un canal de mensajes que pueden utilizar simultáneamente varias aplicaciones.

En ocasiones, las aplicaciones interesadas en la integración utilizan diferentes formatos de datos, con tal propósito, la mensajería ofrece una solución general para eliminar dichas diferencias. La solución es a través de los patrones de transformación de mensajes.

1.4.1. Patrones de Transformación de mensajes

Muchos de los principios incorporados a los patrones de transformación en la mensajería son aplicables a otros estilos basados en la integración. Por ejemplo, la transferencia de archivos tiene que realizar funciones de transformación entre los sistemas. Del mismo modo, la invocación a procedimiento remoto tiene que realizar las solicitudes en el formato de datos especificado por el servicio que se va a llamar, incluso si el formato interno de la aplicación es diferente. [10]

Los patrones de transformación de mensajes que propone la mensajería son:

Envelope Wrapper o Enrollador de sobre: tiene como propósito encapsular el mensaje para que sea transportado. Realiza el empaquetado de un mensaje asegurándose de que el mismo sea compatible con la infraestructura de mensajería. Al efectuar el empaquetado pueden ser añadidos algunos requisitos de seguridad. Cuando el mensaje llega a su destino, para que el mismo sea entendible, se revierte el proceso de enrollado, obteniéndose el mensaje original. [10]

Content Enricher o Enriquecedor de contenido: en ocasiones al sistema solicitante de servicios, le llegan los mensajes faltándole algunos elementos. Cuando ocurre una situación así, el enriquecedor de contenido utiliza la información obtenida del interior del mensaje, para recuperar los datos que faltan de una fuente externa. Para realizar la búsqueda utiliza los campos claves de la información obtenida. Posteriormente agrega los datos recuperados al mensaje, provocando un aumento del mismo a partir de la información adquirida. [10]

Content Filter o Filtro de contenido: genera el efecto contrario del enriquecedor de contenido, o sea, elimina elementos de datos de un mensaje. Es utilizado con el propósito de simplificar el manejo de mensajes, reducir el tráfico de red, cuestiones de seguridad y simplificar la estructura del mensaje. A partir de su aplicación un mensaje puede resultar más significativo, al suprimirse varios campos que sean redundantes e irrelevantes para el receptor. Proporcionado el uso de los elementos que realmente se necesitan. [10]

ClaimCheck o Chequeo de reclamo: se utiliza en el caso que no se quieren eliminar por completo los

elementos de datos, el chequeo de reclamo puede ser muy útil, debido a que solo permite eliminar la información temporalmente. De manera que si se necesitara esta nuevamente, se podrían recuperar, pues los datos eliminados podrían guardarse en un almacén de datos o fichero. [10]

Normalizer o Normalizador: se emplea para aislar el resto del sistema de la variedad de formatos de mensajes entrantes, para ello utiliza un conjunto de canales por cada tipo de mensaje y cada canal se conecta a un traductor personalizado con el propósito de que los mensajes tengan un formato común. El inconveniente de este enfoque es que un gran número de formatos de mensaje se traduce en un número igual de canales de mensajes. [10]

Modelo de Datos Canónicos (MDC): constituye una forma de representación de información común, producida y consumida por aplicaciones, con el propósito de publicar eventos en forma de mensajes salientes de una aplicación hacia otras aplicaciones. [11]

Utilizar correctamente un MDC puede proporcionar grandes beneficios en un entorno SOA, tales como el acoplamiento flexible de aplicaciones, facilidad de mantenimiento de integración y una comprensión común de la información [11]. Su uso permite que las aplicaciones se comuniquen en un solo idioma, facilitando la comunicación entre ellas.

Se encuentra compuesto por el modelo de datos conceptual, modelo de datos lógico, el modelo de análisis de servicios y el modelo de mensaje canónico. Un **modelo de datos conceptual** es un modelo de datos canónico visto al más alto nivel de abstracción. Permite representar los requisitos de información estratégica de la empresa en el ámbito del proyecto SOA y por lo general incluye sólo las entidades y sus relaciones más importantes. [11]

El **modelo lógico de datos** se basa en el modelo de datos conceptual. Permite especificar más al detalle la relación existente entre las entidades y sus atributos. Los atributos incluidos deben ser suficientes para apoyar todos los aspectos de las estructuras de información que se especifica, debido a que se les asigna un tipo de datos físico. [11]

El **modelo de análisis de los servicios** representa las estructuras de datos para el servicio y la especificación de los componentes. Las estructuras expresadas en el modelo de análisis de servicio son un reflejo directo de lo que se encuentra en el modelo conceptual. [11]

El **modelo de mensaje canónico** representa el formato estándar utilizado para intercambiar información

de negocios en un bus de servicio empresarial. Proporciona los formatos de intercambio utilizados, para que todos los componentes sólo necesiten saber su propio formato de datos y el formato de datos por defecto utilizado en el bus de servicio. [11]

Es basado en el modelo lógico de datos, pero más normalizado con el fin de reducir la complejidad de los gráficos de las relaciones y a su vez los mensajes de servicio sean más fáciles de entender por los consumidores de servicios. [11]

La representación más común utilizada por dicho modelo es un conjunto de esquemas XML, permitiendo utilizar este tipo de mensajes. El modelo de mensaje canónico consta de [11]:

- 1- Un conjunto definido de elementos y atributos que representan a las entidades empresariales y sus atributos utilizados en los mensajes. Cada definición incluye:
 - Tipos de datos técnicos, formatos, estructuras y nombres.
 - Las normas que rigen los tipos de valores permitidos.
 - La semántica de los tipos de negocio.
- 2- Un conjunto definido de mensajes, donde cada uno incluye un conjunto relacionado de los tipos definidos anteriormente, así como los elementos y atributos estructurados para proporcionar un documento de negocios con un significado específico.

En resumen proporciona un conjunto de puntos de vista sobre el modelo de datos canónico, exponiendo una pequeña parte de las entidades relacionadas en forma de mensaje reutilizable, pero siempre siguiendo la entidad y estructuras de relación que se define en el modelo de datos canónico. [11]

Sin embargo, existen otros patrones que influyen en como manipular, controlar y acceder a los datos, así como mantener la seguridad de los mismos. A continuación se estarán presentando algunos patrones de acceso a datos que pudieran ser utilizados en la propuesta de solución.

1.4.2. Patrones de acceso a datos

Move Copy of Data o Movimiento de la Copia de Datos: se aplica cuando se desea que otras aplicaciones tengan acceso a la información pero sin tener acceso a la fuente original de los datos. Se soluciona proveyendo acceso a los datos mediante mecanismos de replicación y sincronización a otra fuente que puede ser accedida por las demás aplicaciones consumidoras. [12]

Los beneficios de aplicar el patrón son [12]:

- Optimización de la base de datos destino para el fin que fue creado.
- Autonomía de los datos. La base de datos destino es responsable únicamente por los datos que esta contiene.
- Privacidad de los datos. La base de datos destino no tiene conocimiento de los detalles de la base de datos fuente.
- Seguridad. La información de la base de datos fuente no es accedida por la base de datos destino.

Los riesgos que presentan son [12]:

- Incremento en la complejidad de administración de los datos.
- Impacto en performance al momento de transmitir los datos al destino. Debido a que la transferencia debe de programarse cuando la aplicación no está siendo utilizada por usuarios.
- Seguridad. La base de datos destino debe de ser consistente con las políticas de seguridad de los datos implementadas en la base de datos fuente.

Data Replication o Replicación de Datos: se utiliza cuando se desea implementar el patrón anterior, y adicionalmente, permitir que la información involucrada sea actualizada por las aplicaciones tanto en la base de datos fuente y destino. Así se manejaría la integridad de dichos cambios en ambas fuentes de datos. La solución está en implementar el patrón de Move Copy of Data, pero añadiendo características propias de replicación de datos. Los beneficios y riesgos de aplicar este patrón coinciden con el patrón analizado anteriormente. [12]

Master - Master Replication: Es utilizado cuando se necesita implementar una replicación entre una fuente y un destino, los requerimientos son los siguientes [12]:

- El grupo de datos a replicar es actualizable tanto en la fuente como en el destino.
- Las actualizaciones necesitan ser transmitidas tanto como a la fuente como al destino.
- Conflictos necesitan ser detectados y resueltos.

La replicación necesita de la implementación de dos enlaces de replicación entre la fuente y el destino bidireccionalmente. Durante la transmisión se debe de tener especial cuidado antes de obtener y

manipular la información que es escrita en el destino. En vista que el destino también puede ser escrito y actualizado por las aplicaciones, los datos del destino podrían haber cambiado durante la última transmisión. Un conflicto ocurre cuando los datos de una replicación son también actualizados por el destino durante la transmisión. Antes de escribir la unidad de datos a replicar al destino, los conflictos deben de ser detectados y resueltos. Los conflictos pueden pertenecer a una de las siguientes categorías [12]:

- Conflictos de actualización: ocurren cuando los datos del destino que van a ser actualizados o eliminados fueron actualizados luego de la última transmisión.
- Conflictos de eliminación de datos: ocurren cuando los datos del destino que van a ser actualizados o eliminados fueron eliminados luego de la última transmisión.
- Conflictos de negocio: ocurren cuando los datos pueden ser transmitidos al destino, pero la integridad global de la base de datos destino es corrompida por los datos escritos.

Como beneficio, aporta que puede ser utilizado para efectos de respaldo y backup. Permite que en caso de desastre en una de las bases de datos, solo los cambios realizados en la misma sean afectados. Sin embargo presenta varias limitaciones y riesgos, como son [12]:

- Utiliza métodos de resolución consistente de conflictos. Los dos enlaces de replicación deben de acordar en tener los mismos resultados.
- Necesita vigilancia en la resolución de conflictos. Deben de implantarse mecanismos de monitoreo para tener una trazabilidad de las resoluciones de conflictos realizadas.

Por tanto constituye un patrón a tener en cuenta. Debido a que puede ser utilizado con el propósito de mantener constantemente una copia actualizada de la información importante que se maneja en las empresas. En consecuencia se debe crear una nueva base de datos, y realizar la replicación. En caso de que la información en una de las bases de datos fuera dañada por disímiles razones, no traería aparejado un gran problema. Solo habría que actualizar la base de datos dañada, por la otra que se encuentra actualizada.

Los patrones abordados no son los únicos. Existen otros que de igual forma son considerados patrones de acceso a datos. Sin embargo a partir de la investigación realizada, fueron considerados los descritos anteriormente, pues son los que de manera general pudieran ser utilizados.

Otro elemento importante en la integración de aplicaciones lo constituye el tema relacionado con el almacenamiento de los datos. Los datos representan los hechos y la información acerca de algo, mientras que el almacenamiento de estos, constituye un proceso con una arquitectura periódica y coordinada de la copia de numerosas fuentes en un entorno optimizado capaz del procesamiento analítico e informativo. [13]

1.4.3. Almacenamiento de datos

Un componente importante dentro de este ámbito son las bases de datos, las cuáles ayudan a los sistemas a manejar de manera precisa diferentes volúmenes de información.

Se define una Base de Datos como una serie de datos organizados y relacionados entre sí, y un conjunto de programas que permitan a los usuarios acceder y modificar los mismos. [14]

Las bases de datos constituyen un conjunto de datos organizados y estructurados que siguen un modelo de información que refleja las relaciones que existen entre ellos. Además se diseñan con un objetivo específico y debe ser organizada con una lógica coherente.[15]

Tradicionalmente se ha hecho una distinción clara entre dos tipos de bases de datos:

Bases de datos referenciales: son aquellas que ofrecen registros que a su vez son representaciones de documentos primarios, y entre las que cabe distinguir, las bibliográficas, cuyo contenido son registros de tipo bibliográfico y los directorios, aquí el contenido está referido a la descripción de otros recursos de información, como por ejemplo un directorio de bases de datos.[15]

Bases de datos fuente: son las que ofrecen el documento completo, no una representación del mismo, y entre ellas se pueden encontrar las numéricas que contienen la información de tipo numérico, las textuales, contienen el texto completo de un documento mientras que las mixtas combinan ambos tipos de información. [15]

Actualmente para el almacenamiento efectivo de la información, las bases de datos constituyen un recurso inevitable. Sin embargo no es la única manera de almacenar los datos, existen otras tecnologías que ayudan a las empresas en este sentido.

1.4.4. Otras tecnologías de almacenamiento

Una de las tecnologías utilizadas para el almacenamiento de datos, son los Data Warehouse o almacén de datos. Son varias las definiciones que existen que se refieren al término relacionado, los cuales constituyen una de las principales tecnologías en un contorno donde se necesite que la información utilizada sea gestionada de manera óptima y adecuada.

En el contexto de la informática, es una colección de datos orientada a un determinado ámbito (empresa, organización, entidad), que ayuda a la toma de decisiones en la que se utiliza. Se trata de un expediente completo de una organización, más allá de la información transaccional y operacional, almacenada en una base de datos diseñada para favorecer el análisis y la divulgación eficiente de datos. [16]

Un Data Warehouse es una base de datos corporativa que se caracteriza por integrar y depurar información de una o más fuentes distintas, para luego procesarla permitiendo su análisis desde infinidad de perspectivas y con grandes velocidades de respuesta. [17]

Proporciona una visión global, común e integrada de los datos de la organización, independiente de cómo se vayan a utilizar posteriormente por los consumidores o usuarios. Las bases de datos relacionales son el soporte técnico más comúnmente usado para almacenar las estructuras de estos datos y sus grandes volúmenes. [18]

Entre las funciones que realiza se encuentra contener los datos que son necesarios o útiles para una organización, o sea, que se comporta como un repositorio de datos para posteriormente transformarlos en información útil para el usuario. Además debe entregar la información correcta en el momento óptimo y en el formato adecuado al destino indicado. [19]

Habitualmente en un almacén de datos se guarda información histórica que cubre un amplio período de tiempo. Sin embargo en determinadas ocasiones no se necesita íntegramente esta información, sino sólo sus últimos valores. Para tal situación, lo que se utiliza es un almacén de datos operacional conocido como ODS (del inglés Operational Data Store). [18]

Un ODS es un tipo de base de datos que se utiliza frecuentemente como un área intermedia lógica para un almacén de datos. Mediante el mismo, los datos pueden ser lavados, así como analizar si cumplen con las reglas de negocio correspondientes. Se utiliza para realizar operaciones con los datos antes de ser trasladado al almacén de datos para el almacenamiento a largo plazo. [20]

Otra de las tecnologías que se pueden encontrar relacionadas con el almacenamiento de datos son los Datamart. Los cuáles se encuentran estrechamente relacionados con los Data Warehouse.

Es una base de datos departamental, especializada en el almacenamiento de los datos de un área de negocio específica. Se caracteriza por disponer la estructura óptima de datos para analizar la información al detalle desde todas las perspectivas que afecten a los procesos en un departamento. Un Datamart puede ser alimentado desde los datos de un Data Warehouse, o integrar por sí mismo un conjunto de distintas fuentes de información. [21]

Se puede entender un Datamart como un subconjunto de los datos del Data Warehouse con el objetivo de responder a un determinado análisis, función o necesidad y con una población de usuarios específica. [18]

La diferencia entre ellos es que el Datamart está pensado para cubrir las necesidades de un grupo de trabajo o de un determinado departamento dentro de la organización. Es el almacén natural para los datos departamentales. En cambio, el ámbito del Data Warehouse es la organización en su conjunto. Es el almacén natural para los datos corporativos comunes. [18]

Dentro de la integración de datos uno de los elementos que no puede dejarse pasar por alto es el uso de herramientas que permitan manejar y gestionar los datos de una manera eficaz y sistemática. Un ejemplo de esto son las llamadas herramientas ETL.

1.4.5. Herramientas ETL

ETL es el proceso que organiza el flujo de los datos entre diferentes sistemas en una organización y aporta los métodos y herramientas necesarias para mover datos desde múltiples fuentes a un almacén de datos, reformatearlos, limpiarlos y cargarlos en otra base de datos o bodega de datos. [22]

Es una de las tecnologías más antiguas de integración a nivel de datos, y junto con las Bases de Datos Relacionales han ido aportando beneficios con el tiempo. Incluso en la actualidad dentro de la Arquitectura Orientada a Servicios tiene su puesto en lo que son los “Servicios de Información”. También se pueden usar para el desarrollo de funcionalidades específicas que involucran la integración de entidades que contienen información distinta. Tiene como propósito el intercambio de información entre diferentes almacenes de datos y está orientada a los procesos de datos, basados en repositorios como bases de datos y archivos. Las herramientas de este tipo permiten extraer, transformar y cargar información desde diferentes tipos de fuentes de datos. Por ejemplo:

Extraer: Permite obtener la información de distintas fuentes de datos, principalmente de las bases de datos. Así como extraer datos de archivos como XML, Excel, CSV, Texto, entre otros.[23]

Transformar: Implica la aplicación de una serie de reglas o funciones a los datos extraídos. Los procesos más comunes que se utilizan para la transformación es la conversión, la limpieza de los datos duplicados, la estandarización, filtrar, ordenar, traducir o verificar si las fuentes de datos son inconsistentes.[13]

Cargar: Constituye la última etapa del proceso de ETL y permite guardar la información procesada en bases de datos y archivos de distintos tipos.

Entre los principales objetos que maneja un ETL se pueden encontrar las tablas de las bases de datos, los procedimientos almacenados, archivos XML, archivos de texto, archivos Excel y transformaciones de datos. [23]

Otra característica que ofrecen actualmente los ETL, para la Arquitectura SOA, es que los procesos ETL se pueden ejecutar como servicios web, para lo cual el motor de ETL publica el proceso en formato SOAP, pudiendo ser ejecutado desde cualquier plataforma (Java, .Net), y ser integrado en tecnologías como ESB. [23]

En el contexto de SOA se podría indicar que ETL es la herramienta que permite diseñar y construir servicios de información basados en actividades automatizadas a nivel de datos. (15)

Existen numerosas herramientas y aplicaciones ETL en el mercado entre ellas las más reconocidas son:

- BM Websphere DataStage.
- Business Objects Data Integrator (BODI).
- SAS ETL Studio.
- Oracle Warehouse Builder.
- Informatica PowerCenter.
- Cognos Decisionstream.
- Ab Initio Microsoft SQL Server Integration Services (SSIS).

Algunas libres son:

- Jitterbit.
- Talend Open Studio.
- Scriptella ETL Project
- Pentaho Data Integration.

1.4.6. Interoperabilidad

La integración de sistemas y la información que estos contienen, ha sido un problema que ha tenido a lo largo de los años numerosos intentos por tratar de resolverse. Los datos representan un activo importante para cualquier empresa debido a que estos interactúan directamente con los procesos. El avance de las nuevas tecnologías de la información ha influido en el desarrollo de nuevos sistemas o aplicaciones informáticas más complejos que deben manejar grandes cantidades de datos. Se hace necesario que las empresas despierten en cuánto a este tema lo cual garantizaría trabajar con los datos de manera más rápida, limpia y precisa. Dada la situación existente, se hace difícil la extracción de toda la información necesaria, pero gracias a las nuevas tecnologías de integración, los sistemas informáticos son capaces de comunicarse y compartir datos mediante la utilización de un lenguaje común. Aquí juega un papel muy importante la interoperabilidad, una característica imprescindible dentro del ámbito de la integración de sistemas, la cual se define como:

La capacidad de los sistemas de tecnologías de la información y las comunicaciones, y de los procesos de negocios que ellas soportan, de intercambiar datos y posibilitar compartir información y conocimientos [24]. Permitiendo que la información entre diferentes sistemas se integre, de manera consistente y segura, mediante la adopción de estándares, tecnologías, arquitecturas y herramientas tecnológicas.

Presenta dimensiones de diferentes naturalezas como son la organizativa, la semántica y la técnica. La interoperabilidad organizativa contempla la modelización de los procesos y la colaboración entre las administraciones. La semántica no sólo posibilita que los recursos de información puedan estar conectados, sino que también la información pueda ser interpretable de forma automática y consecuentemente reutilizable por aplicaciones informáticas que no intervinieron en su creación. La técnica es la forma más directa de interconexión de aplicaciones a través de diversos componentes tecnológicos. [25]

1.4.6.1. Interoperabilidad Semántica

La interoperabilidad semántica constituye un aspecto fundamental dentro de la interoperabilidad, por lo que la presente investigación abordará sus principales elementos.

De manera general, la interoperabilidad semántica se ocupa del significado en el uso de los datos y la información y, en concreto, garantiza que el significado preciso de la información intercambiada pueda ser entendido por cualquier aplicación. Para ello, habilita a los sistemas para combinar la información proveniente de otras fuentes y procesarla de una manera integrada y con el sentido adecuado. [26]

Trata de asegurar el significado de los datos para que puedan ser comprendidos sin ambigüedades por diferentes aplicaciones, y esa información pueda ser procesada de una manera significativa. Se refiere a la posibilidad de intercambiar información que diseñada en diferentes lenguajes, con diferente significado o para distintos propósitos, aparezca con un único significado, asimilable y comprensible de forma precisa por los usuarios. Sin la semántica, los datos son simplemente cadenas de binarios sin sentido, los consumidores y proveedores de servicios podrían malinterpretar y corromper los datos y, finalmente, traer efectos no deseados.

1.5. Tecnologías de integración de datos

Con el objetivo de satisfacer a sus trabajadores y clientes, las empresas se han visto en la necesidad de mejorar sus procesos de negocio, de crear nuevos servicios y productos mejorados y de mayor calidad. Como parte del desarrollo informático alcanzado las empresas han creado sistemas o aplicaciones más complejos que necesitan manejar mayores cantidades de información, utilizar información o servicios de otros sistemas. A razón de tal situación, se ha dado la tendencia de integrar sistemas sobre todo a nivel de datos, influenciando en el desarrollo de las tecnologías de integración de datos.

Se ha llegado a una etapa en la que se puede hablar de tecnología de integración de datos, debido a que numerosas empresas líderes en el desarrollo de software han dado un paso importante en este sentido, percatándose de la importancia de la gestión de los datos, y lo que esta representa para el futuro.

Empresas como IBM, Oracle, Informatica Corporation y otras, ofrecen diferentes soluciones flexibles, diseñadas con el objetivo de enfrentar los desafíos que hoy en día generan los volúmenes de datos distribuidos, desarrollando múltiples herramientas para la gestión de la información, generando un nuevo

enfoque en lo que a integración de datos se refiere, permitiendo a las organizaciones controlar sus datos y establecer una visión única del negocio para tomar las decisiones más precisas, concibiendo un aumento de la productividad y mejores beneficios para sus negocios.

Tecnologías que incluyen plataformas de integración, gestión, calidad y virtualización de datos son desarrolladas por estas empresas que interactúan en el mercado de software. Un ejemplo de esto lo constituye la empresa Informatica Corporation que tiene una vasta experiencia socorriendo a otras importantes empresas a rentabilizar sus activos de información. Reconocida como la compañía de integración de datos, ofrece varios software y servicios con este fin.

1.5.1. Tecnologías de Informatica Corporation

Informatica Corporation ofrece software y servicios de integración de datos que permiten a las organizaciones acceder, integrar y confiar en todos sus activos de información, brindándoles ventaja competitiva en la economía global de la información de hoy en día. Hace poco anunció dos informes sobre la creciente necesidad y la rápida actualización de las plataformas de integración de datos. Ambos plasman la importancia que manifiestan los datos para los principales procesos de negocio y las oportunidades que presentan las empresas al contar con datos más flexibles, precisos y de mayor calidad.

Ofrece varias soluciones que ayudan a las empresas a desafiar numerosos retos tecnológicos, empresariales e industriales. Soluciones que les ayudan a acceder e integrar sus activos de información con el objetivo de minimizar los costes, reducir el riesgo y aumentar su reconocimiento en este ámbito. Entre estas soluciones se pueden encontrar la plataforma de Informatica, es la primera solución completa, unificada y abierta, específicamente diseñada para la integración de datos.

Según Informatica Corporation “nuestro software abierto y neutral accede a todo tipo de datos, haciéndolos accesibles, pertinentes y útiles para las personas y los procesos que los necesitan. Con productos que promueven la colaboración en la totalidad de la organización, Informatica reduce el coste, acelera la obtención de resultados y ofrece la escalabilidad necesaria para gestionar proyectos de cualquier tamaño y complejidad”. (9)

Para lograr esta integra plataforma, esta empresa cuenta con gran cantidad de productos, entre ellos incluye:

Para la integración de datos cuenta con productos como Informatica PowerCenter e Informatica PowerExchange. Informatica PowerCenter es una plataforma unificada de integración de datos para acceder, descubrir e integrar datos de cualquier sistema, en cualquier formato, y entregarlos en cualquier punto de la empresa con la frecuencia deseada. Con alta disponibilidad, elevado rendimiento y total escalabilidad. Constituye la base idónea para todas las iniciativas de integración de datos e integración empresarial. (9)

Permite definir y aplicar transformaciones a los datos y después distribuirlos a los sistemas. El producto consta de tres ediciones, cada una orientada a cubrir un tipo de requerimiento. Son la Standard Edition con las opciones básicas, la Advanced Edition que incorpora opciones más avanzadas, y la Real Time Edition, que está orientada a la integración de datos en tiempo real. [27]

Informatica PowerExchange es una herramienta para el acceso a los datos que permite acceder a todo tipo de fuentes de datos sin tener que desarrollar programas de acceso a medida. La posibilidad de acceder a datos operacionales de misión crítica y de entregarlos en cualquier punto de la empresa en el momento que sea necesario, permite a los departamentos de IT sacar el máximo partido a sus recursos y al valor de negocio de los datos. [27] Además permite conectarse con cualquier tipo de bases de datos, Servicios Web, XML, entre otros. Puede integrarse con plataformas como Informatica PowerCenter e Informatica Data Quality.

Para la calidad de datos la plataforma Informatica cuenta con productos como Informatica Data Quality, Informatica Data Explorer e Informatica Identity Resolution. Informatica Data Quality gestiona todo el proceso de calidad de datos, permitiendo la participación de analistas, desarrolladores y administradores de manera organizada.

De su funcionalidad destaca la posibilidad de definición de reglas y servicios de calidad que pueden ser reutilizados en diferentes proyectos de calidad de datos. También dispone de reglas predefinidas y limpieza de direcciones con codificación geográfica para más de 60 países. [27]

Informatica Data Explorer es una herramienta de perfilado de datos que permite realizar el análisis de datos en tres dimensiones mediante el perfilado de datos a nivel de columna, de tabla y entre tablas. No es difícil de usar, y permite investigar, documentar y resolver problemas de calidad de datos.

A partir del análisis y el perfilado se generan metadatos, se relacionan fuentes y destinos, y se crean informes que permiten controlar todo el proceso de calidad de los datos, las anomalías, las carencias y las

mejoras a lo largo del tiempo. Se integra con Informatica PowerCenter e Informatica DataQuality para visualizar resultados, o desplegar especificaciones de limpieza y transformación de datos. [27]

Informatica Identity Resolution es un software de resolución de identidades que permite a las empresas buscar y comparar datos de diferentes sistemas. Utiliza algoritmos de comparación de manera eficiente comparando incluso datos de idiomas diferentes.

El intercambio de datos B2B es el grupo donde se engloban las herramientas que se utilizan para facilitar la integración con datos de otros negocios, con el mundo exterior, donde la manera de acceder a la información, las normas y los protocolos cambian, donde es vital garantizar la calidad de los datos entrantes, y que no se vea comprometida la seguridad de los sistemas internos. [27]

Se encuentra formado por herramientas como Informatica B2B Data Exchange e Informatica B2B Data Transformation las cuáles se integran con otras que conforman la plataforma de Informatica.

Informatica B2B Data Exchange permite definir interna y externamente perfiles de socios con las empresas con las que se intercambia información, de manera que en la misma plataforma se puedan definir reglas para las transacciones y se agilice el protocolo anterior a la puesta en marcha del intercambio. Gestiona eventos, actuales e histórico y permite llevar un control de las transacciones. [27]

Mientras que Informatica B2B Data Transformation está orientada al intercambio efectivo de información entre empresas, ofrece gran flexibilidad en cuanto a formatos, admitiendo datos tanto estructurados como no estructurados. Realiza funciones de transformación y calidad de datos de fácil utilización para que el paso de datos externos a internos sea sencillo y sin necesidad de programación. [27]

1.5.2. Tecnologías de IBM

Es una de las principales empresas a nivel mundial relacionadas con las tecnologías de la información. Tiene como principal objetivo proveer a sus clientes numerosos servicios y tecnologías que los ayuden a tener éxitos en sus negocios. Sus actividades incluyen la investigación, desarrollo, fabricación, comercialización de tecnologías y productos de hardware y software, distribución de servicios tecnológicos e integración de sistemas y consultoría de negocio. Cuenta con una plataforma de integración de datos la cual se agrupa bajo la familia InfoSphere Software dividida en 4 áreas que a su vez conforman el portafolio de soluciones de Information Management de IBM. A continuación se describen brevemente algunas de las aplicaciones que contiene dicha plataforma [28]:

Information Services Director: es uno de los componentes clave de la plataforma de integración de IBM. Proporciona un mecanismo común para publicar y administrar servicios compartidos entre las funciones de Data Quality, Data Integration y Federation. Es también el encargado de enlazar estas aplicaciones con sus metadatos y contiene los servicios de registro, información, seguridad y administración.

Change Data Capture: permite la captura de datos en tiempo real para trabajos de extracción de datos.

Federation Server: posibilita crear una vista de datos federados para un acceso rápido a orígenes de datos, se utiliza por ejemplo para evitar tener que mover los datos físicamente a una nueva ubicación.

Discovery: es una herramienta que permite crear un modelo de datos y relaciones de forma automatizada, es capaz de hacer este descubrimiento en entornos heterogéneos. Es muy útil para poder hacer extracciones de conjuntos de datos.

FastTrack: es una herramienta de creación de reglas de negocio o de conversión de datos. Dichas reglas permiten especificar la lógica de negocio necesaria para traducir los datos fuente en un formato de consumo para una aplicación de destino.

Information Analyzer: permite el análisis de los datos desde el punto de vista de su calidad, formato, precisión, longitud, compatibilidad, validez, etc. Dicha herramienta es muy útil para entender exactamente qué es lo que se tiene dentro de los orígenes de datos, permitiendo analizar previamente a cualquier proceso de integración.

Data Architect: es una herramienta de diseño y modelado de datos. IBM facilita con esta herramienta modelos predefinidos para diferentes tipos de datos e industrias, permitiendo la personalización y realización de cambios sobre los modelos. Además, permite crear un modelo lógico y físico de datos, definir los atributos de los datos, incluidas las limitaciones de dominio y los atributos de privacidad, descubrir, explorar y visualizar la estructura de las fuentes de datos así como identificar, comparar y sincronizar la estructura de dos fuentes de datos.

1.6. Otras Tecnologías

1.6.1. Oracle Data Integration Suite

Proporciona una solución completa, abierta e integrada para los entornos operativos y analíticos. Combina todos los elementos de la integración de datos, movimiento de datos, transformación, sincronización,

calidad de datos, gestión de datos y servicios de datos en tiempo real, para garantizar una información puntual, exacta y uniforme en sistemas heterogéneos. [29]

Oracle Data Integration Suite contiene un conjunto completo de componentes de gestión de datos para la construcción, implementación y administración de datos empresariales para soluciones de integración. Junto con la arquitectura orientada a servicios (SOA), proporciona la agilidad necesaria para la administración orientada a servicios de datos. [30]

Cuenta con la herramienta Oracle Data Integration 11g la cual ofrece una integración de datos en tiempo real de clase empresarial, al permitir una mayor integración con las aplicaciones y tecnologías Oracle, compatibilidad con sistemas heterogéneos adicionales y el mejor rendimiento de su categoría. [29]

Otra suite con la que cuenta es la Oracle SOA Suite, la cual constituye un conjunto de software completo para la creación, implementación y administración de soluciones orientadas a servicios incluyendo el desarrollo de aplicaciones orientadas a servicios y la integración de sistemas informáticos. Algunos de los componentes que utiliza son [31]:

Oracle Service Bus: cuenta con múltiples protocolos para conectarse a las aplicaciones, con el propósito de administrar las interacciones entre servicios heterogéneos.

Oracle Web Services Manager: es una solución completa para asegurar y administrar las arquitecturas orientadas a servicios. Permite que las políticas de administración de identidad y seguridad se definan centralmente y se impongan globalmente. Asimismo permite la definición centralizada de políticas que rigen las operaciones de servicios web, tales como el acceso, la conexión y validación de contenido, y el marco de dichas políticas en torno a los servicios sin modificación de los servicios web existentes.

1.6.2. Suite WSO2

Es una suite compuesta por una plataforma de productos que proporciona la empresa Open Source WSO2, la cual fue fundada en el 2005, y es reconocida como la empresa de código abierto enfocada en el desarrollo de tecnologías para una Arquitectura Orientada a Servicios (SOA). Se encuentra desarrollada bajo el paradigma de código abierto, utilizando la licencia Apache v2.0. Actualmente es utilizada en el centro CDAE como parte del desarrollo de soluciones de integración e interoperabilidad. De momento, entre algunas de las funciones para la que es usada se encuentran:

- Desarrollo de servicios de acceso a datos.
- Ruteo, transformación, enriquecimiento, usando el ESB o Enterprise Service Bus.

Los productos que componen la Suite WSO2 se encuentran contruidos sobre la plataforma de WSO2 Carbon. Algunos de estos productos son:

WSO2 ESB: basado en WSO2 Carbon, soporta varios protocolos de transporte, admitiendo la comunicación entre diferentes aplicaciones. Incluye transformación y ruteo de mensajes, además de permitir la gestión de la seguridad.

Permite a los administradores de sistemas y arquitectos de SOA de forma sencilla el enrutamiento de mensajes, la virtualización, la intermediación, la transformación, la programación de tareas, balanceo de carga, y otras funciones. El tiempo de ejecución ha sido diseñado para ser totalmente asíncrona, sin bloqueo, basado en el motor de mediación Apache Synapse. [32]

WSO2 Data Services Server: es un servidor de servicios de datos que provee un mecanismo muy simple y elegante para tomar los datos y exponerlos, proporcionando una plataforma fácil de usar para crear y hospedar servicios de datos. Facilita la integración de datos en los procesos de negocio, y cualquier servicio en general. [33]

Utiliza muchas características disponibles en la plataforma de WSO2 Carbon. Un servicio de datos se puede resumir en un archivo de descripción XML escrito en Data Services Descriptor Language (DSDL), un lenguaje basado en XML definido por WSO2 para escribir servicios de datos. [33]

1.7. Conclusiones

En el presente capítulo se abordaron los principales conceptos asociados a la Arquitectura Orientada a Servicios y a la integración de datos, aportando elementos importantes que sirven como fundamento para la posterior elaboración de la guía. Se realizó un análisis de las principales tecnologías y herramientas relacionadas con la integración de aplicaciones e integración de datos, concluyéndose que existen herramientas que satisfacen las necesidades de la presente investigación las cuales se tendrán en cuenta en la propuesta de solución. Sin embargo existen otras muy completas, pero en su gran mayoría son herramientas privativas, que tienen un alto valor monetario.

CAPÍTULO 2 “PROPUESTA DE SOLUCIÓN”

2.1. Introducción

En el presente capítulo se presentará la propuesta de solución para resolver el problema científico de la investigación. La solución que se propone estará basada en una guía de referencia para facilitar la integración de datos en una Arquitectura Orientada a Servicios. Por lo que está dirigida a cualquier persona que se encuentre interesada en llevar a cabo esta tarea. Contiene la información necesaria para efectuar satisfactoriamente el proceso de integración de datos en una SOA.

2.2. Propósito de la guía

Las dificultades que afrontan hoy las empresas para lograr la integración de sus aplicaciones a nivel de datos, constituyen la fuente de esta guía. Su objetivo principal es brindar pasos, técnicas y recomendaciones que facilite el intercambio de datos entre diferentes sistemas en una Arquitectura Orientada a Servicios. Además aconsejar y proponer buenas prácticas para facilitar la integración de dichas aplicaciones, permitiendo eliminar las deficiencias existentes para lograr la integración con la calidad requerida. Por otra parte se planea que sea aplicable en diferentes proyectos que estén relacionados con la integración semántica en un entorno SOA, a través del intercambio de datos de sistemas o aplicaciones de una o varias empresas.

2.3. Descripción de la guía

Actualmente las empresas, desde diferentes puntos de vista, necesitan que sus aplicaciones sean integradas. Esto responde a sus necesidades de negocio y más aún cuando se encuentran en un entorno SOA. A continuación se describirán los pasos a seguir para facilitar la integración entre diferentes aplicaciones a nivel de datos. En la siguiente figura se reflejan los pasos a seguir.

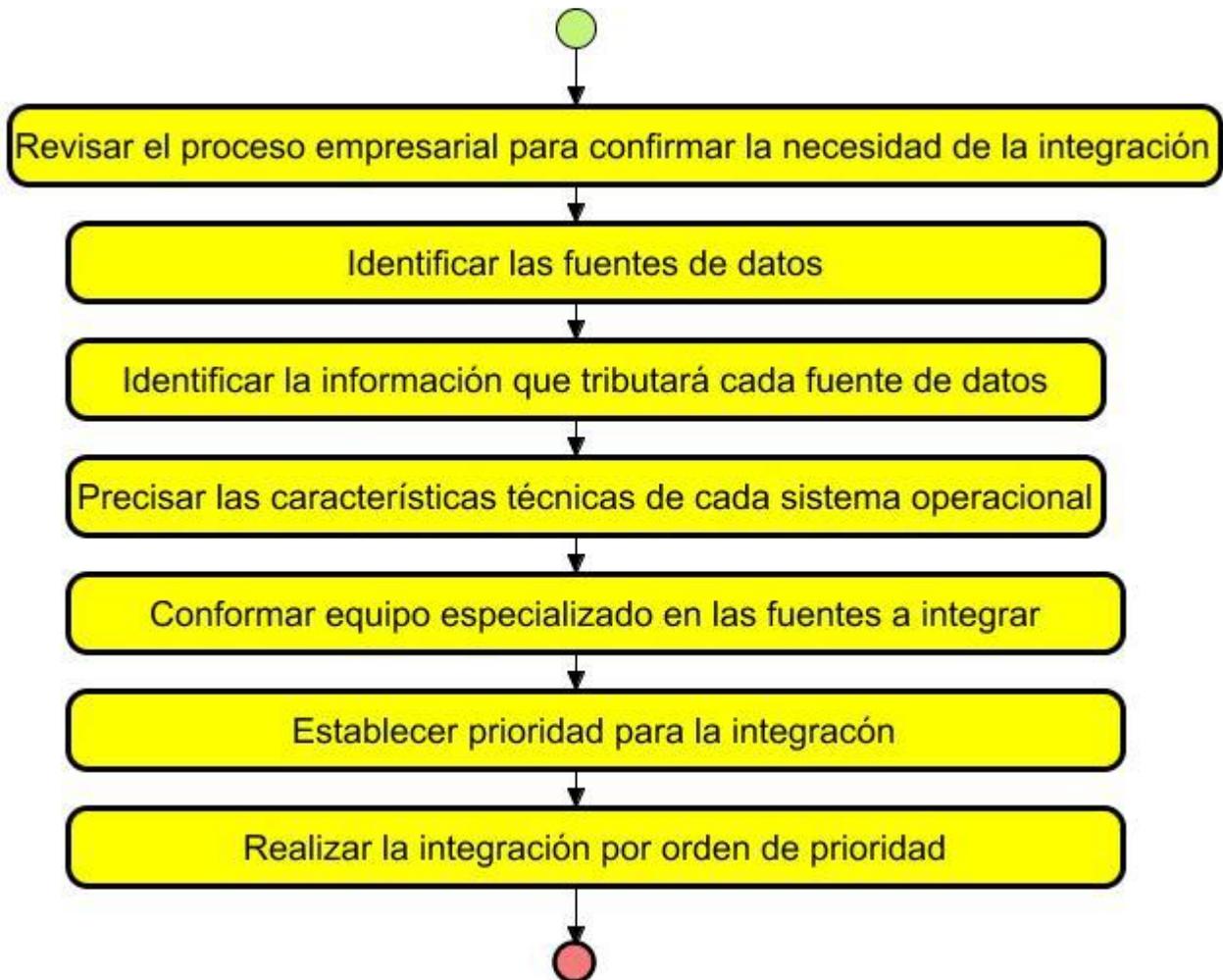


Figura 1. Pasos a seguir.

2.3.1. Revisar el proceso empresarial para confirmar la necesidad de la integración

Se encuentra determinado por las necesidades reales que presentan las empresas, o sea responde a los objetivos estratégicos de cada una de ellas. Dichos objetivos son los que definen las empresas en un determinado tiempo para lograr avances. En consecuencia cada empresa posee sus propias características y en dependencia de la necesidad que tengan eligen que aplicaciones van a integrarse y con qué objetivo. A veces es necesario integrar diferentes aplicaciones en una o varias empresas. Por tal motivo es importante tener un conocimiento profundo del negocio y lo que este implica para tener una concepción clara de lo que se quiere realizar. Con tal propósito es fundamental realizar una entrevista

entre las personas que se encuentran interesadas en la integración, que sea lo más directa y fluida posible. Los participantes de dicha entrevista serán los analistas principales y los arquitectos de software por cada parte de las entidades involucradas. En caso de que sea en una misma empresa, entonces sería entre personas que se encuentran relacionadas con las aplicaciones. Dicha entrevista será el escenario perfecto para que se identifiquen los principales requisitos del negocio, con el objetivo de conocer si resulta provechoso integrar estas aplicaciones.

En caso de que las partes involucradas hayan llegado a la conclusión de efectuar la integración se procede a identificar las fuentes de datos de las aplicaciones implicadas en el proceso de integración.

2.3.2. Identificar las fuentes de datos

Luego que se ha confirmado la integración, el siguiente paso consiste en identificar las fuentes de datos, o sea conocer donde se encuentra la información que se desea transportar, utilizar, transferir y manipular. Actualmente en las empresas, la información que se maneja no siempre se encuentra en el mismo formato o ubicación. Debido a que dicha información se encuentra diseminada en múltiples fuentes de datos, pues los sistemas fueron desarrollados independientemente o por diferentes empresas. A raíz de dicha situación, existen una gran variedad de fuentes, por lo que una correcta identificación de las fuentes y la certeza de que los datos a mover son los correctos se hace indispensable ya que en algunos casos será necesario trabajar con grandes cantidades de datos y no solo con datos específicos.

Generalmente, las empresas constan de varias áreas de trabajo, las cuales se encuentran en relación con las funciones que realiza cada empresa con el fin de lograr sus objetivos y adaptarse a sus necesidades específicas. Dada la gran variedad de áreas existentes, se suele trabajar con diferentes tipos de información y para obtener dicha información, recurren a las fuentes de información. Dentro de estas fuentes podemos encontrar bases de datos, registros de personal, de inventarios, de ventas y de costo. Frecuentemente se pueden encontrar con varias bases de datos en una empresa relacionadas lógicamente entre sí, mediante un software, que realiza funciones de crear, actualizar, recuperar y generar informes. Mientras los registros en su mayoría son gestionados a través de documentos Excel.

Las bases de datos constituyen un sistema de almacenamiento y acumulación de datos que permiten la gestión de la información. Permite obtener y almacenar información de diferentes índoles, como por

ejemplo, datos de los departamentos, de los empleados, de los salarios, de actividades a desarrollar, entre otros. Otras fuentes de datos utilizadas son los repositorios de datos, ODS, Datamart, y Data Warehouse.

La técnica a utilizar para identificar las fuentes de datos consiste en realizar una entrevista o reunión entre los arquitectos de software y técnicos especializados en tecnología de las partes involucradas. Dichos técnicos son los que tienen conocimiento de donde se encuentra almacenada toda la información que manejan sus empresas respectivamente. La reunión tiene como objetivo, que cada técnico exponga las fuentes de datos que manejan información y pueden quedar comprendidas en este proceso. A partir de cada uno de sus planteamientos y en conjunto con los arquitectos han de llegar a la conclusión de cuales fuentes estarían finalmente incluidas en el proceso de integración. Al finalizar deberá llenarse una planilla donde queden expuestas dichas fuentes de datos por cada una de las empresas.

En caso de que la integración se realice dentro de una misma empresa, el proceso es similar. En este caso, la entrevista sería entre los analistas principales y los técnicos especializados en tecnología asociados a los sistemas inmiscuidos en la integración. Consecuentemente, en la planilla quedaría la relación de las diversas fuentes de datos a utilizar para la integración, por cada uno de los sistemas inmiscuidos en el proceso, la cual se muestra en el **Anexo 1**.

2.3.3. Identificar la información que tributará a cada fuente de datos.

A partir de la identificación de las fuentes de datos, se debe determinar la información que va a ser utilizada, lo que permite tener un control de los datos que serán manejados por las aplicaciones. Al identificar la información se pueden encontrar con algunos problemas, como por ejemplo:

- Existe información duplicada en las fuentes de datos.
- Incongruencias debido a estructuras de información, incomprensibles de una aplicación a otra.
- Falta de integridad de la información.

Las aplicaciones en considerables ocasiones manejan datos relacionados con los trabajadores y clientes de las empresas. Habitualmente, es frecuente encontrarse con información repetida en diferentes sistemas de almacenamiento, donde en la gran mayoría presentan diferentes formatos. A partir de tal situación se generan los problemas expuestos anteriormente, viéndose afectado también el negocio de la empresa debido a que no posee una visión unificada de un cliente o trabajador determinado.

2.3.4. Precisar las características técnicas de cada sistema operacional

Es importante determinar las características técnicas de cada sistema que se vaya a integrar, con el objetivo de verificar compatibilidades. Para cumplir a cabalidad esta tarea será necesario realizar una entrevista o reunión. Dicha reunión tendrá como principales representantes a los arquitectos de software y técnicos especializados en las diferentes tecnologías, por cada una de las empresas asociadas a los sistemas inmiscuidos en la integración. El intercambio entre estas personas resultará muy provechoso, puesto que les permitirá determinar las características principales de los sistemas y nodos asociados al proceso. Por cada sistema deberá llenarse una planilla especificando sus características, responsabilizando en este caso a los arquitectos de software. La planilla que deberán llenar precisando las características técnicas de cada sistema operacional se muestra en el **Anexo 2**.

Luego de determinar las características de los sistemas, el ambiente estará creado para ejecutar el siguiente paso.

2.3.5. Conformar equipo especializado en las fuentes a integrar

Consiste en seleccionar un conjunto de personas que serán los encargados de guiar el proceso de integración. Se seleccionará un equipo por cada sistema involucrado, para facilitar el trabajo. Los grupos estarán conformados por un líder de equipo, los especialistas en tecnología y un arquitecto. Dichos equipos serán los encargados de establecer un orden de prioridad para la integración, así como supervisar continuamente cada detalle durante el proceso.

Para conformar un equipo, primeramente se debe designar al líder el cuál será la persona encargada de dirigir todo el proceso, integrar los esfuerzos y alentar al equipo para conseguir la meta. Un arquitecto de software se integrará también al equipo, pues es la persona que se encargará de construir, analizar, pensar, como se va a desarrollar la integración a partir de los conocimientos e ideas que puedan aportar los restantes miembros del grupo. Posteriormente se designará un especialista por tecnología, o sea, un diseñador de base de datos por cada tipo de base de datos que exista, así como un desarrollador por cada lenguaje usado en la implementación de los sistemas.

2.3.6. Establecer prioridad para la integración

Conformados los equipos especializados, se reunirán con el objetivo de establecer un orden de prioridad para realizar la integración. Analizarán la situación y decidirán cuál de los sistemas se integrarán primero, estableciendo a partir de este momento un orden de prioridad para continuar con el proceso. Aquí quedarán definidos los sistemas que serán integrados y los pendientes a integrar en un futuro. El orden puede ser establecido teniendo en cuenta los siguientes aspectos:

- Necesidad de la integración.
- Objetivos estratégicos y tácticos de las empresas.
- Capacidad de información.
- Rendimiento de los sistemas.

Los aspectos a tener en cuenta para establecer el orden de prioridad serán determinados a partir de las características propias de las empresas y entornos en que se desarrolle la integración. Dicho orden deberá quedar plasmado en un documento oficial, el cual será redactado por uno de los integrantes del grupo mientras se desarrolla la reunión. Posteriormente se procede a la integración de los sistemas, puesto que las condiciones han sido creadas para su realización.

2.3.7. Realizar la integración por orden de prioridad

Los pasos anteriormente descritos constituyen la base de este paso, o sea, permiten crear las condiciones para realizar la integración de las diferentes aplicaciones que se encuentran incluidas en el proceso de adopción de una SOA.

Para realizar la integración se propone la aplicación de un conjunto de pasos, los cuales permitirán de manera ordenada y organizada llevar a cabo dicha integración. A continuación en la siguiente imagen se presentan los mismos.

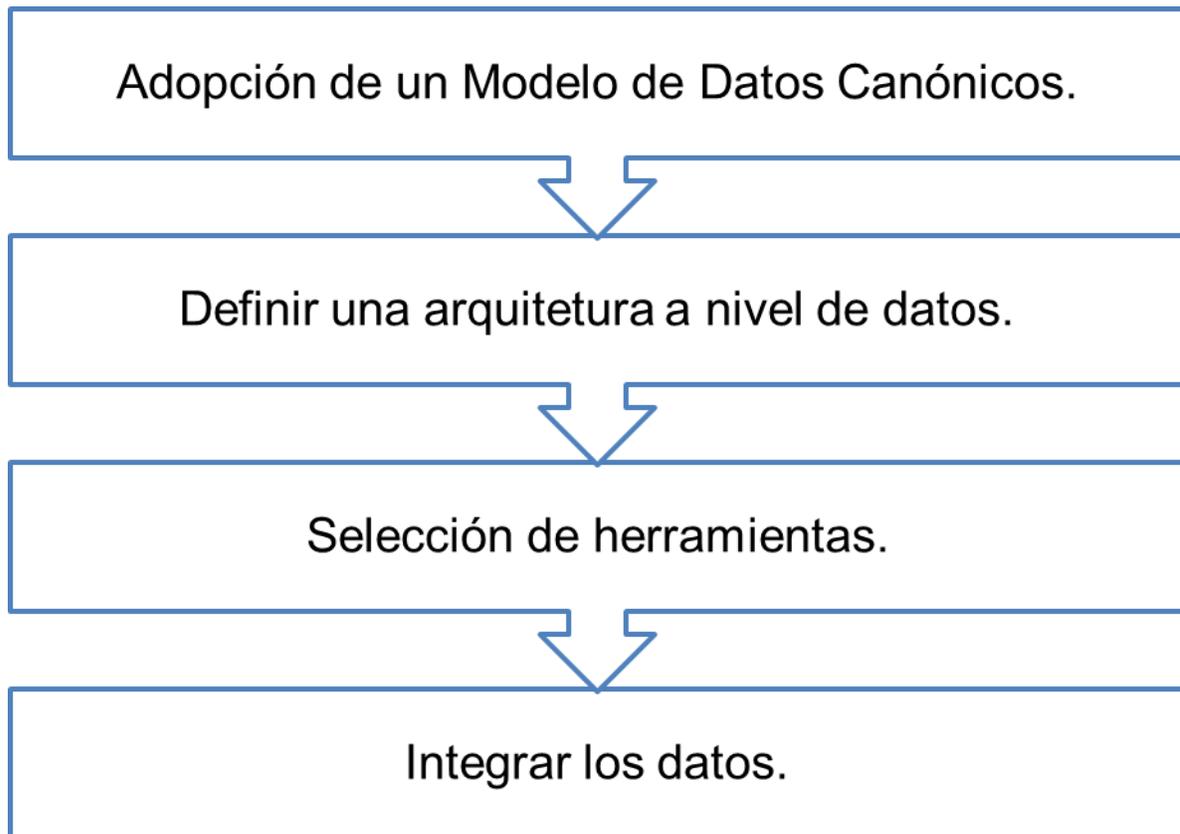


Figura 2. Pasos para realizar la integración.

Uno de los factores a tener en cuenta constituye la estandarización de las estructuras de datos. Poseer un modelo de datos común proporciona buenos niveles de reutilización y flexibilidad organizacional. Sin embargo, es muy probable que la existencia de un modelo de datos común y transversal sea más la excepción que la regla, y que la mayor parte de las organizaciones de gran tamaño tengan bases de datos con esquemas redundantes. También puede que la documentación de los modelos de datos no esté disponible, o sea propietaria. Por último, es muy probable que el grado de acoplamiento, o dependencia estructural, entre las aplicaciones de software y sus respectivos modelos de datos sea tan alto que haga imposible su modificación a un costo razonable. Debido a todo lo anterior, una iniciativa de adopción de SOA debe tomar la estandarización de datos desde otro ángulo, buscando la estandarización de las estructuras a nivel de mensajería [34]. Por tales razones, se propone utilizar la mensajería con el objetivo de estandarizar las estructuras a nivel de datos.

La mensajería constituye un estilo de integración de aplicaciones, que en relación con los otros estilos, es más rápida que la transferencia de archivos, mejor que el encapsulado de base de datos compartida, y más fiable que la invocación a procedimiento remoto [10]. Se encuentra soportada por varios patrones que permiten agilizar el proceso de integración. Los mismos proporcionan soluciones a los diferentes problemas que pudieran ocurrir, solo habría que adaptarlos al entorno en el que se quiere aplicar.

Existen numerosos patrones relacionados con la mensajería, entre los que se encuentra el Modelo de Datos Canónicos que pertenece al grupo de patrones relacionados con la transformación de mensajes y junto al ESB será un factor imprescindible en la arquitectura que se propone.

2.3.7.1. Adopción de un Modelo de Datos Canónicos

El primer paso que se recomienda para realizar la integración consiste en adoptar un Modelo de Datos Canónicos (MDC). Su adopción permite resolver las dependencias de los formatos de datos y la semántica utilizada por las aplicaciones. Sin embargo no siempre es viable utilizar un MDC, en caso de ser 2 aplicaciones no se recomienda, pues puede resultar complicado. Pero en un entorno SOA, donde varias aplicaciones necesitan intercambiar información, resulta muy efectivo su uso.

La adopción de un MDC dependerá en gran medida de las características del entorno donde se desarrolle el proceso de integración. En numerosas situaciones, se suele contar con las herramientas necesarias para acceder a la información que contienen los sistemas y aplicaciones. En caso de encontrarse en tal situación, se podría utilizar un enfoque Bottom-Up, de abajo hacia arriba. El mismo se desarrolla a partir de efectuarle la ingeniería inversa a las estructuras de datos existentes.

Si las empresas o entidades se encuentren interesadas en concebir un modelo más sólido y consistente. Se recomienda utilizar un enfoque Top-Down, de arriba hacia abajo. El mismo se enfoca en los requerimientos del negocio, por lo que puede tardar bastante tiempo. Incluso puede demorar hasta 6 meses la confección, pero el resultado sería una solución bastante persistente y estable.

Si se quiere integrar en un ambiente, donde existe una buena proyección de los planes futuros, la información con la que se cuenta no es vulnerable a cambios y se presagie que la misma tributará posteriormente a las aplicaciones existentes y futuras. Se podría crear o diseñar un conjunto de datos bastante grande, teniendo como beneficio que por un prolongado período de tiempo, el modelo no necesitará modificarse. Sin embargo se necesitará un buen tiempo para determinar toda la información

que contendría el mismo.

De lo contrario aunque se tenga una buena proyección de los planes futuros, es posible que la información sea vulnerable y propensa al cambio. Para este caso no es recomendable utilizar el enfoque anterior, pues el diseño se tornaría inútil habiéndose malgastado tiempo innecesario en una solución que se volvería ineficaz e inservible. Para no incurrir en dicha situación se recomienda diseñar el modelo a partir de un pequeño conjunto de datos. Dicho conjunto, inicialmente contendrá la información relevante para con las aplicaciones involucradas. La misma sería imperceptible, comparada con la generada por el otro enfoque. El beneficio de este punto de vista radica en que se requerirá menos tiempo en diseñar la solución, debido a que el conjunto de información a utilizar se podrá definir con mayor rapidez. Pese a ser una solución más rápida, el modelo requerirá de cambios y actualizaciones por lo que irá creciendo en caso de que nuevas aplicaciones sean incorporadas.

Otro enfoque sería utilizar toda la información del sistema origen. El mismo es percibido desde el punto de vista que una empresa desee comenzar un proyecto basado en una SOA. En este caso la empresa contaría con un solo sistema de interés para comenzar con el proyecto. Aquí se podría utilizar toda la información del sistema origen. Las aplicaciones creadas a partir de este momento deberán ser concebidas desde un inicio con el objetivo de utilizar el modelo generado inicialmente y tendrán que adaptarse al mismo. La contrapartida de este enfoque reside en que dicho modelo podría contener información innecesaria para las nuevas aplicaciones.

Sobre los diferentes enfoques que existen, la definición de un MDC estará marcada por las características, proyecciones y propósitos que tengan las entidades y empresas interesadas en la integración de sus aplicaciones. Sin embargo existen varios consejos que podrían servir de gran ayuda, a pesar de los diversos entornos que se pueden encontrar, como por ejemplo [11]:

1. Comenzar con algo pequeño, o sea, generar la mínima cantidad de datos.
2. Agregar los campos comunes entre la mayoría de las aplicaciones que participan en la integración.
3. El MDC debe contener del 80-90% de la información que es requerida por todas las aplicaciones. Permitiendo que se reduzca al mínimo, el tiempo dedicado a la creación de la forma canónica inicial y se puedan minimizar el número de veces que podría ser cambiada o modificada.

4. Realizar una investigación, pues existen varias organizaciones que han creado representaciones de objetos o estándares que se ajustan a sus necesidades y podrían ser utilizados como punto de partida.

Luego que se haya definido el enfoque para definir un MDC, se deben reunir los diseñadores de bases de datos y el arquitecto de cada uno de los sistemas involucrados, para diseñar un formato único que les permita representar los datos. Luego cada grupo expone su solución a los otros equipos que interactúan y exigen la adopción con los arquitectos de un único MDC. Dicho modelo, dependiendo del entorno en el que se desarrolla la integración podría quedar representado mediante un conjunto de archivos XML, como parte de un Modelo de Mensaje Canónico que podría utilizarse dentro del ESB o podrá ser utilizado para que los diseñadores de bases de datos en conjunto con los arquitectos construyan una nueva base de datos centralizada, permitiendo a los sistemas compartir la información en el nuevo formato.

En caso de que sea necesario construir una nueva base de datos que centralice los datos, dichas figuras muestran cómo quedaría el cambio.

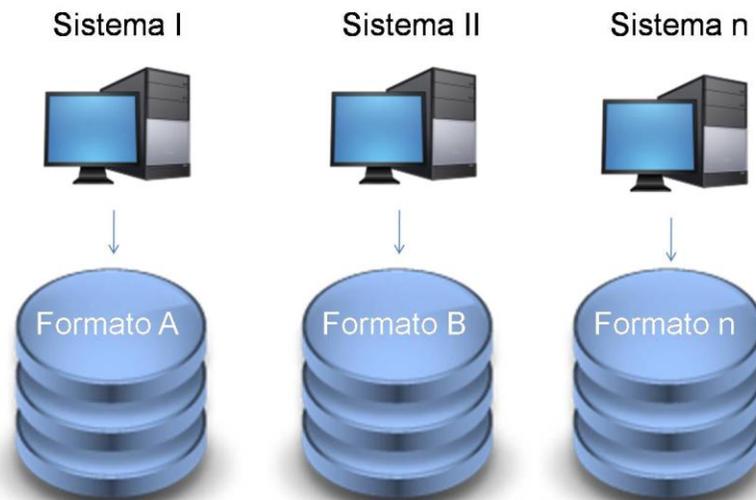


Figura 3. Antes de utilizar MDC.

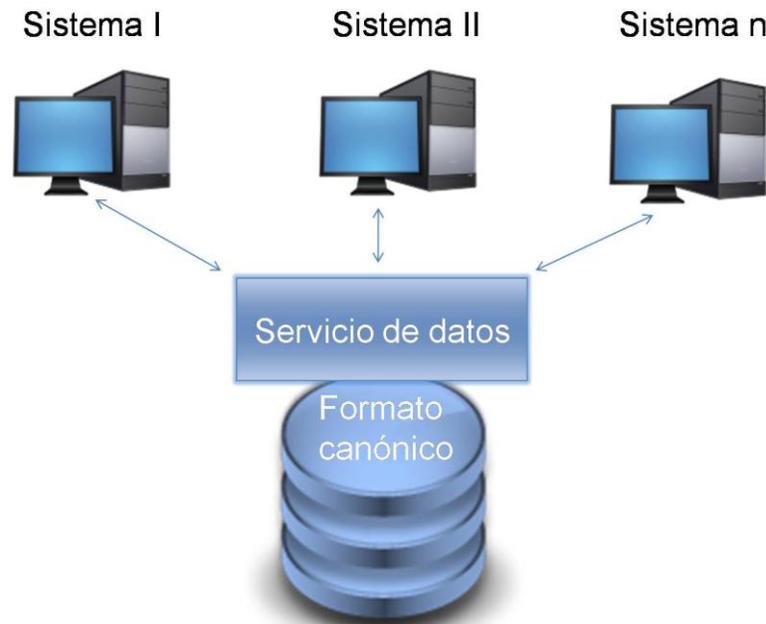


Figura 4. Utilizando un MDC.

2.3.7.1.1. Aspectos a tener en cuenta en el diseño de un MDC.

Luego de haber definido como será construido su MDC, es recomendable enfocarse en el diseño del mismo. Aquí intervienen varios modelos para definir y analizar las estructuras de datos. Dichos modelos son el Modelo Conceptual de Datos, Modelo Lógico de Datos y el Modelo de Mensaje Canónico. A continuación estaremos mostrando algunas habilidades y consejos a tener en cuenta al desarrollar los mismos.

Durante el desarrollo de un modelo conceptual es importante tener en cuenta [35]:

- Listar todas las entidades principales que requiere su empresa.
- Definir cómo las entidades se relacionan entre sí.
- Especificar los atributos de las entidades.

Luego de haber desarrollado el modelo conceptual, habrá que concentrarse en obtener el modelo lógico o físico. Un elemento importante a tener en cuenta en el desarrollo de este modelo consiste en asegurarse que el mismo es fácil de usar y adecuado para lo que se quiere lograr. Por tanto, algunos aspectos importantes a tener en cuenta son:

- Que el modelo sea divisible, o sea, si se pueden utilizar partes de él sin tener que tomar todo.
- Prever en su construcción el uso posteriormente de herramientas de mapeo.
- Los tipos de datos se encuentran definidos correctamente.

Otro aspecto significativo a considerar es la operatividad del modelo y de cómo la gente hace uso de ella. En este sentido, los arquitectos deberían revisar y aprobar cualquier aplicación del modelo, asegurando que el uso del mismo es consistente a través de dominios y puestas en práctica. [35]

Uno de los modelos que intervienen en el diseño de un MDC, es el Modelo de Mensaje Canónico (MMC). El mismo representa el formato estándar utilizado para intercambiar información en un bus de servicio, el cual constituye una pieza fundamental en una SOA. Actualmente en una estrategia de adopción de una SOA es frecuente encontrarse con que los proveedores y consumidores de servicios, utilizan diferentes modelos y meta modelos de mensajes, lo que origina que se intercambien mensajes sintácticamente incompatibles. Cuando esto ocurre es recomendable utilizar un MMC dentro del ESB. La siguiente figura muestra el MMC dentro del ESB [36]:

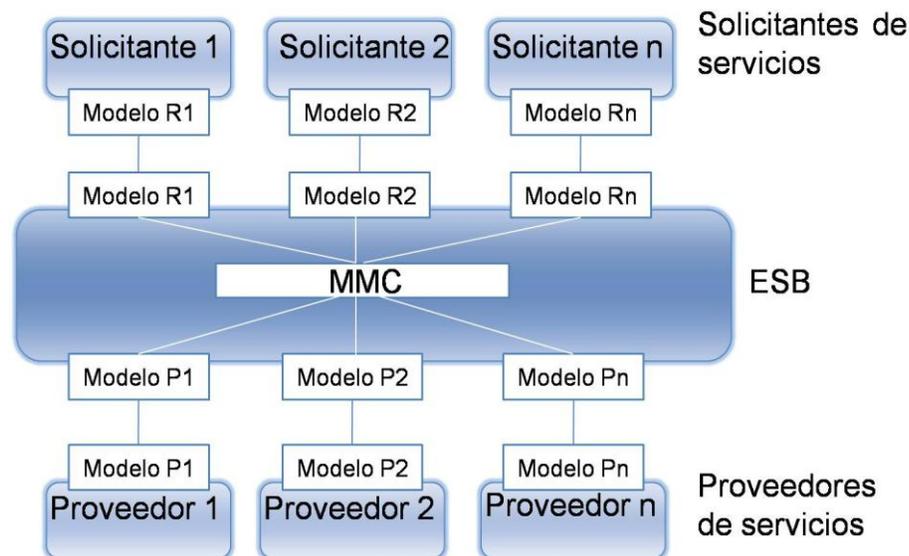


Figura 5. El MMC en un ESB.

La Figura 5 muestra como cada solicitante y proveedor necesita una transformación del MMC, suponiendo que todos ellos tienen modelos diferentes. Si un nuevo servicio, como un solicitante de servicio o proveedor, se está introduciendo y este servicio utiliza un modelo de mensaje propio, sólo la

transformación entre el CMM y el modelo de mensaje específico de la aplicación se debe crear, sin importar el número de aplicaciones que ya participan. [36]

Por ende, la utilización de un MMC estaría reduciendo la complejidad de la conectividad de un ESB, el cual además de establecer la conectividad entre los servicios, incluye la transformación entre los modelos y meta modelos de mensajes diferentes. Normalmente sin el uso del MMC, la complejidad sería de $n * m$, y al insertar dicho modelo, quedaría reducida a $n + m$.

La siguiente figura refleja las transformaciones que tendría que realizar el ESB sin el uso de un MMC. [36]

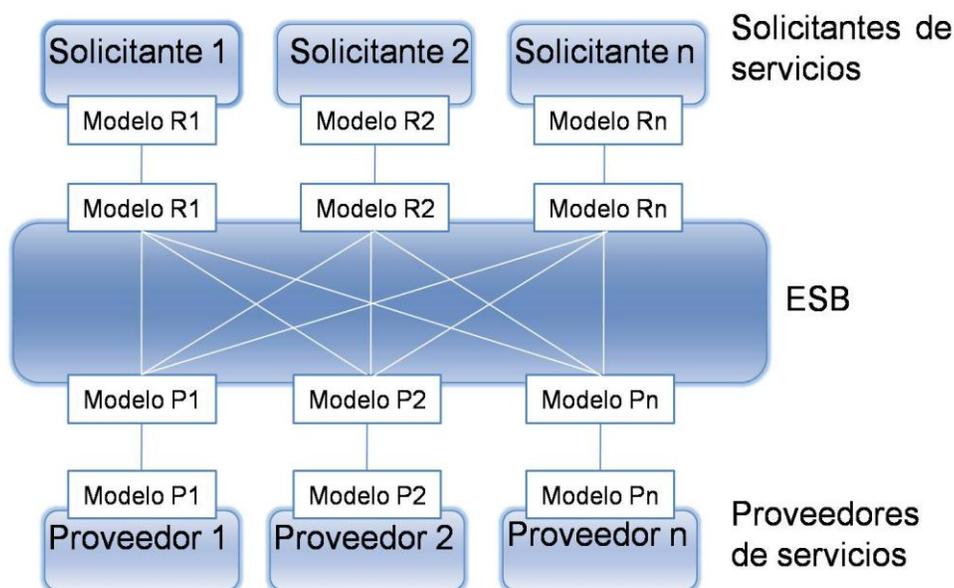


Figura 6. Transformaciones en el ESB sin un MMC.

La Figura 6 ilustra el problema de llevar a cabo la transformación directa entre los solicitantes y proveedores de servicios. Cada servicio (solicitante / proveedor) utiliza un modelo propio para el mensaje de la comunicación. Para que los solicitantes puedan invocar a los proveedores, los modelos de mensajes de los solicitantes deben ser transformados en los modelos de mensaje de los proveedores.

Consecuentemente el ESB deberá comprender todos los modelos de mensaje de propiedad, y los mapas ESB, o transformaciones, entre los modelos de mensaje. Proporcionando las transformaciones para cada interacción posible. Por cada solicitante (n), se debe especificar una transformación por separado a cada proveedor (m), y así sucede respectivamente de un proveedor para cada solicitante. El número total de

transformaciones sería de $n * m$. Si un nuevo proveedor es añadido y todos los solicitantes tienen que interactuar con él, n nuevas transformaciones hay que agregar. Así mismo sucede si un nuevo solicitante se añade y necesita interactuar con todos los proveedores, entonces m transformaciones hay que agregar. En efecto, ampliar el alcance una solución de integración en este tipo de arquitectura resultaría muy costoso. Sin embargo, la disminución en el número de transformaciones muestra la ventaja de utilizar un MMC en el ESB, al reducir el número de transformaciones de $n * m$ a $n + m$ para n solicitantes que invocan servicios de m proveedores. [36]

La Figura 5, refleja uno de los enfoques de cómo puede ser utilizado un MMC en un ESB. En este caso, el ESB necesita [36]:

- Traducir el modelo de mensaje del solicitante para el MMC.
- Traducir del MMC para el modelo de mensaje del proveedor.

Con tal propósito se efectúa una transformación entre cada solicitante y el MMC. Así como entre el MMC y cada proveedor. Suponiendo en cada uno de los casos que son modelos diferentes. Las características esenciales que presenta son [36]:

- Los servicios individuales o aplicaciones conservan sus modelos de mensajes de propiedad.
- El ESB adopta un propio modelo de mensaje externo para cada solicitante y proveedor del servicio.
- El ESB adopta un MMC interno.
- Para cada solicitante, debe haber una transformación del modelo de mensaje solicitante al MMC (una segunda transformación se encarga de la vuelta para transformar el modelo de mensaje del MMC al solicitante).
- Para cada proveedor, debe existir una transformación del MMC al modelo de mensaje del proveedor (una segunda transformación se encarga del regreso, para transformar modelo de mensaje del proveedor al MMC).

Además ofrece un conjunto de ventajas que son mencionadas a continuación [36]:

- Los servicios existentes no tienen que cambiar, por lo que es más fácil llegar a un acuerdo entre los diferentes grupos que tienen intereses contrapuestos en materia de servicios. A veces esta es la única manera para las aplicaciones heredadas y empaquetados que están conectados a la ESB.
- Este enfoque es bastante típico en una SOA.

- Limita el número de transformaciones que deben ser desarrollados y gestionados de $n * m$ a $n + m$.
- El ESB proporciona un punto de aplicación para la gestión centralizada y la gestión de todas las transformaciones.
- Es más fácil introducir la lógica de la mediación reutilizable, ya que la lógica puede utilizar el MMC en vez de modelos de mensajes específicos del servicio.

Sin embargo una de las limitaciones que presenta dicho enfoque es que se debe realizar una doble transformación entre las aplicaciones y servicios que interactúan, aumentando así la sobrecarga de procesamiento. Para los modelos de mensajes complejos o de los productos ESB con las limitaciones de rendimiento de transformación, esta sobrecarga puede ser significativa. [36]

El uso de un MMC desde el enfoque planteado anteriormente constituye un aspecto a tener en cuenta para las empresas que cuentan con una estrategia de adopción de SOA. Desde este punto de vista las aplicaciones implicadas pueden mantener sus modelos de mensajes de propiedad. Puesto que en ocasiones nos encontramos con modelos complejos, que serían muy difíciles de cambiar.

Otro enfoques relacionados con la utilización de un MMC en un ESB es conocido como servicios de traducción y consiste en que los solicitantes y proveedores de servicios, tienen que traducir sus modelos originales en el MMC. Aquí el ESB no es responsable de la transformación de los modelos de mensajes. La infraestructura de las aplicaciones solicitantes y traductores de servicios traducen su modelo de mensaje de propiedad al MMC. [36]

En la siguiente figura queda reflejado como quedaría representado.

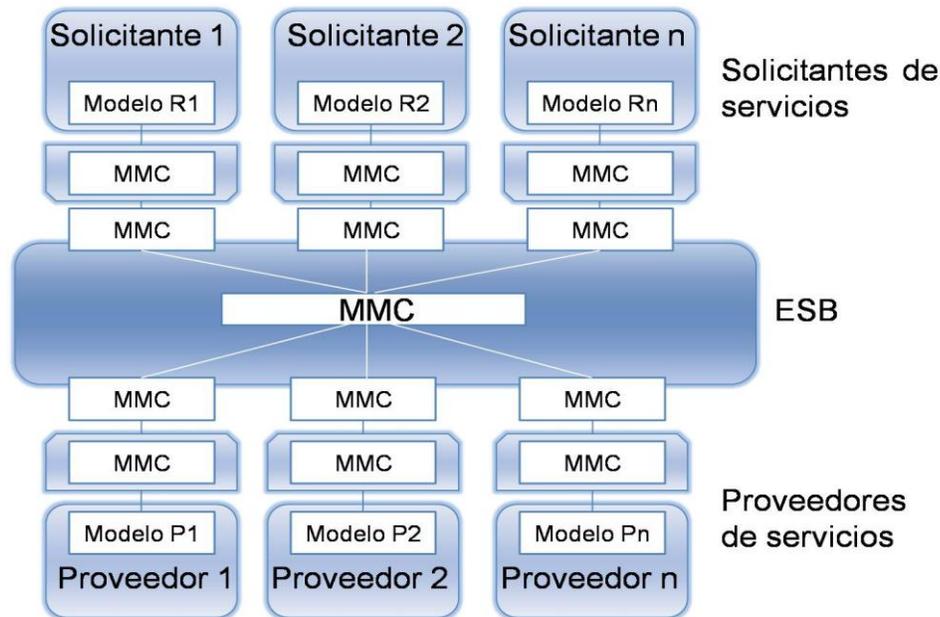


Figura 7. Solicitantes y proveedores de servicios traducen su modelo propietario al MMC.

Las características esenciales que presenta son [36]:

- El ESB adopta el MMC a nivel interno como externo y simplemente se conecta a los solicitantes y los proveedores.
- Los solicitantes y proveedores se encargan de transformar el modelo de propiedad al MMC, ocurriendo en el entorno del cada uno respectivamente.

Las ventajas que proporciona son [36]:

- El ESB delega la transformación a los servicios, lo que elimina la carga de gestión operativa en el ESB.
- Las organizaciones o ambientes donde hay personal suficiente para el desarrollo de una aplicación, pero sólo muy poco apoyo en la ESB se pueden beneficiar.
- En los casos en que el modelo de los solicitantes y los proveedores coinciden, no hay otras transformaciones.

La desventaja que presenta es que las transformaciones deben ser implementadas dentro de la infraestructura del servicio. Constituyendo en determinados casos un problema, pues existen aplicaciones

que sus estructuras de datos resultan un poco complejas. Por consiguiente resultaría muy costoso y complicado tratar de adaptar sus estructuras de datos al MMC.

El otro enfoque es conocido como el escenario ideal, donde los mensajes utilizados por los solicitantes y proveedores de servicios han sido definidos con el MMC. Aquí no se requiere de transformaciones puesto que todas las aplicaciones utilizan el MMC. La siguiente figura muestra el escenario ideal para utilizar un MMC. [36]

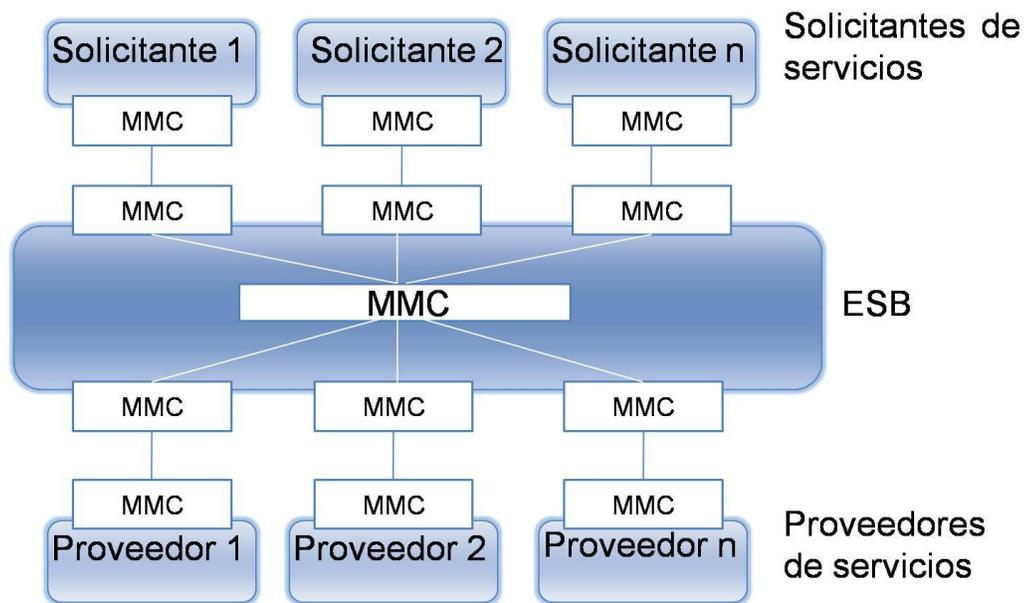


Figura 8. Solicitantes y proveedores de servicios traducen su modelo propietario al MMC.

Las características que presentan son [36]:

- Los solicitantes y proveedores de servicios utilizarían la MMC externa y no utilizarían su propio modelo de mensajes.
- El ESB no se tiene que preocupar por traducir, su papel será el de dirigir las llamadas de servicio.

Las ventajas que proporciona son [36]:

- Constituye el escenario ideal para cuando se creen nuevas aplicaciones, ya que pueden adoptar el MMC como modelo de datos internos y, por tanto, reducir el esfuerzo de desarrollo de las

transformaciones que hayan de llevarse a cabo para la asignación entre un modelo de datos interno y el modelo de mensaje externo.

- Distribuye la responsabilidad de tal manera que los solicitantes y proveedores deben garantizar la conformación MMC, reduciendo la carga operativa en el ESB, proporcionándole más libertades, pues no se tiene que preocupar por la transformación de mensajes.
- Se eliminan los costes de transformación, representando una reducción de la complejidad y aumento del rendimiento.

Sin embargo presenta varias desventajas [36]:

- Puede resultar muy costoso y complejo consumir lo que se pretende. Pues en caso de contar con sistemas legados, sería muy difícil que estos cambiaran su modelo interno para adaptarse al MMC.
- Lo mismo ocurriría con otros sistemas que estén por integrarse. Tratar de adaptar sus modelos implicaría un costo adicional afectando la inversión de las entidades.

Es importante tener en cuenta los aspectos y consejos anteriormente expuestos para adoptar un MDC. Los cuales permiten tener una concepción más clara de cómo desarrollar algunos de los modelos que intervienen en el proceso. Además es preciso recalcar que para la utilización de un MMC en un ESB, se recomienda utilizar el primer enfoque el cual es mucho más asequible para las empresas interesadas en implantar una SOA.

Sin embargo, el MDC por sí solo no garantiza la integración a nivel de datos, por lo que necesita de otros patrones de transformación que lo apoyen a concretar esta idea.

2.3.7.1.2. Patrones a tener en cuenta

Envelope Wrapper o Enrollador de sobre: Tiene como propósito encapsular el mensaje para que sea transportado. A continuación se exponen los pasos para realizar el proceso de encapsulado del mensaje [10]:

1. La fuente del mensaje publica el mensaje en un formato que normalmente no cumple con los requisitos de la infraestructura de mensajería.
2. El enrollador de mensaje, envuelve el mensaje y lo transforma a un formato que se ajuste al sistema de mensajería. Durante este proceso se pueden incluir nuevos campos a la cabecera del

mensaje, el mensaje puede ser cifrado y añadirse credenciales de seguridad.

3. El sistema de mensajería transporta el mensaje.
4. Antes de que el mensaje llegue a su destino, se realiza el proceso inverso del enrollado. Durante esta fase se pueden eliminar algunos de los campos de cabecera, descifrar el mensaje, o verificarse las credenciales de seguridad.
5. El receptor recibe el mensaje.

La siguiente imagen muestra el proceso anteriormente expuesto

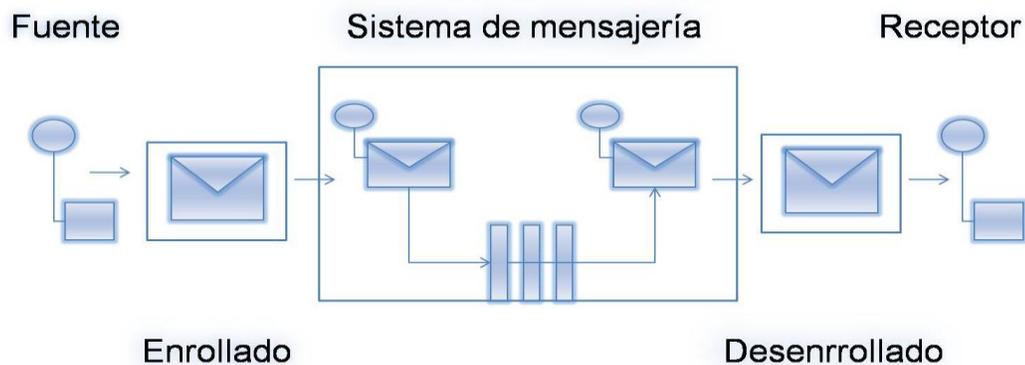


Figura 9. Proceso de encapsulado del mensaje.

Content Enricher o Enriquecedor de contenido: Utiliza la información obtenida del interior del mensaje, para recuperar los datos que faltan de una fuente externa. Los datos adicionales por el enriquecedor de contenido seguramente fueron obtenidos de diferentes formas. A continuación se verán algunas de las situaciones más frecuentes a partir de las cuales se obtiene la nueva información [10]:

- 1- **Cálculo:** El enriquecedor de contenido puede ser capaz de calcular la información que falta y mediante un algoritmo incorporar la información adicional. En consecuencia, no se necesitaría ninguna fuente de datos externa.
- 2- **Medio Ambiente:** Del mismo modo podría recuperar los datos del sistema operativo, donde el caso más habitual es el de una marca de tiempo. En caso que no se incluyera en el mensaje esta información, el enriquecedor de contenido podría obtener la hora actual del sistema operativo y

añadirlo al mensaje.

- 3- **Otro sistema:** Es la opción más común utilizada por el enriquecedor de contenido y se basa en recuperar los datos perdidos desde otro sistema. Permite obtener la información desde distintas fuentes fuente de datos, incluyendo bases de datos, archivos, directorio LDAP, otra aplicación o un usuario que ingrese manualmente los datos que faltan.

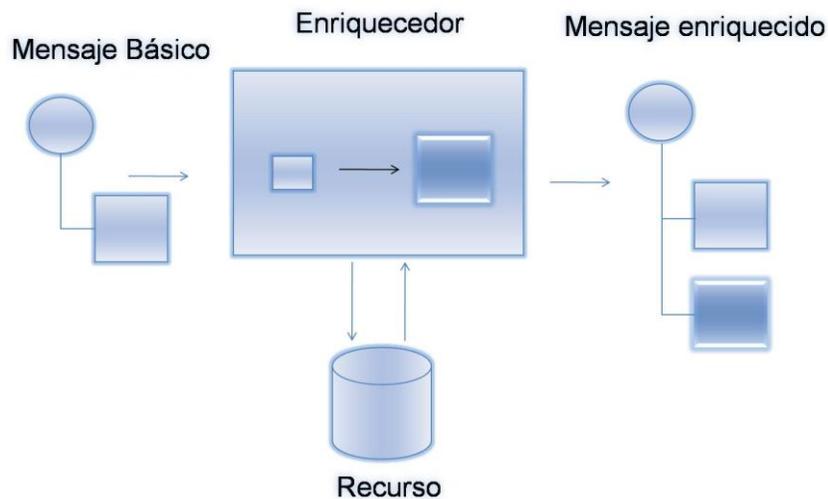


Figura 10. Patrón Enriquecedor de contenido.

Dicho patrón generalmente es usado con la intención de resolver las referencias contenidas en un mensaje, permitiendo mantener los mensajes pequeños y fáciles de manejar. Usualmente se considera pasar simples referencias a objetos en lugar de pasar un objeto completo con todos los elementos de datos. [10]

Cuando el sistema procesa un mensaje, tiende a recuperar los elementos de datos necesarios sobre la base de las referencias a objetos incluidos en el mensaje original. Consecuentemente se reduce el volumen de datos en los mensajes originales, pero requiere de búsquedas adicionales en el recurso. Sin embargo, si un mensaje pasa a través de varios intermediarios antes de llegar al receptor, con una referencia de objeto se puede disminuir significativamente el tráfico de mensajes. [10]

Con este propósito se puede insertar un enriquecedor de contenido antes de que el destinatario final

cargue la información que falta en el mensaje. Si el mensaje contiene datos que quizás no desee para llevar a lo largo del camino, podemos utilizar un ClaimCheck o Chequeo de Reclamo para almacenar los datos y obtener una referencia a él. [10]

Los patrones de **Chequeo de reclamo** y **Filtro de contenido**, son utilizados con el propósito de reducir la estructura de los mensajes. Sin embargo se recomienda utilizar un chequeo de reclamo, pues el mismo solo permite la eliminación de la información de manera temporal, pues la información eliminada se podría guardar en un almacén o fichero, que luego podría recuperarse. Sin embargo utilizar un filtro de contenido no garantiza que los datos eliminados sean restaurados posteriormente.

Normalizer o Normalizador: es utilizado con la intención de traducir los mensajes a un formato específico.

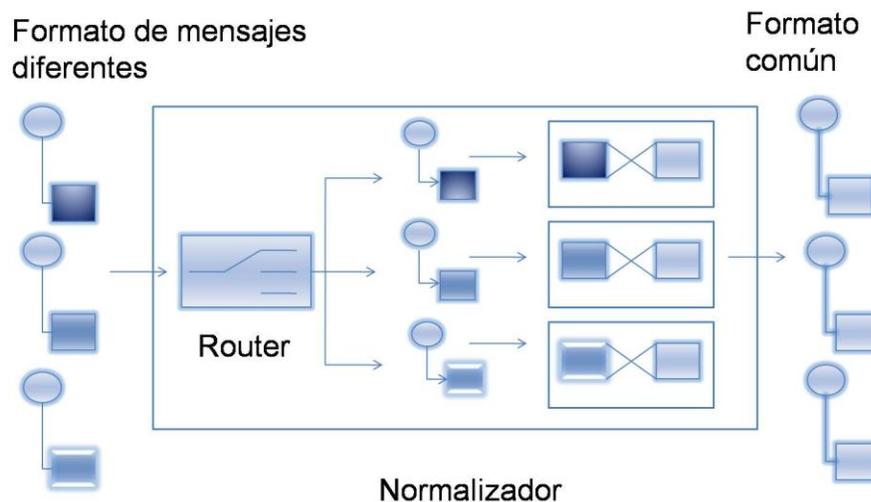


Figura 11. Patrón Normalizador.

En un entorno donde se desarrolla una SOA, los servicios de mensajería contribuyen al intercambio de datos entre diferentes aplicaciones. Para que el intercambio a nivel de mensajes sea lo más preciso posible, se recomienda utilizar los patrones antes descritos. Los cuales de cierta manera constituyen una guía para erradicar los diferentes problemas que vienen aparejados al utilizar información de otras aplicaciones.

2.3.7.2. Definir una arquitectura a nivel de datos

El próximo paso lo constituye la definición de una arquitectura a nivel de datos, el cual es considerado un requisito importante e imprescindible para lograr la integración a nivel de datos en una SOA. A partir del estudio realizado, la arquitectura de datos que se propone se ve reflejada en la siguiente figura.



Figura 12. Arquitectura de datos.

La arquitectura definida, fue concebida de manera genérica para un entorno SOA donde varias aplicaciones necesitan integrarse a nivel de datos. Con tal propósito se especifica la utilización de un ESB al cual podrían conectarse las aplicaciones. Su utilización permite proveer una infraestructura que consiste fundamentalmente en un servicio de mensajería al que se puede acceder por medio de diferentes protocolos de transporte. El mismo actuará como una capa encargada de gestionar las solicitudes realizadas por los consumidores. Entre las principales funciones que podría desarrollar se encuentran:

- Transportar mensajes entre los elementos de la infraestructura SOA.

- Soportar múltiples protocolos para el transporte de los mensajes.
- Ofrecer funciones de seguridad sobre los mensajes transportados.

Otro elemento incluido en la arquitectura definida es una Capa de Acceso a Datos, la cual actuará como un componente intermedio con la intención de respaldar el trabajo realizado por el ESB el cual gestiona los pedidos realizados por los clientes, delegando algunas de sus funciones a la misma. La capa de acceso a datos tiene como propósito obtener la información solicitada, residente en los Sistemas de Información Empresarial (EIS, por sus siglas del inglés) los cuales constituyen los medios de almacenamiento utilizados por las empresas. Entre dichos medios de almacenamiento, en caso que se haya adoptado un MDC, quedaría incluido el nuevo almacén de datos construido por los diseñadores de bases de datos y arquitectos.

La propuesta de utilizar un Almacén de Datos Operacional en la arquitectura de datos, pretende establecer un área intermedia que permita analizar la información. Dicho proceso incluye la limpieza de los datos así como la comprobación de los requisitos que deben cumplir para posteriormente ser depositados en dicho almacén. Un papel fundamental juegan los procesos ETL, los cuales se enfrasan en las funciones relacionadas con el movimiento de los datos.

Un elemento a tener en cuenta al definir la arquitectura es el de unificar la información contenida en los distintos sistemas de almacenamiento, o sea establecer una base de datos centralizada a partir de la información contenida en los mismos. Sin embargo la idea anterior sería posible solo en algunos escenarios de integración y conveniente desde determinado punto de vista. La situación precedente responde a la presencia de sistemas antiguos que se hallan en las empresas al momento del despliegue de la nueva infraestructura SOA, conocidos como sistemas legados.

En determinadas ocasiones los sistemas no llevan muchos años de utilidad y fueron construidos con tecnologías más recientes. De encontrarse en una situación parecida, se podría admitir la idea de construir un almacén de datos, que utilizase un único formato de datos. De lo contrario, si los sistemas son muy antiguos sería muy difícil y costoso realizar una maniobra así que unifique los datos. Por tales motivos, la opción más prudente sería tratar de interoperar con dichos sistemas, al establecer la nueva infraestructura. En tal caso sería necesaria la utilización de un Modelo de Mensaje Canónico en el ESB con el propósito de intercambiar información en un formato estándar a partir del uso de diferentes modelos de datos por las diferentes aplicaciones.

Por otro lado, cuando se requiere implantar una arquitectura SOA, no siempre se cuenta con el mismo escenario. Un ejemplo consiste en llevar a cabo dicho proceso a una empresa, donde existen varias aplicaciones con sus sistemas de almacenamiento. Si la información que utilizan en determinadas circunstancias es redundante y duplicada, se recomienda construir una base de datos centralizada, que permita registrar la información globalmente. Surge entonces la posibilidad de prescindir del uso de las fuentes de datos que antes eran utilizadas. En tal caso, se resolverían problemas de duplicidad, almacenamiento, actualización y seguridad.

Si la adopción de una SOA, abarca a más de una empresa, quizás no sea factible unificar la información en una nueva base de datos y eliminar las fuentes anteriores. Resulta probable que utilizar las fuentes de datos que se encontraban en su dominio sea más viable que recurrir a otra fuente ubicada lejos del ambiente de trabajo, lo cual ocasionaría problemas de acceso. En este caso resultaría útil emplear las 2 variantes, o sea, construir una base de datos donde la información se encuentre centralizada y a la vez que las aplicaciones continúen empleando sus fuentes originales. De forma tal que las aplicaciones manejen directamente sus fuentes y a la vez se actualice la nueva base de datos periódicamente. Se permitiría el uso de un mecanismo de replicación, con la intención de mantener una copia de la fuente que contiene la información centralizada. En caso de desastre en algunas de las empresas, donde la información se vea dañada, lograrían adquirir sus datos a partir de la misma. De igual manera si por alguna razón dejase de funcionar la base de datos que contiene la información centralizada, los datos pueden ser restaurados al estado previo al desastre a partir de una réplica.

El inconveniente de dicha solución radica en que se requiere de un tiempo apropiado para realizar todas las actualizaciones pertinentes. Se recomienda que el proceso se ejecute en un horario donde las aplicaciones no se encuentren generando un excesivo tráfico, es decir, la latencia de la red sea mínima.

2.3.7.3. Selección de Herramientas

Luego de definir la arquitectura es importante seleccionar las herramientas que ayudarán en el proceso de integración. En la arquitectura propuesta se especifica la utilización de un ESB, con tal propósito se recomienda utilizar el ESB de la suite WSO2 como bus de servicio, debido a que el CDAE, ya tiene experiencia en el trabajo con dicha herramienta además de ser una herramienta Open Source. Igualmente perteneciente a la suite de WSO2 se propone utilizar el Data Service Server, el cual estaría apoyando a la capa de acceso a datos en funciones relacionadas con la gestión de los datos. Para el manejo de los

procesos ETL, se recomiendan herramientas como Jitterbit, Talend Open Studio, Scriptella ETL Project, Pentaho Data Integration, que son herramientas libres.

Por otra parte, existen otras herramientas y plataformas muy completas y eficaces en la integración de datos desarrolladas por compañías reconocidas en la industria del software como son:

- Plataformas de Informatica Corporation.
- Plataforma de IBM.
- Oracle Data Integration Suite.

En caso de optar por herramientas propietarias, se recomienda el uso de algunas de estas soluciones, las cuales cuentan con herramientas para el manejo de procesos ETL, así como para la gestión, calidad, limpieza y virtualización de los datos.

2.3.7.4. Integrar los datos

El presente paso se fundamenta a partir del desarrollo de los pasos anteriormente descritos. El mismo se verá reflejado desde el momento en que se adopte un MDC, pasando por la definición de una arquitectura a nivel de datos y posteriormente seleccionando las herramientas que apoyarán el proceso de integración.

La implementación de cómo llevar a cabo la integración se realizará y dependerá en gran medida de las características de cada una de las empresas que se encuentren inmersas en dicho proceso, así como de los modelos definidos y las herramientas utilizadas en la arquitectura especificada.

2.4. Conclusiones

En el presente capítulo se realizó la propuesta de una guía para facilitar la integración entre diferentes aplicaciones a nivel de datos en una Arquitectura Orientada a Servicios. Con la proposición de la misma se persigue darle cumplimiento al objetivo principal de la investigación. Por tales motivos se expone un flujo de pasos, los cuales son analizados y descritos detalladamente, con la finalidad de facilitar la integración. Por otra parte se describieron algunas técnicas utilizadas para consumir los pasos definidos. Además se realizó la propuesta de una arquitectura de datos para una SOA, donde se explican las tecnologías y componentes seleccionados para el desarrollo de la misma.

CAPÍTULO 3 “VALIDACIÓN DE LA PROPUESTA”

3.1. *Introducción*

Luego de realizar la propuesta de una guía que permita facilitar la integración de datos en una SOA, se hace necesario su validación y aceptación. El presente capítulo tiene como objetivo principal validar la guía propuesta, mediante el método Delphi, utilizando el Método de expertos. Se estará realizando una descripción del mismo, permitiendo que se obtenga un análisis de los resultados de su aplicación. Con dicho propósito se trabajara con un grupo de expertos que a partir de su criterio se podrá determinar si se encuentra en condiciones de ser aplicada.

3.2. *Método Delphi*

El método Delphi, creado por la compañía Rand Corporation, es considerado como una de las técnicas subjetivas de pronosticación más confiable. Fue creado con el objetivo de elaborar pronósticos a largo plazo referentes a posibles acontecimientos que podrían tener lugar en varias ramas de la ciencia, la técnica y la política. De esta manera, se define como la utilización sistémica del juicio intuitivo de un grupo de expertos, para obtener un grupo de opiniones informadas. [37]

Se encuentra fundamentado por la elaboración estadística de las opiniones del grupo de expertos en el tema tratado donde se reflejan cada una de las valoraciones de los expertos, las cuales pueden estar basadas en análisis lógico o en su propia experiencia intuitiva.

Entre sus principales características se encuentran:

Anonimato: ningún experto conoce la identidad de los otros que componen el grupo seleccionado.

Iteración y realimentación controlada: cuando se muestra el mismo cuestionario varias veces se evidencia la iteración. Además, los resultados obtenidos con los cuestionarios anteriores se van presentando, de esta forma los expertos conocen los diferentes puntos de vista y pueden ir modificando su opinión si consideran que los argumentos presentados son más apropiados que los suyos.

Respuesta del grupo en forma estadística: a los expertos no se les presenta, solamente, la información del punto de vista de la mayoría, sino todas las opiniones, de esta forma se muestra el grado de acuerdo que se ha obtenido.

Heterogeneidad: los expertos que conforman el grupo seleccionado pueden pertenecer a distintas ramas de actividad sobre las mismas bases. [38]

La calidad de los resultados obtenidos por este método dependen de:

- La elaboración de los cuestionarios.
- Las predicciones de los expertos consultados.

Mediante encuestas o entrevistas se pueden obtener el criterio de los expertos. Además de que pueden ser empleados en cualquier momento de la investigación. Su aplicación ofrece un conjunto de ventajas, las cuales se reflejan a continuación [39]:

- Permite la formación de un criterio con mayor grado de objetividad y el consenso logrado sobre la base de los criterios es muy confiable.
- La tarea de decisiones sobre la base de los criterios de expertos, obtenido por éste tiene altas probabilidades de ser eficiente.
- Permite valorar alternativas de decisión.
- Un requisito imprescindible para garantizar el éxito del método, evitar conflictos entre expertos y crear un clima favorable a la creatividad, es ser anónimo.
- El experto se siente involucrado plenamente en la solución del problema y facilita su implantación.
- De ello es importante el principio de voluntariedad del experto en participar en la investigación y la confidencialidad de su opinión.

Para realizar la validación, el método Delphi no será aplicado en su totalidad. Se estará utilizando una variante propuesta por Silvia Colunga y Georgina Amayuela [40], que consta de algunas características del mismo que permitirán propiciar mayor objetividad a los criterios de los especialistas a partir de la introducción de escalas valorativas.

Para llevar a cabo la aplicación del método es necesario tener presente tres fases fundamentales, las cuales se muestran a continuación:

- Selección de los expertos.
- Elaboración y aplicación de las encuestas, para la validación de la propuesta.

- Procesamiento de la información y explotación de resultados.

A continuación se manifiesta como será aplicado el método en la investigación actual.

3.3. Aplicación del método Delphi

El primer paso en la aplicación del método, consiste en la selección de un grupo de personas a los cuales se les aplica una encuesta que permita medir, a partir de los resultados obtenidos, el grado de conocimiento que el encuestado posee relacionado con el tema, a través del Cálculo del Coeficiente de Competencia que define el método Delphi.

3.3.1. Selección de los expertos

Se considera un experto a una persona o un grupo de personas que tienen alguna experiencia, que les permite emitir un criterio concluyente de cualquier problema y exponer valoraciones importantes con un alto nivel de conocimiento. [37]

Por tanto para la presente investigación un experto es considerado aquella persona que sea capaz de brindar valoraciones y recomendaciones respecto a la integración de datos en una Arquitectura Orientada a Servicios.

El método Delphi, no establece un valor para el número de expertos a seleccionar. Sin embargo los investigadores de Rand Corporation opinan que es preciso utilizar un mínimo de 7 y un máximo de 30 expertos.

Cumpliendo con los parámetros establecidos por los investigadores de Rand Corporation, fueron escogidos 10 posibles expertos, para los cuales se tuvieron en cuenta los siguientes criterios para su selección:

- Graduado del Nivel Superior.
- Experiencia laboral. Un año de experiencia como mínimo.
- Capacidad de análisis y pensamiento lógico.
- Vinculación al desarrollo de proyectos productivos.
- Conocimientos y habilidades sobre Arquitectura Orientada a Servicios (SOA).

- Conocimientos sobre integración de datos en una SOA.
- Prestigio en el colectivo de trabajo.
- Disposición para participar en la validación.

Los criterios relacionados anteriormente, permitirán que las opiniones brindadas por el grupo de posibles expertos sean válidas. Sin embargo se tendrán en cuenta cualidades como sinceridad y responsabilidad con el propósito de que sean más confiables. Los posibles candidatos fueron profesores y trabajadores de la Universidad de las Ciencias Informáticas (UCI) con más de un año de experiencia y con conocimientos de SOA.

Luego de haberse determinado los expertos, cumpliendo con los requisitos propuestos, se realizó una encuesta de autoevaluación para medir el grado de conocimiento de cada uno, con el objetivo de seleccionar a los que conozcan más del tema que se está investigando. La encuesta se aplicó a 10 profesores pertenecientes al CDAE y SIAI, y de ellos se escogieron una muestra de 8 expertos. El listado con los expertos seleccionados se muestra en el **Anexo 3**, mientras que la encuesta de autovaloración es mostrada en el **Anexo 4**.

A partir de los resultados de la encuesta se puede determinar el nivel de conocimiento que posee el experto, mediante el cálculo del Coeficiente de Competencia, que define el método Delphi.

El objetivo de calcular el coeficiente de competencia es conocer si el nivel de conocimientos que poseen los expertos así como las fuentes de argumentación son las apropiadas, de manera tal que garantice la confiabilidad en los resultados.

Para calcular el coeficiente de competencia (K) se utiliza la fórmula:

$$K=1/2 (Kc +Ka)$$

Figura 13. Fórmula para obtener el coeficiente de conocimiento.

Donde Kc es el Coeficiente de Conocimientos y Ka el Coeficiente de Argumentación. El Coeficiente de Conocimientos se corresponde con la información que posee la persona acerca del tema que se investiga, por lo que sus valores oscilan de 0 a 10 y se multiplica por 0,1.

El coeficiente de conocimiento (Kc) se obtiene de la primera tabla que aparece en el formulario.

0	1	2	3	4	5	6	7	8	9	10
								X		

Tabla 1. Obtener el coeficiente de conocimientos.

En caso que se halla marcado un 0, significa que el experto no posee conocimiento sobre el tema y si marcó un 10, implica que el experto disfruta de un amplio conocimiento. Para el ejemplo mostrado en la **Tabla 1** el experto posee un $Kc = 0,8$.

El coeficiente de argumentación (Ka) se obtiene del resultado de la suma de los puntos alcanzados a partir de las respuestas obtenidas de la segunda pregunta realizada en la encuesta de autovaloración. En la tabla expuesta en la 2da pregunta, el experto debe marcar, cuáles fueron las fuentes que en su consideración, le permitió elegir su evaluación respecto al nivel de conocimiento especificado en la **Tabla 1**.

Sin embargo el método Delphi asigna un valor para cada una de las posibles respuestas de los expertos, para calcular el valor de Ka. A continuación se muestra la tabla con los valores asignados:

No	Fuentes de Argumentación	Grado		
		Alto	Medio	Bajo
1	Experiencia	0.5	0.4	0.2
2	Documentación válida y elaborada por expertos nacionales.	0.05	0.05	0.05
3	Documentación válida y elaborada por expertos extranjeros.	0.05	0.05	0.05

4	Análisis teóricos realizados por usted.	0.3	0.2	0.1
5	Su propio conocimiento	0.05	0.05	0.05
6	Su intuición	0.05	0.05	0.05
Totales		0.1	0.8	0.5

Tabla 2. Valores asignados para determinar Ka.

Quando el experto realiza la encuesta, la **Tabla 2**, aparece sin los valores asignados por el método Delphi. Solo debe enfocarse en marcar con una cruz todas las fuentes, utilizando los grados Alto (A), Medio (M), Bajo (B) que hayan influido en el conocimiento adquirido. Posteriormente utilizando los valores de la **Tabla 2**, se calcula el valor de Ka. Si el experto al realizar la encuesta, seleccionó de la siguiente manera:

No.	Fuentes de argumentación	Grado de influencia		
		Alto	Medio	Bajo
1.-	Experiencia.	x		
2.-	Documentación válida y elaborada por expertos nacionales.		x	
3.-	Documentación válida y elaborada por expertos extranjeros.		x	
4.-	Análisis teóricos realizados por usted.		x	
5.-	Su propio conocimiento del tema.		x	
6.-	Su intuición.		x	

Tabla 3. Ejemplo de la selecciones de un experto.

El coeficiente de argumentación sería:

$$K = 1/2 (0.8 + 0.9) = 0.85$$

Una vez calculado los valores de Kc y Ka, se podría calcular el coeficiente de competencia. El resultado

debería interpretarse mediante los siguientes intervalos:

- Si $0,8 < k < 1,0$ el coeficiente de competencia es Alto
- Si $0,5 < k < 0,8$ el coeficiente de competencia es Medio
- Si $k < 0,5$ el coeficiente de competencia es Bajo

De los 10 expertos iniciales a los que se les aplicó la Encuesta de Autovaloración, fueron seleccionados 8 para continuar con la ejecución del método. Para formar parte de dicho grupo, fueron escogidos aquellos expertos que presentaban un coeficiente de competencia Medio y Alto. En la siguiente tabla se muestran los resultados obtenidos en la encuesta de autovaloración:

Expertos	Kc	Ka	K	Competencia
E1	0.7	0.8	0.75	Medio
E2	1.0	1.0	1.0	Alto
E3	0.6	0.8	0.7	Medio
E4	0.2	0.7	0.45	Bajo
E5	0.5	0.7	0.6	Medio
E6	0.4	0.5	0.45	Bajo
E7	0.5	0.8	0.65	Medio
E8	0.5	0.8	0.65	Medio
E9	0.6	0.8	0.7	Medio
E10	0.8	0.9	0.85	Alto

Tabla 4. Resultados de la encuesta de autovaloración.

Los expertos 4 y 6 dejarán de formar parte de la selección para la validación de la propuesta, debido a que su Coeficiente de Competencia es Bajo. Por tanto el grupo para validar la propuesta estará conformado por 8 expertos. La relación de los expertos finales, se muestra en el **Anexo 5**.

3.3.2. Elaboración del cuestionario para validar la propuesta

A partir de la selección de los expertos, se procede a validar la guía para facilitar la integración entre diferentes aplicaciones a nivel de datos en una Arquitectura Orientada a Servicios. El cuestionario fue elaborado teniendo en cuenta que las preguntas fueran reveladas en muy adecuado (MA), bastante adecuado (BA), adecuado (A), poco adecuado (PA) y no adecuado (NA). El cuestionario aplicado se muestra en el **Anexo 6**.

En la confección del cuestionario que se muestra en el **Anexo 6**, se tuvieron en cuenta los siguientes objetivos:

- Demostrar que los pasos que se proponen son útiles y necesarios para cumplir con los objetivos de la presente investigación. (Pregunta 2)
- Demostrar la efectividad de los pasos en la guía propuesta. (Pregunta de la 1.1 a la 1.7)
- Demostrar que la Arquitectura de datos presentada es suficiente para cumplir con los objetivos trazados. (Pregunta 3)
- Conocer la opinión de los expertos en relación a la guía propuesta. (Pregunta 4)
- Identificar aspectos negativos y observaciones que permitan mejorar la guía presentada. Al final de la encuesta se hallaba un apartado dedicado a dar una valoración general de la guía.

3.3.3. Desarrollo práctico y explotación de los resultados

Las encuestas fueron enviadas a los 8 expertos seleccionados, donde se le explicó a cada uno, los objetivos y las condiciones prácticas para el desarrollo de la encuesta (plazo de respuesta y garantía de anonimato). Al obtener los resultados, se procedió al análisis de las mismas, con el propósito de conocer el nivel de aprobación de la guía propuesta.

Los resultados se recogen en una tabla de frecuencias absolutas, utilizándose en la confección de las mismas, el programa Microsoft Excel 2010. Los números asignados representan la cantidad de expertos que valoraron los diferentes aspectos según la escala asignada. Las evaluaciones se corresponden con los valores en la escala de muy adecuado (MA), bastante adecuado (BA), adecuado (A), poco adecuado (PA) y no adecuado (NA). Los resultados son recogidos en la siguiente tabla:

Tabla de frecuencias absolutas							
No	Elementos	MA	BA	A	PA	NA	Total
1	A1	1	4	3	0	0	8
2	A2	6	1	1	0	0	8
3	A3	5	3	0	0	0	8
4	A4	2	3	3	0	0	8
5	A5	1	2	5	0	0	8
6	A6	3	5	0	0	0	8
7	A7	2	5	1	0	0	8
8	A8	1	5	2	0	0	8
9	A9	1	2	5	0	0	8
10	A10	2	4	2	0	0	8
Total de aspectos a evaluar 10							

Tabla 5. Tabla de frecuencias absolutas.

Luego de recoger todos los datos, se siguieron un conjunto de pasos para obtener los resultados. A continuación se presentan los pasos que se siguieron:

Primer paso: se construye una tabla de frecuencias acumuladas, donde el número de una fila se obtiene sumándole el anterior, excepto el primero.

Tabla de frecuencias acumuladas							
No	Elementos	MA	BA	A	PA	NA	Total

1	A1	1	5	8	8	8	8
2	A2	6	7	8	8	8	8
3	A3	5	8	8	8	8	8
4	A4	2	5	8	8	8	8
5	A5	1	3	8	8	8	8
6	A6	3	8	8	8	8	8
7	A7	2	7	8	8	8	8
8	A8	1	6	8	8	8	8
9	A9	1	3	8	8	8	8
10	A10	2	6	8	8	8	8

Tabla 6. Tabla de frecuencias acumuladas.

A continuación en la **Tabla 7** se muestra la cantidad de valoraciones proporcionadas por los expertos a partir de los diferentes aspectos según la escala asignada. Mientras que la Figura 14, muestra el porcentaje que las mismas representan.

Muy Adecuado	24
Bastante Adecuado	34
Adecuado	22
Poco Adecuado	0
No Adecuado	0

Tabla 7. Cantidad de valoraciones por los expertos.

Frecuencias Acumuladas

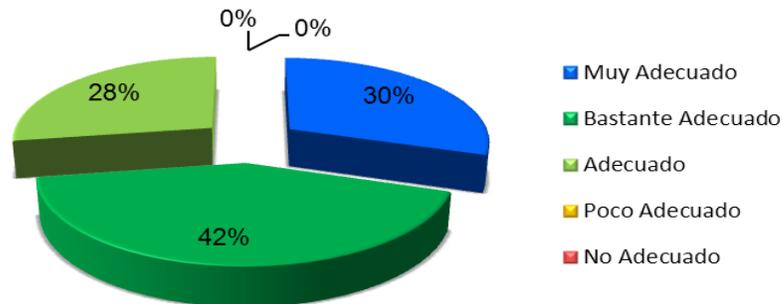


Figura 14. Frecuencias Acumuladas.

Segundo paso: se construye una nueva tabla, la cual contendrá las frecuencias relativas acumuladas. Los valores de la nueva tabla se obtienen dividiendo cada valor de la tabla anterior, entre el número total de expertos. Al llenar la tabla, es necesario tener en cuenta que la última columna es obviada, pues sólo se necesitan 4 puntos de corte para obtener 5 intervalos.

Tabla de frecuencias relativas acumuladas					
No	Elementos	MA	BA	A	PA
1	A1	0.125	0.625	1	1
2	A2	0.75	0.875	1	1
3	A3	0.625	0.9999	1	1
4	A4	0.25	0.625	1	1
5	A5	0.125	0.375	1	1
6	A6	0.375	0.9999	1	1
7	A7	0.25	0.875	1	1

8	A8	0.125	0.75	1	1
9	A9	0.125	0.375	1	1
10	A10	0.25	0.75	1	1

Tabla 8. Tabla de frecuencias relativas acumuladas.

Tercer paso: Se buscan las imágenes de los elementos de la tabla anterior por medio de la función (DISTR.NORM.ESTAND.INV). A la tabla se le agregan tres nuevas columnas y una fila para colocar los valores siguientes:

- Suma de las columnas.
- Suma de filas.
- Promedio de las filas (P). Se obtiene al dividir entre 4 (número de categorías) la suma de todas las filas.
- Valor de N, se halla al dividir la suma de las sumas entre el resultado de multiplicar el número de indicadores por el número de preguntas.
- El valor N-P es el valor promedio que otorgan los expertos para cada indicador propuesto.

En la siguiente tabla se muestran los resultados obtenidos:

Puntos de Corte	N =	1,44	Grado de
------------------------	------------	-------------	-----------------

No	Elementos	C1	C2	C3	C4	Suma	P	N-P	Adecuación
1	A1	-1,15	0,32	3,72	3,72	6,61	1,65	-0,21	Bastante Adecuado
2	A2	0,67	1,15	3,72	3,72	9,26	2,32	-0,87	Muy Adecuado
3	A3	0,32	3,72	3,72	3,72	7,76	2,59	-1,14	Muy Adecuado
4	A4	-0,67	0,32	3,72	3,72	7,08	1,77	-0,33	Bastante Adecuado
5	A5	-1,15	-0,32	3,72	3,72	5,97	1,49	-0,05	Bastante Adecuado
6	A6	-0,32	3,72	3,72	3,72	7,12	2,37	-0,93	Muy Adecuado
7	A7	-0,67	1,15	3,72	3,72	7,92	1,98	-0,54	Bastante Adecuado
8	A8	-1,15	0,67	3,72	3,72	6,96	1,74	-0,30	Bastante Adecuado
9	A9	-1,15	-0,32	3,72	3,72	5,97	1,49	-0,05	Bastante Adecuado
10	A10	-0,67	0,67	3,72	3,72	7,44	1,86	-0,42	Bastante Adecuado
Suma		-5,95	3,65	37,20	37,20	72,10	Total MA		3
Puntos de Corte		-0,60	0,46	3,72	3,72		Total BA		7

Tabla 9. Puntos de corte

Las sumas obtenidas en las cuatro primeras columnas nos dan los puntos de cortes (-0.60, 0.46, 3.72, 3.72), los cuales permiten determinar la categoría o grado de adecuación según la opinión de los expertos consultados.

La Figura 15 muestra la categoría resultante partir de los resultados de la validación realizada por los expertos. Con el propósito de determinar si la misma es muy adecuada, bastante adecuada, adecuada o poco adecuada. Los resultados se obtendrían a partir de los siguientes intervalos:

- Muy Adecuado para valores menores a -0.60.
- Bastante Adecuado para valores entre – 0.60 y 0.46.
- Adecuado entre 0.46 y 3.72.
- Poco Adecuado los mayores de 3.72

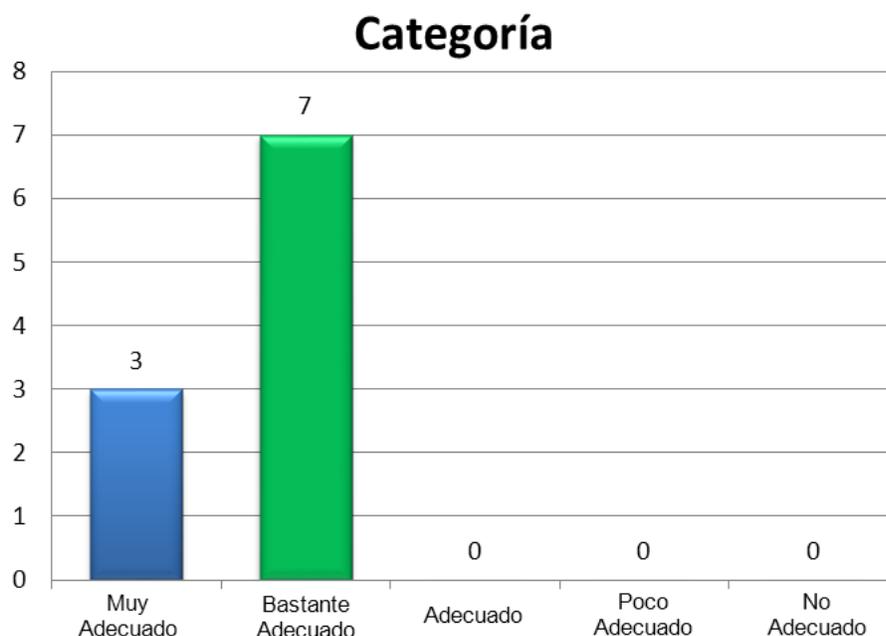


Figura 15. Categoría resultante.

Conforme a los efectos obtenidos en la encuesta, se puede llegar a la conclusión que la validación resulta Bastante Adecuada. Por tales motivos se considera, que a partir de resultados satisfactorios obtenidos, los pasos presentados quedan incluidos dentro de la guía propuesta, así como su validación.

3.4. Conclusiones

El presente capítulo estuvo dirigido a realizar la validación de la guía propuesta para facilitar la integración de diferentes aplicaciones a nivel de datos en una Arquitectura Orientada Servicios. Dicha validación fue realizada a partir de la aplicación del método de validación Delphi. Para su aplicación fue necesario realizar un proceso de selección de expertos, para lo que se tuvo en cuenta que los mismos tuvieran conocimiento y experiencia acerca del tema. En definitiva fueron seleccionados 10 expertos y a partir del análisis de los cuestionarios aplicados a ellos, los resultados obtenidos pueden catalogarse como satisfactorios.

En relación a las opiniones, se puede concluir que la propuesta fue validada por la totalidad del grupo de expertos seleccionados, sin la ocurrencia de abstenciones, comprobando la veracidad de la propuesta.

CONCLUSIONES GENERALES

Se considera que la presente investigación ha cumplido con los objetivos propuestos, pues:

- Se elaboró el marco teórico sobre el estado del arte referente a la integración de datos en una Arquitectura Orientada a Servicio.
- Se realizó un análisis de las definiciones, técnicas y herramientas relacionado con la integración de datos. Donde se caracterizaron los aspectos más importantes asociados al dominio del problema, aportando los conocimientos necesarios para realizar una propuesta sólida.
- Se definió una guía para facilitar la integración de diferentes aplicaciones a nivel de datos en una SOA, proponiéndose un conjunto de pasos que incluyen técnicas, arquitectura, consejos y patrones.
- Se validó la guía propuesta mediante el método Delphi, utilizando como elementos de certificación los criterios y experiencias de un grupo de expertos, los cuales catalogaron positivamente los pasos que la conforman. La misma fue categorizada de Bastante Adecuada para facilitar la integración a nivel de datos en una SOA, permitiendo minimizar el número de errores durante el proceso de integración.

RECOMENDACIONES

Teniendo en cuenta los resultados obtenidos una vez concluida la propuesta de solución, se sugiere:

- Aplicar la guía propuesta en futuros proyectos que asuma el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales, con el propósito de obtener resultados reales.
- Considerar las posibilidades de extender la guía propuesta a otros centros que utilizan una Arquitectura orientada a Servicios.
- Continuar con el estudio referente la integración de datos en una SOA, que permita agilizar y facilitar la integración de diferentes aplicaciones, así como establecer mejoras a la solución propuesta.

REFERENCIAS BIBLIOGRÁFICAS

1. Corporation, M. *La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada al mundo real*. 2006 [cited; Available from: http://www.findthatfile.com/search-4840203-hPDF/download-documents-070717-real_world_soa.pdf.htm].
2. Jiménez, G.M.C. *Tecnologías de web semántica orientada al desarrollo de servicios web*. 2010 [cited; Available from: <http://cepra.utpl.edu.ec/handle/123456789/1706>].
3. Colan, M. *Service-Oriented Architecture expands the vision of web services, Part 1*. 2004 [cited 2012 Enero]; Available from: <http://www.ibm.com/developerworks/library/ws-soaintro/index.html>.
4. Pineda, J.C.R. *Soa y web services*. 2011 [cited; Available from: <http://www.slideshare.net/jcrubio/03-soa-y-web-services>].
5. Technologies, K. *Adaptación a SOA*. 2009 [cited 2012 5 de Junio]; Available from: <http://www.kapowtech.es/soa-y-woa.htm>.
6. Archilla, P.R. *Web services y arquitecturas orientadas al servicio*. 2010 [cited; Available from: [ofiweb.ugr.es/pages/prtweb/webservices/!](http://ofiweb.ugr.es/pages/prtweb/webservices/)].
7. Smalltree, H. *ESB Tutorial*. 2011 [cited 2012 Mayo]; Available from: <http://searchsoa.techtarget.com/tutorial/ESB-Tutorial>.
8. Abellán, A.G.d.S.y.S.J. *Arquitectura SOA para la integración entre software libre y software propietario en entornos mixtos*. [cited 2012 Marzo]; Available from: <http://www.geograma.com/es/publicaciones/arquitectura-soa-para-la-integraci-n-entre-software-libre-y-software-propietario-en-entornos-mixtos.html>.
9. Inc, T.S. *El papel de un bus de servicios empresariales (ESB) en una SOA*. 2006 [cited; Available from: <http://ebookbrowse.com/2008-06-08-articulo-el-papel-de-un-bus-de-servicios-empresariales-esb-en-una-soa-pdf-d309169436>].
10. Gregor Hohpe, B.W., *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions* 2003: Addison Wesley.
11. Hudson, C. *How to create a canonical data model?*. 2007 [cited; Available from: <http://reallifefserviceorientedarchitecture.blogspot.com/2007/11/how-to-create-canonical-form.html>].
12. Ernesto Marquina, J.D.P. *Guía de Patrones, Prácticas y Arquitectura .NET*. 2008 [cited].
13. Tools, E. *DATA WAREHOUSING*. [cited; Available from: <http://www.etltools.org/data-warehousing.html>].
14. Alfonso. *Definición de Bases de Datos*. [cited 2012 Febrero]; Available from: <http://www.dataprix.com/22-definici-n-bases-datos>.
15. Lapuente, M.J.L. *Hipertexto, el nuevo concepto de documento en la cultura de la imagen*. 2011 [cited 2012 Febrero]; Available from: <http://www.hipertexto.info>.
16. Paredes, C. *Diseño de una Data Warehouse*. 2012 [cited 2012 Marzo]; Available from: <http://ladatawarehouse.blogspot.com/>.
17. Sinnexus. *Datawarehouse*. 2011 [cited 2012 Marzo]; Available from: http://www.sinnexus.com/business_intelligence/datawarehouse.aspx.
18. Curto, J. *Data Warehousing, Data Warehouse y Datamart*. 2007 [cited 2012 Febrero]; Available from: url:<http://informationmanagement.wordpress.com/2007/10/07/data-warehousing-data-warehouse-y-datamart/>.
19. Robles, J.d.J.L. 2011 [cited 2012 Marzo]; Available from: <http://www.buenastareas.com/ensayos/Dataware-House/3575796.html>.

20. Oracle, S. *Operational Data Store (ODS)*. 2009 [cited 2012 Febrero]; Available from: <http://searchoracle.techtarget.com/definition/operational-data-store>.
21. Sinnexus. *Datamart*. 2011 [cited 2012 Marzo]; Available from: http://www.sinnexus.com/business_intelligence/datamart.aspx
22. ETL-Tools.Info. *Proceso ETL*. 2011 [cited 2012 Marzo]; Available from: http://etl-tools.info/es/bi/proceso_etl.html.
23. Agenda, S. *Orquestadores y SOA*. 2007 [cited 2012 Marzo]; Available from: <http://www.soaagenda.com/journal/articulos/orquestadores-y-soa/>.
24. Gómez, L.F., *Interoperabilidad en los sistemas de información documental*. 2007. 3 p. 18.
25. Tecnologías, D.G.d.T.y.N. (2009) *Marco de referencia para la plataforma de interoperabilidad. Volume*,
26. Jiménez, J.I.C.M.G.C.E., *Bases para una Estrategia Iberoamericana de Interoperabilidad*. 2010.
27. Carlos. *Informatica 9, una completa plataforma de Integración de Datos*. 2010 [cited 2012 Marzo]; Available from: <http://www.dataprix.com/informatica-9-completa-plataforma-integracion-datos>.
28. Soto, D. *El portfolio de IBM (IBM Information Management Software)*. 2010 [cited 2012 Marzo]; Available from: <http://integracionycalidad.blogspot.com/2010/02/el-portfolio-de-ibm-ibm-information.html>.
29. Oracle. *Oracle Data Integration*. [cited 2012 Abril]; Available from: <http://www.oracle.com/es/products/middleware/data-integration/index.html>.
30. Oracle. *Oracle Data Integration Suite*. [cited 2012 Abril]; Available from: <http://www.oracle.com/us/products/middleware/data-integration/data-integration-suite/index.html?ssSourceSiteId=ocomes>.
31. Oracle. *ORACLE SOA SUITE*. 2008 [cited 2012 Abril]; Available from: <http://www.oracle.com/technetwork/es/middleware/soasuite/documentation/oracle-soa-suite-427128-esa.pdf>.
32. Inc, W. *Welcome to WSO2 Enterprise Service Bus (ESB) v4.0.3* 2011 [cited 2012 Mayo].
33. Inc, W. *Welcome to WSO2 Data Services Server, 2.6.2*. 2011 [cited 2012 Mayo].
34. Silva, R.O. *Estandarización de estructuras de datos para la adopción de SOA*. 2012 [cited 2012 Marzo]; Available from: <http://joviedosoa.blogspot.com/2012/01/estandarizacion-de-estructuras-de-datos.html>.
35. Ideas, E. *SOA and Canonical Data Model (CDM)*. [cited 2012 Mayo]; Available from: <http://www.eai-ideas.com/architecture-ideas/soa-and-canonical-data-model-cdm>.
36. Mei Selvage, G.F., Dr. Guenter Sauter, Dr. Eoin Lane *Exploring the Enterprise Service Bus, Part 3: Four approaches to implementing a canonical message model in an ESB*. 2008 [cited 2012 Abril]; Available from: <http://www.ibm.com/developerworks/webservices/library/ar-esbpat3/>.
37. Vicenta Tabera Urgellés, Y.A.J., *Proceso para la identificación de servicios en el contexto de una arquitectura SOA*. 2010, Universidad de las Ciencias Informáticas: Ciudad de La Habana,. p. 80.
38. Serrano, V.C., *Modelo de Aseguramiento de Calidad durante el desarrollo de proyectos bajo una iniciativa SOA*. 2010, Universidad de las Ciencias Informáticas: Ciudad de la Habana,. p. 80.
39. Dianella Rosales Pérez, R.M.C., *Propuesta de Modelo de Estructura y Evolución para un Centro de Excelencia SOA*. 2010: Ciudad de la Habana,. p. 79.
40. Amayuela, S.C.y.G., *La Psicología Educativa, su objeto, métodos y problemas principales*. 2003, Universidad de Camagüey: Camagüey.