



UNIVERSIDAD JOSÉ ANTONIO PÁEZ

**DESARROLLO DE UN SOFTWARE DE INTERFACES  
QUE PERMITA IMPLEMENTAR UN MODELO DE  
INTEGRACIÓN PARA LA INTEROPERABILIDAD  
DE LOS SISTEMAS MEDIANTE PATRONES  
DE INTEGRACIÓN EMPRESARIALES.**

**Autor:**  
Victor D. Archila M.  
20.698.042

Urb. Yuma II, calle N° 3. Municipio San Diego,  
Teléfono: (0241) 8714240 (master) – Fax: (0241) 8712394



**REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERIA DE COMPUTACIÓN**

**DESARROLLO DE UN SOFTWARE DE INTERFACES QUE PERMITA  
IMPLEMENTAR UN MODELO DE INTEGRACIÓN PARA LA  
INTEROPERABILIDAD DE LOS SISTEMAS MEDIANTE PATRONES DE  
INTEGRACION EMPRESARIALES.**

Trabajo de Grado presentado como requisito parcial para optar al título de  
**INGENIERO EN COMPUTACIÓN**

**Autor:**  
Victor D. ArchilaM.  
C.I. 20.698.042

**Tutor:**  
Soraya Aguilar

San Diego, Septiembre de 2015



REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERIA DE COMPUTACIÓN

### ACEPTACIÓN DEL TUTOR

Quien suscribe, Ingeniera Soraya Aguilar, portadora de la cédula de identidad N° 13.509.090, en mi carácter de tutor del trabajo de grado presentado por el ciudadano Victor David Archila Mendoza, portador de la cédula de identidad N° 20.698.042, titulado **DESARROLLO DE UN SOFTWARE DE INTERFACES QUE PERMITA IMPLEMENTAR UN MODELO DE INTEGRACIÓN PARA LA INTEROPERABILIDAD DE LOS SISTEMAS MEDIANTE PATRONES DE INTEGRACIÓN EMPRESARIALES**, presentado como requisito parcial para optar al título de Ingeniero en Computación, considero que dicho trabajo reúne los requisitos y méritos suficientes para ser sometido a la presentación pública y evaluación por parte del jurado examinador que se designe.

En San Diego, a los días del mes de Septiembre del año 2015.

---

Lic. Soraya Aguilar.

C.I. 13.509.090



REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERIA DE COMPUTACIÓN

San Diego, Septiembre de 2015

### ACTA DE REVISIÓN DEL PROYECTO DE TRABAJO DE GRADO

Quienes suscriben esta Acta, dejan constancia que el Proyecto de Trabajo de Grado: **DESARROLLO DE UN SOFTWARE DE INTERFACES QUE PERMITA IMPLEMENTAR UN MODELO DE INTEGRACIÓN PARA LA INTEROPERABILIDAD DE LOS SISTEMAS MEDIANTE PATRONES DE INTEGRACIÓN EMPRESARIALES** ha sido revisado y, cumpliendo con los requisitos exigidos para su aprobación, recomiendan su tramitación ante el organismo académico correspondiente.

Ing. Soraya Aguilar

\_\_\_\_\_

\_\_\_\_\_

Tutor Académico

Firma

Fecha

Ing. Alicia Pizzella

\_\_\_\_\_

\_\_\_\_\_

Tutor Metodológico

Firma

Fecha



REPÚBLICA BOLIVARIANA DE VENEZUELA  
UNIVERSIDAD JOSÉ ANTONIO PÁEZ  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERIA DE COMPUTACIÓN

## DESARROLLO DE UN SOFTWARE DE INTERFACES QUE PERMITA IMPLEMENTAR UN MODELO DE INTEGRACIÓN PARA LA INTEROPERABILIDAD DE LOS SISTEMAS MEDIANTE PATRONES DE INTEGRACIÓN EMPRESARIALES

**Autor:** Victor D. Archila M.

**Tutor:** Soraya Aguilar

**Fecha:** Septiembre, 2015

### RESUMEN

Una de las principales necesidades que enfrentan hoy los negocios es la de integrar sus procesos en forma “transversal” a través de los diferentes aplicativos. Tradicionalmente la integración de los procesos se llevó a cabo, en algunos casos manualmente, haciendo difícil el poder modificar ágilmente los procesos o el crear procesos nuevos reutilizando los componentes existentes. Como resultado de estos esfuerzos surge la Arquitectura orientada a servicios (SOA), que permite construir aplicaciones e integrar las existentes a partir de servicios. La herramienta que permite la integración entre servicios, son los ESB (BUS de Servicio Empresarial). Este trabajo se enmarca dentro de la modalidad de proyecto especial. El desarrollo del sistema se realizó bajo la metodología de Arquitectura de Modelado Orientada a Servicios (SOMA). El resultado, un modelo de integración para la interoperabilidad de los sistemas mediante patrones de integración empresariales.

**Palabras Claves:** Integración, SOA, API, SOMA, ESB, MQ, Mediación

## INDICE GENERAL

|  |    |
|--|----|
| RESUMEN.....   | ix |
| INTRODUCCIÓN .....                                       | 1  |
| CAPITULO   |    |
| I EL PROBLEMA  |    |
| .....  | 11 |
| 1.1 Planteamiento del Problema.....                      | 11 |
| 1.2 Formulación del Problema .....                       | 13 |
| 1.3 Objetivos del Problema .....                         | 13 |
| 1.3.1 Objetivo General.....                              | 13 |
| 1.3.2 Objetivos Específicos.....                         | 13 |
| 1.4 Justificación de la Investigación.....               | 13 |
| 1.5 Alcance de la Investigación.....                     | 16 |
| II MARCO TEÓRICO .....                                   | 17 |
| 2.1 Antecedentes .....                                   | 17 |
| 2.2 Bases Teóricas.....                                  | 22 |
| 2.2.1 Tecnologías de Información.....                    | 22 |
| 2.2.2 Sistemas de Información .....                      | 23 |
| 2.2.3 Estándar EDIFACT .....                             | 23 |
| 2.2.4 Base de Datos.....                                 | 25 |
| 2.2.5 Arquitectura Tecnológica Empresarial.....          | 26 |
| 2.2.6 Integración de Aplicaciones Empresariales.....     | 26 |
| 2.2.7 Middleware .....                                   | 27 |
| 2.2.8 Arquitectura de Software .....                     | 27 |
| 2.2.9 Arquitectura Orientada a Servicios .....           | 27 |
| 2.2.10 Patrón .....                                      | 29 |
| 2.2.11 Servicios Web .....                               | 29 |
| 2.2.12 Bus de Servicios Empresariales .....              | 31 |
| 2.3 Definición de Términos Básicos .....                 | 31 |
| III MARCO METODOLÓGICO                                   | 33 |
| 3.1 Tipo de Investigación .....                          | 33 |
| 3.2 Diseño de la Investigación .....                     | 33 |
| 3.3 Nivel de Investigación.....                          | 34 |
| 3.4 Los Instrumentos de Recolección de Información ..... | 34 |

|   |    |
|---|----|
| 3.5 Población y Muestra.....  | 34 |
| 3.6 Fases Metodológicas .....   | 35 |
| IV RESULTADOS.....  | 38 |
| 4.1 Modelado .....  | 38 |
| 4.1.1 Herramientas involucradas en la solución.....   | 38 |
| 4.1.2 Funcionalidades existentes en el sistema legado.....  | 39 |
| 4.1.3 Reglas de negocios.....   | 40 |
| 4.2 Diseño.....   | 40 |
| 4.2.1 Estudio de la arquitectura definida para el proyecto.....   | 40 |
| 4.2.2 Identificación de servicios candidatos.....   | 41 |
| 4.2.3 Generación de los contratos de los servicios candidatos.....  | 42 |
| 4.3 Desarrollo y pruebas .....  | 43 |
| 4.3.1 Repositorio de servicios.....   | 43 |
| 4.3.2 Especificación de servicios.....  | 44 |
| 4.3.3 Pruebas a menor escala .....  | 54 |
| 4.3.4 Documento arquitectónico del proyecto.....  | 54 |
| 4.4 Realización e Implementación.....   | 56 |
| 4.4.1 Desarrollo del prototipo para la Prueba de Concepto.....  | 56 |
| 4.4.2 Protocolo de intercambio de mensajes con el sistema legado.....   | 57 |
| 4.4.3 Desarrollo completo del modelo de integración para la interoperabilidad de los sistemas mediante patrones de integración empresariales..... | 57 |
| V CONCLUSIONES Y RECOMENDACIONES .....  | 59 |
| 5.1 Conclusiones .....  | 59 |
| 5.2 Recomendaciones.....  | 60 |
| REFERENCIAS.....  | 61 |

## ÍNDICE DE GRÁFICOS

### GRÁFICOS

|   |    |
|---|----|
| 1 Elementos involucrados en la arquitectura de la solución (ESB-MQ) .34     | 34 |
| 2 Extracto de ejemplo de un contrato de servicio (WSDL) .....36             | 36 |
| 3 Representación del repositorio de los servicios y operaciones .....37     | 37 |
| 4 Vista general del servicio web ICGInditex ..... 37                        | 37 |
| 5 Flujo principal de la operación Out_VentasTotalesTienda .....38           | 38 |
| 6 Flujo principal de la operación Out_VentasTotalesTiendaB64 .....39        | 39 |
| 7 Vista general del servicio web bB_Ambar_WS .....39                        | 39 |
| 8 Flujo principal de la operación Procesar_maestra_articulos .....40        | 40 |
| 9 Flujo principal de la operación Ingresar_articulosICG .....40             | 40 |
| 10 Flujo principal de la operación Historial .....41                        | 41 |
| 11 Flujo principal de la operación Evento .....41                           | 41 |
| 12 Flujo principal de la operación Crear_tablas_sistema .....42             | 42 |
| 13 Flujo principal de la operación Validar_reglas_negocios .....42          | 42 |
| 14 Vista general del servicio web SFTP .....43                              | 43 |
| 15 Flujo principal de la operación Cargar .....43                           | 43 |
| 16 Flujo principal de la operación Descargar .....44                        | 44 |
| 17 Vista general del servicio web FTP .....45                               | 45 |
| 18 Flujo principal de la operación Listar_archivos_Nuevos .....46           | 46 |
| 19 Flujo principal de la operación Listar_archivos .....46                  | 46 |
| 20 Flujo principal de la operación Evento_rollback .....47                  | 47 |
| 21 Flujo principal de la operación envío de reportes Inditex España .....48 | 48 |
| 22 Flujo principal de la ruta de mediación SFTP_channel_with_log .....48    | 48 |
| 23 Flujo principal de la ruta de mediación sftp_dead_letter_queue_log 49    | 49 |

## INTRODUCCIÓN

Los sistemas de información han recorrido durante los últimos 20 años un camino que lleva desde los primitivos sistemas monolíticos difíciles de administrar, pasando por sistemas cliente/servidor, hasta las arquitecturas basadas en servicios, en las que la funcionalidad de las aplicaciones se implementa en la forma de componentes reutilizables e invocables mediante interfaces estándar, que pueden combinarse para crear funciones progresivamente más complejas. Hoy en día toda la industria, tanto los proveedores de tecnología como los consultores y los usuarios, están de acuerdo en que este último enfoque permite mejorar la calidad de las aplicaciones y dar una mejor respuesta a las necesidades de los negocios. En base a esto, la empresa Textiles Zanzibar C.A. que componen las cadenas de tiendas Zara C.A, Bershka C.A y Pull&Bear C.A a nivel nacional, solicitó el requerimiento de la creación de un sistema que ayude a mejorar el proceso de salida de ventas a España, el cual, se realiza en gran medida de forma manual, pudiendo ser automatizado, y en donde además, se le otorgue al personal que labora en el área de tecnología, un medio para gestionar las tareas humanas que éste tenga. Para satisfacer esta necesidad, se planteó a dicha entidad, un sistema Middleware (lógica de intercambio de información entre aplicaciones) basado en la Arquitectura Orientada a Servicios, que ofreciera al personal que allí labora una interfaz amigable, y en donde además se automatizaran aquellos procesos que aún no lo están, centralizando en una sola herramienta la realización de todas las tareas correspondientes a la consulta de información en los sistemas legados. A continuación, se dará una breve introducción de los capítulos.

Capítulo I se presentará una descripción del problema, se darán a conocer los objetivos tanto el general como los específicos y la justificación de la investigación.

Capítulo II se explicará brevemente conceptos teóricos de suma importancia para entender un poco más el paradigma de desarrollo aquí encontraremos los antecedentes y las bases teóricas.

Capítulo III describir cada una de las fases por las que pasó el proyecto así como la metodología utilizada.

Capítulo IV se presentarán los resultados obtenidos en este trabajo de grado.

Finalmente, en el capítulo V se presentan las conclusiones y recomendaciones.

# **CAPÍTULO I**

## **EL PROBLEMA**

### **1.1 Planteamiento del Problema**

En la actualidad las empresas necesitan tener una gran flexibilidad para poder adaptarse fácilmente a lo que un entorno empresarial muy exigente les demanda. Las arquitecturas tecnológicas tradicionales como: “Cliente/Servidor”, “Definición Web”, “Programación por capas” entre otras no han podido dar una respuesta adecuada a esta necesidad debido a que en ellas las aplicaciones están pensadas generalmente con paradigmas verticales, pensadas para un propósito específico y limitado, de modo que su integración resulta muy compleja. Esto dificulta a su vez la rápida adaptación de los procesos para poder aprovechar nuevas oportunidades de negocio o responder más rápido a problemas internos. En la actualidad hay una forma relativamente nueva y poco explotada de concebir los sistemas de información, denominada Arquitectura Orientada a Servicios (SOA), que parece brindar la mejor respuesta de la que se dispone hasta ahora para dotar a las organizaciones de la flexibilidad requerida. En una SOA, la funcionalidad aplicativa se brinda a través de componentes denominados servicios, que presentan interfaces estándar bien definidas y que representan funciones de negocio. Los servicios se pueden combinar en lo que se denomina orquestación, permitiendo implementar de ese modo procesos de negocio sumamente ágiles y flexibles.

Un ejemplo de esto, El Centro Nacional de Tecnología de la Información (CNTI) en conjunto con los encargados de la planificación y financiamiento de los proyectos en el área de tecnología proponen la interoperabilidad de las instituciones del Estado apoyándose en la adopción de tecnologías informáticas abiertas libres y la aplicación de una SOA para así acelerar el Gobierno Electrónico, con el objetivo de establecer un acercamiento entre las instituciones estratégicas del estado y garantizar tanto la sustentabilidad como la escalabilidad de los sistemas involucrados.

La empresa Textiles Zanzibar C.A.

que componen las cadenas de tiendas Zara C.A, Bershka C.A y Pull&Bear C.A se han visto en la necesidad de reformular y adaptarse a una nueva forma de integrar los sistemas debido a las cantidades de sistemas involucrados que componen este grupo de empresas, aunque se han aplicado patrones de integración a través de archivos de texto delimitados o posicionales que son una solución relativamente rápida y puede aplicarse tanto a problemas básicos como complejos se requiere un constante muestreo de la estructura, generación de los archivos a integrar y dependiendo del volumen de la data, el proceso de verificación de los mismos puede corresponder a una tarea posterior de depuración, también se aplican base de datos intermedias las cuales ofrecen un punto en común entre sistemas, que a través del desencadenamiento de disparadores o aplicaciones externas que utilicen la data temporal ingresada del almacén de datos principal de un sistema, puedan alimentar uno o más sistemas externos, esta aproximación les permite exponer solo la data de interés entre sistemas, se aplica el traspaso de archivos mediante el protocolo FTP o SFTP para así transmitir información entre sistemas exteriores a la red local.

Tomando en cuenta la cantidad de sistemas involucrados y los distintos patrones de integración empresarial aplicados, aún tienen que lidiar con grandes desarrollos de integración para la adquisición de nuevos sistemas debido a que la dinámica de la empresa produce un volumen de data considerable, el cual sería una tarea casi imposible de procesar manualmente ya que eso pudiera generar incoherencia y un mal funcionamiento entre sistemas, el cual puede traer pérdidas a la empresa y problemas a nivel organizacional debido a que se emplean más recursos humanos a tareas que no se tenían contempladas pudiendo entorpecer los procesos internos de la empresa ya que no se puede procesar la data en un tiempo prudente para utilizar la información generada en los cuerpos internos de la organización.

Por lo tanto surge la propuesta de generalizar el proceso de integración entre sistemas mediante un modelo interoperable basado en software libre que les permita gestionar mejor la integración entre sistemas con el uso de un bus de servicios de empresarial (ESB) que permita gestionar colas de mensajes para garantizar la efectiva comunicación entre sistemas y a través del uso del ESB exponer los

sistemas involucrados en los procesos de negocios como servicios de interfaces de programación de aplicaciones (API) que son descritos mediante un lenguaje de marcas extensible (XML) que establece un método de comunicación generalizado entre servicios y definir la lógica de orquestación con el uso de rutas de mediación para definir la operatividad de los servicios que componen el ESB.

## **1.2 Formulación del Problema**

De acuerdo a lo planteado se formula la siguiente interrogante: ¿De qué manera optimizar el proceso de interoperabilidad entre sistemas?

## **1.3 Objetivos del Problema**

### **1.3.1 Objetivo General**

Desarrollar un software de interfaces que permita implementar un modelo de integración para la interoperabilidad de los sistemas mediante patrones de Integración Empresariales.

### **1.3.2 Objetivos Específicos**

- Diagnosticar la problemática existente en los sistemas empresariales debido a la falta de integración entre aplicaciones.
- Diseñar y estudiar la arquitectura del sistema.
- Implementar y acoplar individualmente los componentes: mensajería, orquestador y servicios requeridos que formaran parte del modelo de integración.
- Realizar prueba de concepto sobre los servicios correspondientes.
- Implantar y configurar el modelo de integración. Caso de estudio Empresa: Textiles Zanzibar C.A. (Zara, Bershka y Pull&Bear a nivel nacional)

## **1.4 Justificación de la Investigación**

Muchas empresas a nivel mundial se encuentran en procesos de fusión, adquisición e integración lo cual las fuerza a cambiar constantemente. Sin embargo la tecnología que las soporta sigue siendo rígida y en algunos casos muy costosas, esto limita el crecimiento que puedan tener en el futuro.

Un problema común que afrontan las organizaciones es el de integrar sus sistemas y procesos compartiendo datos e información en tiempo “real”. Lo que se ve en la realidad es la existencia de información incoherente o mal funcionamiento entre sistemas debido a la operatividad de sistemas aislados. Para poder tener información actualizada y sistemas integrados de forma flexible muchas empresas utilizan herramientas de código abierto, muchas de las cuales suelen estar a la vanguardia de la investigación a diferencia de otras tecnologías comerciales que no lo están debido a la falta de escalabilidad de sus arquitecturas. Las ventajas de adquirir un modelo de integración es que permite tener una forma unificada de integrar los sistemas mediante interfaces de comunicación entre aplicaciones llevando a una forma estandarizada de implementar integraciones entre sistemas, adicionalmente en la aplicación del modelo se utiliza el patrón de bus de datos empresarial (ESB) y la cola de mensajerías (MQ) el cual sigue los estándares de la norma ISO/IEC 9126 que hace referencia a la calidad del software. La adopción de una arquitectura basada en servicios, en las que la funcionalidad de las aplicaciones se implementa en la forma de componentes reutilizables e invocables mediante interfaces estándar, pudiendo combinarse para crear funciones progresivamente más complejas y así resolver incidencias más complejas adaptándose más fácil a los requerimientos sin afectar la operatividad.

Adicionalmente los beneficios que entrega la solución, pueden dividirse en dos niveles distintos: a nivel de empresa y a nivel de organización tecnológica. Desde el punto de vista corporativo:

Mejora la toma de decisiones: Al integrar la información dentro de un conjunto de aplicaciones dinámicas compuestas, los directivos disponen de más información, exacta y actualizada a menor tiempo, por lo que podrán reaccionar de manera ágil y rápida al surgir problemas o cambios.

Mejora la productividad de los empleados: Un acceso óptimo a los sistemas, información, y a la posibilidad de mejorar los procesos, permiten que las empresas puedan aumentar la productividad individual de los empleados. Esto porque al poder acceder a la información que necesiten, ayudará a que su productividad se multiplique en una gran cantidad de

escenarios nuevos.

Se

potencian las relaciones con clientes y proveedores: Las ventajas que ofrece SOA trascienden las fronteras y límites de la propia organización, los procesos de fusión y compra de empresas se vuelven más rentables debido a la sencillez de la integración de aplicaciones diferentes, y la integración con socios comerciales, y optimización de la cadena de suministro, bajo esta modalidad, son objetivos totalmente factibles. Con SOA se puede conseguir mejorar la capacidad de respuesta.

Desde el punto de vista de los departamentos de tecnologías de información (TI):

Aplicaciones más productivas y flexibles: La estrategia de orientación a servicios, permite a TI conseguir una mayor productividad de los recursos TI ya existentes (aplicaciones y sistemas instalados o los más antiguos), y obtener mayor valor de éstos para la organización.

Desarrollo de aplicaciones más rápido y económico: Gracias a la integración flexible de todos los datos, esfuerzos de ingeniería incrementales y la creación de un repositorio de servicios utilizables que se pueden combinar en servicios de mayor nivel, se logra una aceleración en el desarrollo de proyectos de TI y una disminución en los costos del desarrollo de soluciones y de los ciclos de prueba, pues se eliminan redundancias y se consigue su puesta en producción en menos tiempo.

Aplicaciones más seguras y manejables: proporciona una infraestructura y documentación común, para desarrollar servicios seguros y gestionables, además de facilitar la posibilidad de añadir nuevos servicios y funcionalidades para gestionar los procesos de negocios críticos. Puesto que se accede a los servicios y no a las aplicaciones, en consecuencia optimiza las inversiones realizadas en TI potenciando la capacidad de introducir nuevas capacidades y mejoras.

Minimización del riesgo de tiempo de inactividad o pérdidas de datos: Esto debido a que ofrece rendimiento, escalabilidad, seguridad y alta disponibilidad sin precedentes.

Mejora de la capacidad para innovar y diferenciarse: Mediante datos integrales, precisos y oportunos para desarrollar la nueva funcionalidad de forma rápida.

## **1.5 Alcance de la Investigación**

Con la aplicación de un software de interfaces que permitirá implementar un modelo de integración para la interoperabilidad de los sistemas mediante patrones de integración empresariales. Se pretende consolidar una base de infraestructura de software para la empresa Textiles Zanibar C.A. que permita a la empresa integrar sus sistemas de una forma más transparente y unificada, con la intención de automatizar procesos y dar a la empresa la posibilidad de escalabilidad a través de la reutilización de componentes, asimismo la posibilidad de adaptarse más fácilmente a los cambios que pueden sufrir los sistemas debido a la lógica de negocios. La metodología a utilizar en este proyecto será la metodología de Arquitectura de Modelado Orientada a Servicios (SOMA) es un método de desarrollo e integración (Ciclo de vida del proceso) basada en principios de ingeniería de software para producir soluciones mediante la Arquitectura Orientada a Servicio (SOA). Éste método incluye orientación prescriptiva aplicable durante las fases e iteraciones requeridas en el ciclo de vida de un proceso de desarrollo de software. Las principales funciones que tendrá la aplicación, son las siguientes:

- Enviar a España el reporte de ventas por tienda de las cadenas de tiendas que componen la empresa Textiles Zanibar C.A a nivel nacional (ZARA, BERSHKA, PULL&BEAR).
- Llevar un control de auditoria de los reportes enviados a España
- Gestionar el volumen de procesos a través de una cola mediante el intermediario de mensajes y el uso de rutas de mediación.
- Desarrollar una interfaz abstracta de comunicación que permita la comunicación con el servidor SFTP para el envío de reportes.
- La aplicación contará con una versión de consola que será el proceso automatizado y una versión de escritorio en caso que se necesite ejecutar el aplicativo de forma manual.

Por lo tanto las herramientas a utilizar para el desarrollo de dicha aplicación serían Java 7 y C# como lenguajes de programación, y bases de datos de textos debido a su capacidad de aprovisionamiento.

## CAPÍTULO II

### MARCO TEÓRICO

#### 2.1 Antecedentes

En relación con el propósito perseguido en esta investigación, como es la interoperabilidad de los sistemas mediante patrones de integración empresariales, se analizan los siguientes trabajos:

Son muy diversas y variadas las investigaciones realizadas donde se observan serios intentos y propuestas arquitecturales para resolver el problema interoperabilidad de los sistemas; así como metodologías que ponen en práctica sin traumas, la implementación del paradigma SOA junto con métodos de integración apropiados. En esta sección se hará mención de aquellos trabajos que han antecedido a la presente investigación, que de alguna manera u otra han ayudado a la misma y han servido de base para la solución del problema planteado.

Pérez (2008), propuso en su tesis doctoral titulada “**Enfoque orientado a servicios para la reutilización de procesos y herramientas de modelización presentada en la Universidad de Grenoble. Grenoble, Francia**”, en el cual se plantea un enfoque basado en conceptos teóricos, para rehusar procesos existentes, modelos y herramientas, con el fin de encontrar una respuesta a las metas de los gerentes de proyectos de desarrollo de software. Su propuesta estuvo basada en tres niveles abstractos o capas: nivel intencional, que trata con las metas de modelado. El nivel organizacional, el cual ofrece servicios a los grupos de proyectos en términos de procesos de desarrollo simplificados, y nivel operacional, la cual ofrece servicios a los diseñadores de modelos.

Por la manera en la que está estructurada la investigación se puede decir que es de apoyo a la presentada, debido a que en ambas muestran la necesidad de la reutilización de procesos con la finalidad de ver estos procesos como servicios y así poder dar una solución más acertada a lo que concierne al desarrollo de software por lo tanto esta investigación es de gran aporte debido a que se seguirán las

recomendaciones teóricas desarrolladas en la investigación con el fin de poder obtener los beneficios de la reutilización de procesos.

Papazoglou y Van Den Heuvel (2005), en su escrito titulado **“Services oriented architecture: approaches, technologies and research issues publicada en un escrito en Very Large Data Bases Journal (VLDB)”**, donde expusieron las tecnologías y enfoques que unifican los principios y conceptos de una arquitectura orientada a servicios.

Este escrito constituyó un aporte importante pues ponen de manifiesto los problemas que tienen las empresas modernas para integrarse tecnológicamente con sus clientes o asociados de negocio, lo que trae consigo la necesidad de integrar también sus procesos de sus negocios, ya sean internos o externos; esto por supuesto requiere de una implantación que sea rápida y que los servicios de los negocios se ensamblen con quien lo necesite de una manera transparente.

Adicionalmente el documento se centra en el bus de servicios empresarial y describe una serie de funciones que están diseñadas para ofrecer un “backbone” de fácil uso para arquitectura orientada a servicios, basadas en estándares, lo que amplía la funcionalidad del middleware, todo mediante la conexión de componentes y sistemas heterogéneos y además ofrece servicios de integración. Por último, el documento propone un enfoque para ampliar la SOA convencional, para así satisfacer los requerimientos esenciales de un ESB, que incluyen capacidades, tales como: orquestación de servicios, enrutamiento "inteligente", aprovisionamiento, integridad y seguridad del mensaje, así como la gestión de servicios.

Como producto final, propusieron una arquitectura SOA, junto con un ESB como middleware, esta propuesta es importante para el estudio ya que da una visión de la arquitectura principal del modelo de integración para la interoperabilidad de los sistemas mediante patrones de integración empresariales a construir. A pesar que dicha publicación fue realizada en el 2005, aún mantiene vigencia ya que los traumas de diseño de la arquitectura orientada a servicios, hoy en día continúan siendo un problema para establecer esta filosofía en los sistemas, por ello siguiendo su propuesta de una arquitectura orientada a servicios extendida con un middleware

ESB, aunado a su experiencia, garantizan un diseño arquitectural bastante robusto y escalable

Papazoglou y Van Den Heuvel (2006), como expertos en el área de arquitectura de software; realizaron otra publicación titulada “**Service-oriented design and development methodology, esta vez publicada en el International Journal of Web Engineering and Technology (IJWET)**”, el cual tiene como objetivo, examinar una metodología de desarrollo de servicios desde el punto de vista tanto de los productores como de los consumidores de servicios, además hacen una revisión de la gama de elementos de esta metodología. En el mismo ponen de manifiesto la dificultad que presentan las empresas para implementar SOA. Por lo que en el artículo ofrecen una visión de los métodos y las técnicas utilizadas en el diseño y desarrollo SOA. Para darle solución a tales dificultades, luego de revisar distintos métodos y técnicas, donde se describe una metodología experimental para el diseño y desarrollo de una SOA; según los autores esta metodología refleja un intento en definir los principios del diseño y desarrollo que son aplicados tanto en los servicios web como en los procesos de negocio. La metodología toma en cuenta un conjunto de los modelos de desarrollo y varios escenarios para la realización de servicios.

El resultado de este artículo pone de manifiesto las actividades en torno a la prestación de servicios, la implementación, ejecución y seguimiento. Según los autores estas actividades serán cada vez más importante en el mundo de los servicios, ya que contribuyen al concepto de capacidad de servicio de adaptación, donde los servicios y procesos continuamente puede transformarse ellos mismos para responder a las demandas del medio ambiente y los cambios sin comprometer la eficiencia operativa y financiera. Esta investigación es de gran aporte debido a que se utilizara la arquitectura orientada a servicios para el desarrollo de este proyecto.

La arquitectura orientada a servicios es un concepto relativamente nuevo. Las primeras definiciones de SOA datan del año 1996 y fue alcanzando su impacto de manera progresiva desde entonces. El concepto de SOA fue adquiriendo mayor importancia a medida que las necesidades de flexibilidad, facilidad de

mantenimiento, integración, reutilización de lógica y datos, y adaptación de la plataforma de tecnologías de información a los procesos de negocio se hacían más evidentes en las empresas. Para entender mejor el concepto, debe primero comprenderse el fundamento base de esta arquitectura: los servicios. En el contexto empresarial, el concepto de servicios es utilizado muy frecuentemente desde hace varios años, ya que permite una mejor flexibilidad y eficiencia en su operación. Siguiendo con la idea anterior las tecnologías de información y las empresas utilizan servicios en casi todos los procesos.

Estos servicios son proporcionadas por alguna entidad (proveedor del servicio) interna o externa a la organización, la cual estará encargada de entregar al solicitante (consumidor del servicio) el insumo necesario para continuar la operación. De esta manera, las empresas utilizan servicios de banca, facturación, contabilidad, entrega de paquetes, de limpieza, de información de crédito y de seguridad, entre muchos otros procesos.

Zulma Díaz (2006), su trabajo de postgrado titulado “**Metodo para el desarrollo de servicios web, presentado en la Universidad de Los Andes Facultad de Ingeniería Postgrado en Computación Mérida -Venezuela**”, donde representa en su investigación una oportunidad de alcanzar la interoperabilidad de los sistemas actuales. Donde a través de servicios web se permita continuar utilizando los sistemas legados de las organizaciones y que estos, a su vez, se comuniquen con otras organizaciones, utilizando Internet como canal de comunicación. Desde el punto de vista de la Ingeniería del Software

Se determinó un aporte importante el cual ofrece a las organizaciones una posibilidad de integrar sistemas que no estuvieron pensados para tal fin mediante la implementación de servicios web y así no tener que descartar estos sistemas legados que pueden o no, ser un pilar base para la organización siendo irremplazables para las mismas, cabe resaltar que inicialmente, SOA fue influenciado por el concepto de Servicios Web (Web Services), al ser estos las primeras implementaciones de servicios en el ámbito del software. Sin embargo, se debe tener claro que una arquitectura SOA no implica la implementación de servicios web, y la implementación de servicios web no necesariamente implica una arquitectura SOA.

Este modelo define una serie de patrones para los elementos de la arquitectura, entre ellos: la interacción de los componentes, el diseño y exposición de los servicios. El modelo de interacción de SOA específica, en principio, la comunicación entre tres componentes básicos: el proveedor del servicio, el consumidor del servicio y el Bus de servicios empresarial.

Arsanjani, (2004), realizó una publicación titulada, “**Service-oriented modeling and architecture, presentado en la conferencia IBM, Software Group**”, en el cual explica la relevancia del Bus de Servicios Empresarial (ESB, por sus siglas en inglés). Este elemento, en general, provee funcionalidades de conectividad, mediación y registro de servicios, sin embargo, las funcionalidades incluidas en el ESB dependerán de cada implementación. Por medio de sus capacidades de mediación, el ESB disminuye la complejidad de las conexiones. Además, facilita la interoperabilidad entre los distintos elementos de la arquitectura, aunque estos pueden ser totalmente heterogéneos en su tecnología de implementación.

A través de esta publicación se pudo determinar, que el uso de un ESB reduce las dependencias entre los componentes, lo cual permite desacoplar las implementaciones de los servicios de su utilización y, de esta manera, facilitar el reemplazo futuro de componentes y la adaptación de nuevos servicios.

Daily Echeverría, Hernán Astudillo, Rodrigo Estrada (2010), su artículo titulado “**ESB-QM: Modelo de Calidad para productos ESBs presentado en la Universidad Técnica Federico Santa María, Av. España 1680 Valparaíso, Chile**”, donde su investigación remarca la relevancia que han adquirido los ESB como soluciones de integración que surgen en respuesta a las necesidades de las SOA especialmente para satisfacer problemas de integración utilizando las ventajas de la misma y manteniendo sus principios. Añadiendo la especificación y evaluación de la calidad de los productos de software como un factor clave para garantizar la calidad adecuada de los mismos.

Los resultados de esta investigación son de suma importancia pues proporciona una herramienta a los arquitectos de software, que les permite tratar a los EBS como productos estándares y así poder gestionar bajo modelo

estándar ESB-MQ poder contemplar la calidad y evaluación del software a implementar para garantizar su buen funcionamiento.

## **2.2 Bases Teóricas**

Representan la descripción de los elementos teóricos planteados por diferentes autores, que permiten al investigador fomentar el proceso de conocimientos, así como también, fundamentar los planteamientos señalados y así sea más fácil su entendimiento. Por tal motivo, a continuación se presentan las siguientes literaturas relacionadas con la problemática abordada. En este sentido, Sabino, C. (2010), opina lo siguiente sobre las bases teóricas:

Son aquellas teorías de distintas áreas de conocimiento que por estar relacionados o vinculados con la investigación que se realiza, se toman en consideración para obtener un cuadro o marco de información y sustentación que permitan y ayuden a la interpretación de los hechos o fenómenos objeto de estudio. (p.5)

Por consiguiente con las bases teóricas se busca la información principal acerca de aspectos específicos en la descripción del contenido, es decir, se recoge el material de mayor relevancia para el desarrollo de la investigación y su conceptualización.

### **2.2.1 Tecnologías de Información**

Las organizaciones para el fortalecimiento de sus capacidades de innovación, hacen uso intensivo de las tecnologías de información, a fin de que estas le ayuden a enfrentar la competitividad presente en el mundo globalizado y lograr tener más éxito en el mercado.

La Asociación de Tecnología de Información de América (ITAA), citada por Hernández (2008), afirma que la tecnología de información es “el estudio, diseño, desarrollo, implementación, soporte o dirección de los sistemas de información computarizados, en particular de software de aplicación y hardware de computadoras”. Así mismo, Bologna y Walsh, (1997), señalan que la tecnología de la información se entiende como aquellas herramientas y métodos empleados para recabar, retener, manipular o distribuir información, pues la tecnología de la

información se encuentra generalmente asociada con las computadoras y las tecnologías afines aplicadas a la toma de decisiones”.

Así, las tecnologías de información, es el conjunto de herramientas y métodos que permiten gestionar la información en pro del desarrollo de las organizaciones mediante la mezcla de la computación y las comunicaciones. Al respecto, Bologna y Walsh (1997), describen las aplicaciones de la tecnología de información, las cuales son: el almacenamiento de datos, información para la toma de decisiones y control, sistema de toma de decisiones organizacionales, planeación de recursos empresariales y administración del conocimiento.

### **2.2.2 Sistemas de Información**

Laudon y Laudon (1996), definen los sistemas de información (SI) como “un conjunto de componentes interrelacionados que reúne (u obtiene), procesa, almacena y distribuye información para apoyar la toma de decisiones y el control en una organización”, y sus actividades están determinadas por: entrada, procesamiento y salida de datos.

### **2.2.3 Estándar EDIFACT**

UNACE (1996). United Nations Economic Commission for Europe, En un principio se estableció UN / EDIFACT (las normas de las Naciones Unidas para el intercambio electrónico de datos para la administración, el comercio y el transporte) comprenden un conjunto de normas acordadas internacionalmente, directorios y directrices para el intercambio electrónico de datos estructurados, entre los sistemas de información computarizados independientes.

Donde la facilitación del comercio se ocupa de los requisitos y procedimientos relacionados con el flujo de información necesaria para el movimiento internacional de mercancías. Tradicionalmente, estos requisitos se han sustanciado en forma de documentos y procedimientos llevados a cabo a través de la manipulación de tales documentos en papel; por lo tanto, era natural que los esfuerzos iniciales de facilitación deban centrarse en la simplificación y normalización de los documentos de comercio exterior.

Durante décadas los estándares EDI fueron el formato dominante en el intercambio de datos para comercio electrónico y ofrecían a las organizaciones un método rápido y preciso de transmitir datos de transacciones. EDI antecede a tecnologías de comercio integradas muy populares como sistemas ERP, sistemas CRM, formatos de BDs y otras tecnologías relacionadas con cadenas de suministro. Por ello, la asignación y transformación de datos supone un componente importante de cualquier proyecto de implementación de EDI.

SAP es un sistema que implementa EDI y desde sus inicios, es un programa para aplicaciones de negocios. "SAP AG" es la empresa multinacional alemana creadora de lo que hoy todo el mundo conoce como SAP. Ellos son los responsables de todo lo que hoy el mundo realiza con este potente programa, y de todas las metodologías y "buenas prácticas" que han ido desarrollando y el mundo ha ido aplicando a lo largo de los años (desde 1970 hasta nuestros días).

SAP es un ERP, porque partiendo de la definición de ERP (Enterprise Resource Planning) que en castellano sería Sistema de Planificación de Recursos Empresariales, siendo un sistema modular que combina muchas áreas de la organización entre sí formando así un todo integrado que posibilita la comunicación e interacción de los datos, procesando así grandes cantidades de datos y obteniendo información útil para la toma de decisiones. Ofreciendo soporte a la gestión de importación para operaciones de comercio exterior, independientemente del país, y contiene todos los datos y estructuras necesarias para ello. Sus estructuras de datos cumplen la norma UN-EDIFACT, que permiten un intercambio de datos electrónico basado en el intercambio de archivos de texto delimitados o posicionales siguiendo los estándares establecidos en una operación EDI por sus siglas (Intercambio electrónico de datos). Sin embargo los avances tecnológicos, han hecho que surjan nuevas alternativas de manipulación de la información y métodos de transmisión factibles como lo son las bases de datos de intercambio, ya que la integración de archivos de texto tiene que lidiar con la generación y procesamiento de archivos de gran peso debido al volumen de data a transmitir lo cual puede conllevar a que exista una mal formación en la estructura de los archivos generados o que la data que existe en ellos este truncada por no contar con los

recursos necesarios para escribir en el sistema de archivos el volumen de datos generados.

#### **2.2.4 Base de Datos**

Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.

Según Damián Pérez Valdés, A 2007 “Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos. Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que queramos guardar en la tabla, cada fila de la tabla conforma un registro.”

Las bases de datos son comúnmente utilizadas en el tema de integración de servicios como lo que se conoce como base de datos intermedias que brindan un punto común entre uno o más sistemas en el que se almacenan datos que se quieren compartir entre ellos brindando la seguridad que ofrecen las bases de datos con usuarios basado en roles y permitiendo aplicar funciones o procedimientos almacenados al momento de manipular los datos o exponiendo solo los datos que aplicaciones o sistemas externos requieran para su buen funcionamiento.

Otro aspecto fundamental de las bases de datos es la sincronización de los datos que se refiere al proceso de propagación de los cambios en los datos y el esquema entre el publicador y los suscriptores después de haber aplicado la instantánea inicial en el suscriptor. La sincronización puede producirse de forma continua, lo que es típico de la replicación transaccional. A petición, para lo que es la replicación de mezcla o según una programación, que es la replicación de instantáneas.

ICG Software es un conjunto de aplicaciones que se apoya en el uso de multiple bases de datos y técnicas de replications conocido como un grupo multinacional español que fabrica soluciones informáticas para pequeñas, medianas

y grandes empresas de los sectores de la hostelería, hotel y retail, desde el software Punto de Venta hasta la gestión centralizada de cadenas y franquicias. Donde utilizan la arquitectura cliente servidor para la implementación de sus productos en la que mediante procesos de sincronización permiten tener un almacén de datos central en el cual reportan todas las operaciones, realizando procesos de sincronización entre bases de datos permitiéndole incluso trabajar de forma offline y una vez que se restablezca la comunicación entre los sistemas sincronizar los datos entre ellos sin tener que parar la operatividad de la empresa.

### **2.2.5 Arquitectura Tecnológica Empresarial**

García (2007), Define arquitectura tecnológica empresarial como el conjunto de componentes, servicios y procedimientos que dirigen y soportan el desarrollo y funcionamiento de una solución de negocio para una organización, garantizando la calidad, completitud y operatividad del resultado. Desde esta perspectiva, una arquitectura debería contemplar:

- Los componentes software orientados a satisfacer requerimientos de negocio o tecnológicos (arquitectura de ejecución).
- Las herramientas, estándares y metodologías para el óptimo desarrollo de software (arquitectura de desarrollo).
- La infraestructura técnica (hardware y software) para soportar el proceso de desarrollo y operación del sistema (arquitectura técnica).
- Los mecanismos, técnicas y componentes que garanticen la operación, nivel de servicio, mantenimiento y seguridad de la solución software (arquitectura de operación).

### **2.2.6 Integración de Aplicaciones Empresariales**

Según Frantz (2008), la integración de aplicaciones empresariales EIA (Enter prise Application Integration) son soluciones de integración cuyo objetivo es mantener sincronizadas un conjunto de aplicaciones, aportar nuevas funcionalidades y que se ejecutan en el lado del servidor dentro de la misma empresa, con el objetivo de mantener las aplicaciones funcionando en sincronía y de forma exógena poder integrar sus funcionalidades. Este tipo de solución suele

conectar dos o más aplicaciones por medio de un flujo exógeno de datos y/o comandos, capaz de integrar funcionalidades de las aplicaciones involucradas sin que las aplicaciones conozcan la solución. Es decir, una solución EAI aporta de forma exógena funcionalidad de una aplicación a la otra, o incluso, nuevas funcionalidades, además de mantener las aplicaciones independientes una de las otras y coordinarlas.

### **2.2.7 Middleware**

Es un módulo intermedio que actúa como conductor entre sistemas permitiendo a cualquier usuario de sistemas comunicarse con varias fuentes de información u otros sistemas que se encuentran conectadas por una red.

Para Bernstein (1996), “middleware es un software que ayuda a una aplicación interactuar o comunicarse con otras aplicaciones, redes, hardware y /o sistemas operativos. Este software ayuda a los programadores de librarlas de ellos complejas conexiones necesarias en un sistema distribuido. Ya que, proporciona herramientas para mejorar la calidad, la seguridad, el paso de mensajes, servicios de directorio, archivo servicios, base de datos, entre otros, que pueden ser invisibles para el usuario.”

### **2.2.8 Arquitectura de Software**

Presman (2002), hace referencia que "la Arquitectura de Software de un sistema de programa o computación es la estructura de las estructuras del sistema, la cual comprende los componentes del software, las propiedades de esos componentes visibles externamente, y las relaciones entre ellos".

### **2.2.9 Arquitectura Orientada a Servicios**

Carrascoso y Chaviano (2008), “La Arquitectura Orientada a Servicios es un concepto de arquitectura de software que define la utilización de servicios como construcciones básicas para el desarrollo de aplicaciones. Es una arquitectura de una aplicación donde las funcionalidades se definen como servicios independientes, con interfaces accesibles, bien definidas, que pueden ser llamadas en secuencias dadas para formar procesos de negocios”.

Reynoso, 2006. "SOA ha surgido como la mejor manera de afrontar el desafío de hacer más con menos recursos. Promete hacer la reutilización y la integración mucho más fáciles, ayudando a reducir el tiempo de desarrollo y aumentando la agilidad organizacional. No sorprendentemente, el 80% de las organizaciones de IT están implementando aplicaciones usando SOA con servicio web subyacentes. SOA proporciona mayor flexibilidad para afrontar los cambios tanto en el ambiente de negocios como en la infraestructura tecnológica"

Carrascoso y Chaviano (2008), mencionan que los principales componentes de una arquitectura orientada a servicios son:

**Servicios**, un servicio de negocio es un componente reutilizable de software, con significado funcional completo, y que está compuesto por:

- Contrato: especificación de la finalidad, funcionalidad, forma de uso y restricciones del servicio.
- Interfaz: mecanismo de exposición del servicio a los usuarios.
- Implementación: debe contener la lógica o el acceso a datos.

#### **Tipos de servicios.**

- Servicios básicos: pueden estar centrados en datos o en lógica y encapsulan funcionalidades como cálculos complejos, acceso a datos y reglas complejas de negocio.
- Servicios intermediarios: servicios adaptadores, bases de datos, entre otros. Suelen ser servicios sin estado.
- Servicios de proceso: servicios de negocio que encapsulan la lógica de proceso. Suelen conservar estado y pueden residir en herramientas BPM.
- Servicios públicos: servicios accesibles por terceros (fuera de la organización).

**Repositorio de servicios**, proporciona facilidades para descubrir servicios y adquirir la información necesaria para su uso, en particular fuera del alcance temporal y funcional del proyecto en el que se crearon. Además de la propia información de contrato, los repositorios pueden proporcionar información acerca de:

- Localización

- Personas de contacto.
- Restricciones Técnicas

**Bus de servicios**, el bus de servicios es el elemento de las arquitecturas SOA que conecta los servicios con sus consumidores y que proporciona:

- Conectividad: el propósito principal de un bus de servicios es interconectar a los participantes de una arquitectura SOA.
- Soporte a la heterogeneidad de tecnologías: debe ser capaz de conectar a participantes basados en distintos lenguajes de programación, sistemas operativos, entornos de ejecución y protocolos de comunicación.
- Soporte a la heterogeneidad de paradigmas de comunicación: debe ser capaz de mantener distintos modos de comunicación (por ejemplo comunicaciones síncronas y asíncronas).

**Consumidores de servicios**, definimos consumidores de servicios como aquellos elementos de una arquitectura SOA que:

- Pueden descubrir servicios a través de un repositorio.
- Realizan llamadas a los mismos de acuerdo al contrato y a través del interfaz definido a tal efecto.

### 2.2.10 Patrón

Reynoso (2004), propone que "un sistema bien estructurado está lleno de patrones", se podría ver a un patrón como una solución a un problema determinado, una codificación de algo específico basado en un conocimiento específico acumulado por la experiencia.

Según Fowler (2002), "cada patrón describe un problema que ocurre una y otra vez en nuestro ambiente, y luego describe el núcleo de la solución a ese problema, de tal manera que puedes usar esa solución un millón de veces más, sin hacer jamás la misma cosa dos veces".

### 2.1.11 Servicios Web

Las Arquitecturas Orientadas a Servicios, en contraste con las Arquitecturas Orientadas a Objetos, están formadas por servicios de aplicación altamente interoperables y débilmente acoplados. Donde estos servicios se pueden ver como

la evolución en complejidad de un componente distribuido, diferenciándose en que son Alves y otros (2006):

- Mucho menos acoplados con sus aplicaciones cliente que los componentes.
- Menor granularidad que los componentes.
- Son controlados y administrados de manera independiente.
- Expone su funcionalidad a través de protocolos abiertos e independientes de plataforma. Incluso arriesgando el rendimiento y consumo de recursos.
- Son transparentes de su localización en la red, de esta manera garantizan escalabilidad y tolerancia.

Los servicios web comunican aplicaciones, lo cual permite que se comparta información independientemente de cómo se hayan creado, cuál sea el sistema operativo o la plataforma en que se ejecutan y cuáles sean los dispositivos utilizados para obtener acceso a ellas. La comunicación se caracteriza por el intercambio de mensajes a través de lenguaje de etiquetado extensible XML y por ser independientes del protocolo de comunicación. Para lograr esta independencia, el mensaje XML se envuelve de manera apropiada para cada protocolo gracias a la creación del protocolo de transporte SOAP (Simple Object Access Protocol: Protocolo de Acceso a Objetos Simples).

El Lenguaje de Descripción de los Servicios Web (WSDL: Web Services Description Language), expresado en XML, describe cómo acceder al servicio, de qué funciones dispone, qué argumentos necesita y qué devuelve cada uno.

La otra tecnología fundamental implicada es la de Descripción Universal, Descubrimiento e Integración (UDDI: Universal Description, Discovery, and Integration). UDDI es un directorio de servicios web donde se puedan publicar los servicios ofrecidos, dar características del tipo de servicio, y realizar búsquedas.

En resumen, SOAP define un protocolo XML para la interoperabilidad básica entre servicios, WSDL introduce un lenguaje común para describir servicios y UDDI provee la infraestructura requerida para publicar y descubrir servicios programáticamente. Juntas, estas especificaciones permiten a las aplicaciones interactuar siguiendo un modelo débilmente acoplado e independiente de la plataforma subyacente.

### **2.1.12 Bus de Servicios Empresariales**

TIBCO Software Inc (2008), un bus de servicios empresariales (ESB) es 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. 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 primer paso clave para el establecimiento de una SOA empresarial. Los ESB constituyen los cimientos de una SOA y pueden complementarse con capacidades de productividad adicionales, como la orquestación de servicios y los registros. Este documento trata sobre los requisitos para el Establecimiento de un ESB capaz de satisfacer las necesidades de una SOA lista para operar en un marco empresarial.

En informática un bus de servicios de empresa (ESB) consiste en un combinado de arquitectura de software que proporciona servicios fundamentales para arquitecturas complejas a través de un sistema de mensajes (el bus) basado en las normas y que responde a eventos. Los desarrolladores normalmente implementan un ESB utilizando tecnologías de productos de infraestructura de middleware que se basan en normas reconocidas.

### **2.3 Definición de Términos Básicos**

**Backbone (columna vertebral):** Mecanismo de conectividad primario en un sistema distribuido. Todos los sistemas que tengan conexión al backbone (columna vertebral) pueden interconectarse entre sí, aunque también puedan hacerlo directamente o mediante redes alternativas.

**Bus de servicio empresarial (ESB):** En informática un bus de servicios de empresa consiste en un combinado de arquitectura de software que proporciona servicios fundamentales para arquitecturas complejas a través de un sistema de mensajes (el bus) basado en las normas y que responde a eventos. Los

desarrolladores normalmente implementan un ESB utilizando tecnologías de productos de infraestructura de middleware que se basan en normas reconocidas.

**Gestión de proceso de negocios (BPM):** En tecnología y gestión empresarial, BPM por sus siglas en inglés (Business Process Management). Siendo BPM una decisión de transformación organizacional, vista como un mecanismo o herramienta de gestión empresarial que permite alinear las diferentes dimensiones (organización, procesos, conocimiento y estrategia), trasladando a la organización de una gestión funcional a una gestión de procesos de negocio exitosa.

**Middleware o lógica de intercambio de información entre aplicaciones:** es un software que asiste a una aplicación para interactuar o comunicarse con otras aplicaciones, o paquetes de programas, redes, hardware y/o sistemas operativos. Éste simplifica el trabajo de los programadores en la compleja tarea de generar las conexiones y sincronizaciones que son necesarias en los sistemas distribuidos.

**Offline o fuera de línea:** Referente a una operación o a una unidad funcional, cuando no está bajo control directo del sistema al cual está asociada. Las unidades fuera de línea no están disponibles para su uso inmediato a solicitud del sistema, aunque pueden ser operadas independientemente (como el navegador de un computador que no se encuentre conectado a Internet).

**Software:** Se conoce como *software al equipo lógico o soporte lógico* de un sistema informático, que comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos que son llamados hardware.

## **CAPÍTULO III**

### **MARCO METODOLÓGICO**

En este capítulo se presenta la metodología que permitió desarrollar el presente Trabajo. Se muestran aspectos como el tipo de investigación, las técnicas y procedimientos que fueron utilizados para llevar a cabo dicha investigación.

#### **3.1 Tipo de Investigación**

Tomando en cuenta la situación de la problemática planteada, se considera que dicha investigación se encuentra dentro de la modalidad de proyecto especial, ya que se orienta en una creación de software, Las Normas de Trabajo de Grado de la Universidad José Antonio Páez (UJAP, julio 2007), define proyecto especiales como:

Proyectos especiales consistirá en las creaciones tangibles, susceptibles de ser utilizadas como soluciones a problemas demostrados, o que respondan a necesidades e intereses de tipo cultural. Se incluyen en esta categoría los trabajos de elaboración de libros de textos y de materiales de apoyo educativo, el desarrollo de software y hardware, prototipos y de productos tecnológicos en general.

#### **3.2 Diseño de la Investigación**

Después de describir de forma detallada y precisa el problema, se determinó el diseño de la investigación, dicho diseño está enmarcado dentro de la investigación de campo, tal como se refiere en Las Normas de Trabajo de Grado de la Universidad José Antonio Páez (UJAP, julio 2007), donde señala:

Se entenderá por investigación de campo, el análisis sistemático de problemas en la realidad, con el propósito bien sea de describirlos, interpretarlos, entender su naturaleza y factores constituyentes, explicar sus causas y efectos, o predecir su ocurrencia, haciendo uso de métodos o enfoques de investigación conocidos o en desarrollo. Los datos serán recogidos en forma directa de la realidad; en este sentido se trata de investigaciones a partir de datos originales o primarios. Sin embargo, se aceptarán también estudios sobre datos censales o muestrales no

recogidos por el estudiante, siempre y cuando se utilicen los registros originales con los datos no agregados; o cuando se trate de estudios que impliquen la construcción o uso de series históricas y, en general, la recolección y organización de datos publicados para su análisis mediante procedimientos estadísticos, modelos matemáticos, econométricos o de otro tipo.

### **3.3 Nivel de Investigación**

La siguiente investigación es de nivel descriptivo, ya que los datos obtenidos, son analizados y descrito de forma detallada, según la situación planteada por dicha empresa, estableciendo características de los procesos que son estudiados para poder comprenderlos Según Hernández R., Fernández C. y Baptista P. (1997) “Las investigaciones descriptivas buscan especificar las propiedades importantes de personas, grupos, comunidades o cualquier otro fenómeno que sea sometido a análisis. Miden o evalúan diversos aspectos, dimensiones o componentes del fenómeno o fenómenos a investigar”. En el caso de la presente investigación, se precisa un estudio descriptivo debido a que los investigadores destacan las características de los fenómenos observados a fin de comprenderlos y poder plantear soluciones.

### **3.4 Los Instrumentos de Recolección de Información**

Para el desarrollo de esta investigación fue necesario utilizar herramientas que permitieron recolectar el mayor número de información necesaria, con el fin de obtener un conocimiento más amplio de la realidad de la problemática. Por naturaleza del estudio se requirió la recopilación documental, que se trata del acopio de los antecedentes relacionados con la investigación. Para tal fin se consultaron documentos escritos, formales e informales.

### **3.5 Población y Muestra**

Una población según Tamayo y Tamayo M. (2003), es definida como “La totalidad del fenómeno a estudiar en donde las unidades de población poseen una característica común, la cual se estudia y da origen a los datos de la investigación”;

por su parte, Hernández et al (2010) define la muestra como “La muestra es, en esencia, un subgrupo de la población”.

Población: La empresa Textiles Zanzibar C.A. que componen las cadenas de tiendas Zara C.A, Bershka C.A y Pull&Bear C.A, Consultorias y sistemas ICG y bestBOON C.A. Veintisiete (27) tiendas a nivel nacional.

### **3.6 Fases Metodológicas**

Para alcanzar el éxito en el proyecto, es fundamental seguir una metodología que garantice el correcto desenvolvimiento del proyecto en todas sus fases. Para el desarrollo de este proyecto se utilizó la metodología de Arquitectura de Modelado Orientada a Servicios. Es un método de desarrollo e integración (Ciclo de vida del proceso) basada en principios de ingeniería de software para producir soluciones mediante la Arquitectura Orientada a Servicio (SOA).

#### **Fase I: Fase de Diagnóstico**

Durante esta fase se realizó un análisis de la funcionalidad del sistema de operaciones de negocios la empresa Textiles Zanibar C.A, así como un estudio a través de tutoriales, guías y manuales de las herramientas utilizadas para el desarrollo de este proyecto,

Por ende, las actividades realizadas durante esta primera fase fueron las siguientes:

- Estudio de las herramientas involucradas en la solución.
- Estudio de las funcionalidades existentes en el sistema legado.
- Evaluar reglas de negocios.

#### **Fase II: Fase de Diseño.**

En esta fase se dio inicio a la definición de los servicios candidatos, así como los componentes y flujos a ser utilizados a lo largo de este proyecto a través de la información obtenida de la fase de diagnóstico que sirvieron de guía a lo largo de la fase de diseño. A continuación se presentan las actividades realizadas:

- Estudio de la arquitectura definida para el proyecto
- Identificación de servicios candidatos.
- Generación de los contratos de los servicios candidatos

### **Fase III: Fase de Desarrollo y pruebas**

En esta fase se puso en práctica el diseño. Aquí fue donde se diseñaron los componentes de forma detallada, y en donde además se determinó cuáles de los servicios eran factibles, generando una especificación de servicios para cada uno de ellos. Además, se definió la codificación base64 como codificación para el estándar de los mensajes de los servicios con la finalidad de unificar la serialización del traspaso de mensaje a ser implementado. Adicionalmente a esto, se definió el flujo del proceso del modelo de integración para la interoperabilidad de los sistemas mediante patrones de integración empresariales a ser desarrollado en esta fase.

Para esta fase, se realizaron las siguientes actividades:

- Elaboración del repositorio de servicios.
- Elaboración de la especificación de servicios.
- Elaboración de pruebas a menor escala en el cual se someten los servicios para determinar cuáles de los servicios candidatos eran factibles y cuáles no.
- Generación del documento arquitectónico del proyecto.
- Adaptación e inclusión de los servicios que representaban el alcance.

### **Fase IV: Fase de Realización e Implementación.**

En esta fase, en primer lugar, se validaron las decisiones de realización haciendo una exploración de viabilidad sobre las decisiones arquitectónicas y los factores de riesgo del proyecto a través de pruebas en un servidor destinado para tal fin. Luego, durante la implementación, tanto los servicios como los procesos fueron desarrollados de acuerdo con las decisiones derivadas de la fase de desarrollo y pruebas.

Finalmente, los servicios son orquestados y los flujos de mensaje son establecidos. Las actividades correspondientes son las siguientes:

- Desarrollo de un prototipo con la finalidad de mitigar riesgos y aclarar dudas técnicas de las herramientas así como de la comunicación con los sistemas legados.
- Definición del protocolo de intercambio de mensajes con el sistema legado.
- Integración del prototipo con el sistema legado.
- Desarrollo completo del modelo de integración para la interoperabilidad de los sistemas mediante patrones de integración empresariales.

## **CAPITULO IV RESULTADOS**

En este capítulo se explica de forma detallada todo el proceso de realización del proyecto. El mismo se dividirá según las fases presentadas en el marco metodológico.

### **4.1 Modelado**

La primera fase fue dedicada al entendimiento de los requerimientos y conocer en un primer acercamiento a las herramientas que utilizaremos. A continuación se explica detalladamente las actividades realizadas.

#### **4.1.1 Herramientas involucradas en la solución.**

Se realizó un estudio de las posibles herramientas que se ajusten a la solución y a través de ese estudio se definió el conjunto de herramientas que fueron utilizadas para desarrollar el proyecto. Las herramientas utilizadas se describen brevemente a continuación:

- **Java:** Este lenguaje fue utilizado para desarrollar todas las tareas de integración y de orquestación, así como también los servicios web.
- **Apache Karaf:** es un contenedor de servicios moderno y polimórfico. Es ligero, de gran alcance, y es un contenedor de gama empresarial listo para usarse en producción. Pudiendo implementar diferentes aplicaciones en Karaf utilizando un contenedor para cada una de las aplicaciones. Con esta flexibilidad, Karaf es el recipiente perfecto para microservicios, integración de sistemas, datos grandes, y mucho más. El cual será utilizado para realizar la implementación de los servicios de la solución.
- **Apache Camel:** es un motor de enrutamiento y mediación basado en reglas que provee una implementación basada en objetos Java de los patrones propuestos en de integración empresarial, ya sea empleando un API o bien un lenguaje específico del dominio declarativo expresado en Java, para configurar las reglas de enrutamiento y mediación. Este será el motor implementado para establecer la orquestación entre servicios permitiendo realizar flujos de operaciones complejas partiendo de operaciones atómicas.
- **Apache ActiveMQ:** es un intermediario de mensajería de código abierto que implementa plenamente la especificación de Java Message Service (JMS). Ofrece la implementación de patrones empresariales de integración tales como clustering, múltiples almacenes para mensajes, así como la capacidad de emplear

cualquier administrador de base de datos como proveedor de persistencia JMS, caché y persistencia de journals. Esta herramienta se implementó dentro del bus de servicios empresarial apoyándose en las rutas de mediación para gestionar el traspaso de mensajes entre operaciones de una forma segura y organizada gestionando colas de mensajes e incluso en algunas operaciones de forma transaccional.

- **KahaDB:** es una base de datos de persistencia de archivos basada en un ambiente local con el intermediario de mensajes. Se ha optimizado para la persistencia rápida y es el mecanismo del almacenamiento predeterminado de ActiveMQ en las versiones más recientes. KahaDB utiliza menos descriptores de archivos lo que optimiza el almacenamiento y proporciona una recuperación más rápida ante errores de traspaso de mensajes. Se utilizó la configuración por defecto del intermediario de mensajería por su alta compatibilidad con esta herramienta de persistencia de datos y su alta compatibilidad con el intermediario de mensajería.
- **C#:** es un lenguaje de programación que se ha diseñado para compilar diversas aplicaciones que se ejecutan en .NET Framework. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Se utilizó esta herramienta debido a que las operaciones de los servicios web fue pensada para ser consumido bajo el sistema operativo Window haciendo esta herramienta de gran utilidad debido a su compatibilidad nativa con este sistema operativo. Bajo esta herramienta se construyeron los clientes que consumirán las operaciones de los servicios web tanto como sus rutinas de consola para una ejecución desde el programador de tareas del sistema operativo.

#### **4.1.2 Funcionalidades existentes en el sistema legado.**

En este punto se estudiaron las características del sistema utilizado hasta los momentos por la entidad. De esta manera, se obtuvo una visión más clara del proceso deseado, así como también, de las funcionalidades con las que contaba para saber cuáles de ellas podían ser reutilizadas y cuáles debían ser agregadas con la finalidad de optimizar el proceso. La reutilización es uno de los factores importantes de SOA, ya que representa una mejora en cuanto al tiempo de desarrollo y de la calidad del sistema, ya que permite obtener funciones específicas que son relevantes. Durante esta fase del estudio se consideró utilizar una herramienta de ingreso de data seguro conocido como “DataExchange” que ofrece el sistema

legado de ICG que es el sistema involucrado en el proceso de integración y así mantener la integridad de los datos y transparencia en el proceso de interacción con el sistema legado.

#### **4.1.3 Reglas de negocios.**

En la evaluación de las reglas de negocios se realizó un estudio en el cual se basan las operaciones que se exponen en los servicios web para satisfacer la necesidad de los procesos y así cumplir con los objetivos e incluso se adicionaron reglas que permiten la optimización de los procesos como lo son la validación de los datos consultados por las operaciones.

#### **4.2 Diseño.**

En esta fase se dio inicio a la definición de los servicios candidatos, así como los componentes y flujos a ser utilizados a lo largo de este proyecto a través de la información obtenida de la fase de diagnóstico que sirvieron de guía a lo largo de la fase de diseño. A continuación se presentan las actividades realizadas

##### **4.2.1 Estudio de la arquitectura definida para el proyecto.**

En la siguiente figura se aprecian los elementos involucrados en la arquitectura de la solución SOA. En primer lugar, se encuentra el bus de servicios empresarial, que es la infraestructura de software que permite ejecutar los servicios web. Entonces, cuando una operación es requerida, se inicia el proceso. Desde el proceso son llamados una serie de servicios expuestos desde el bus de servicios empresarial, que funge como orquestador de estos y además como mediador entre los procesos y los sistemas legados involucrados en el proceso de integración.

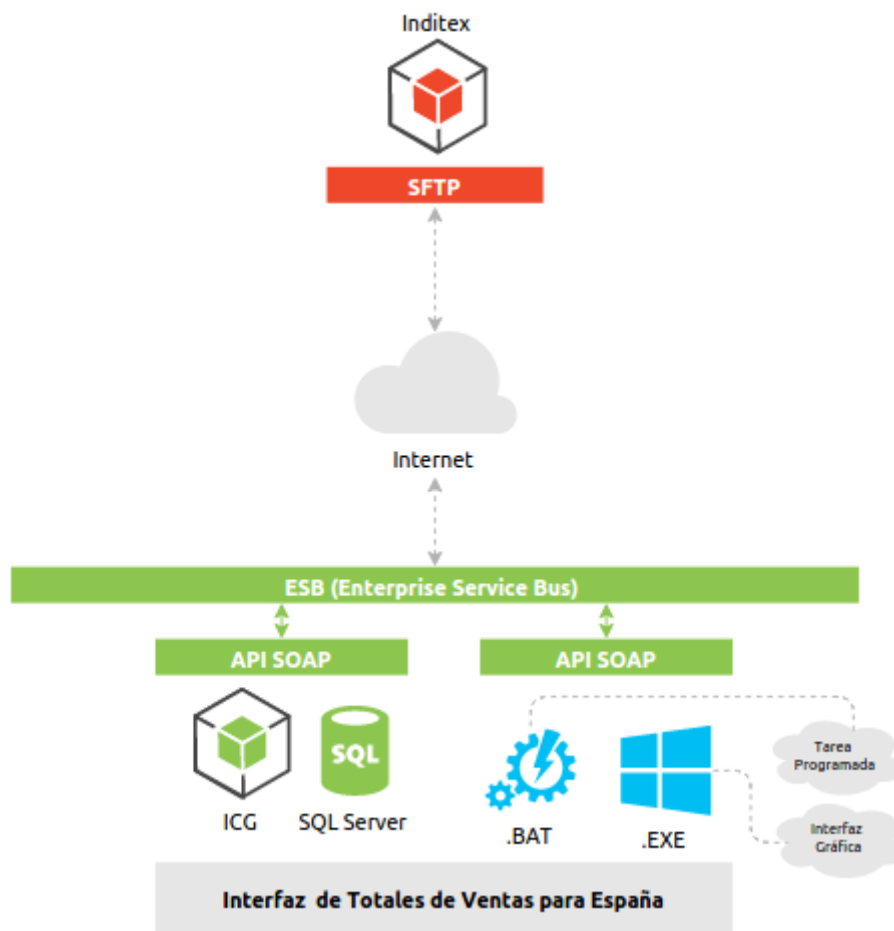


Gráfico No.1 – Elementos involucrados en la arquitectura de la solución (ESB-MQ). Fuente: Victor Archila, (2015)

#### 4.2.2 Identificación de servicios candidatos.

Habiendo analizado el sistema actual, se consideró una serie de servicios candidatos que podrían ser incluidos en la solución. Con el objetivo de optimizar el portafolio de servicio, cada servicio fue analizado en función de una serie de criterios tanto técnicos como funcionales y se identificó cuáles servicios debían ser expuestos. Los criterios utilizados para la selección de los servicios basados en las mejores prácticas de SOA se describen a continuación:

- a) **Alineación al negocio:** Evalúa el nivel de correspondencia entre el negocio y el servicio, basado en los siguientes puntos:
  - Funcionalidad empresarial de soporte a los procesos y objetivos empresariales.
  - Intención de financiar el servicio en todo su ciclo de vida.

- Uso y exposición del servicio hacia socios externos y/o potenciales clientes.
  - Existencia actual y en el futuro de reutilización del servicio.
- b) **Servicio Componible:** Determina la posibilidad que el servicio participe en la composición de otros servicios los cuales son determinados por los siguientes puntos:
- El servicio soporta los requerimientos no funcionales.
  - El servicio es sin estado no deben guardar información alguna sobre datos de sesión, ni sobre eventos previos, ni sobre resultados de servicios invocados previamente.
  - El servicio es independiente y puede cumplir por si solo un objetivo empresarial.
  - El servicio puede cooperar con otros servicios para cumplir objetivos empresariales.
  - La tecnología de implementación del servicio es neutral.
- c) **Descripción:** Evalúa las capacidades del servicio de poder auto describirse, basados en los siguientes puntos:
- El servicio posee una descripción externa la cual es distinta e independiente de la implementación física subyacente.
  - El servicio puede ser descubierto y enlazado con la descripción del mismo.
  - Contiene la descripción del servicio metadatos que permite la identificación del mismo.
- d) **Reusabilidad:** Evalúa las capacidades del que el servicio pueda ser reusado por diferentes procesos de negocio.
- e) **Viabilidad Técnica:** La viabilidad técnica asegura que el servicio se pueda implementar y desplegar, basado en los siguientes puntos:
- El esfuerzo de implementación y gestión del servicio es razonable y fácilmente viable.
  - El servicio puede ser implementado por un número mínimo de sistema proveedores.
  - No existen retos tecnológicos explícitos conocidos ni anticipados.

De esta forma, se obtuvo una serie de servicios que fueron los considerados para el desarrollo de la solución.

#### 4.2.3 Generación de los contratos de los servicios candidatos.

Durante esta etapa se modelaron los contratos de los servicios web que especifica la forma en cómo comunicarse con las operaciones que pertenecen a cada uno de los servicios expuestos. Para la descripción de los contratos se usó el

lenguaje de descripción de servicios web (WSDL) que aplica un formato XML que se utiliza para describir servicios Web.

WSDL describe la interfaz pública de los servicios Web, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.

```
-<wSDL:definitions name="ICGInditex" targetNamespace="http://www.bestboon.com/service/">
  -<wSDL:types>
    -<xsd:schema targetNamespace="http://www.bestboon.com/service/">
      -<xsd:element name="Out_VentasTotalesTiendaB64Request">
        -<xsd:complexType>
          -<xsd:sequence>
            <xsd:element name="fecha_inicial" type="xsd:string"/>
            <xsd:element name="fecha_final" type="xsd:string" /> </xsd:element>
            <xsd:element name="tienda" type="xsd:string" /> </xsd:element>
            <xsd:element name="caja" type="xsd:string" /> </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    -<xsd:element name="Out_VentasTotalesTiendaB64Response">
      -<xsd:complexType>
        -<xsd:sequence>
          <xsd:element name="archivos" type="xsd:string"/>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  </wSDL:types>
</wSDL:definitions>
```

*Gráfico No.2 – Extracto de ejemplo de un contrato de servicio (WSDL).* Fuente: Victor Archila (2015)

### 4.3 Desarrollo y pruebas

En esta fase se pone en práctica el diseño. Aquí es donde se diseñaron los componentes de forma detallada, y en donde además se determinó cuáles de los servicios eran factibles, generando una especificación de servicios para cada uno de ellos.

#### 4.3.1 Repositorio de servicios.

Una vez obtenidos los servicios que resultaron factibles, se procedió a realizar el repositorio de servicios. En éste paso, para cada servicio identificado se especificaron los siguientes puntos.

- **Nombre:** Nombre que identifica el servicio.
- **Descripción:** Una breve especificación del significado de cada servicio.

- **Operaciones del servicio:** Esta sección contiene una descripción de las diferentes operaciones que contiene el servicio para exponer su funcionalidad.
- **Vista general de las operaciones:** Se representa el flujo principal de la función que se realiza en las operaciones expuestas.

|   |   |
|---|---|
| <b>ICGInditexPortType</b> <ul style="list-style-type: none"> <li>• Out_VentasTotalesTienda</li> <li>• Out_VentasTotalesTiendaB64</li> </ul>   | <b>Endpoint address:</b> <a href="http://localhost:8040/services/ICGInditex">http://localhost:8040/services/ICGInditex</a><br><b>WSDL :</b> <a href="http://www.bestboon.com/service/ICGInditex">http://www.bestboon.com/service/ICGInditex</a><br><b>Target namespace:</b> <a href="http://www.bestboon.com/service/">http://www.bestboon.com/service/</a>   |
| <b>bB_Ambar_WSPortType</b> <ul style="list-style-type: none"> <li>• ingresar_articulos_icg</li> <li>• historial</li> <li>• crear_tablas_sistema</li> <li>• procesar_maestra_articulos</li> <li>• validar_reglas_negocios</li> <li>• evento</li> </ul> | <b>Endpoint address:</b> <a href="http://localhost:8040/services/bB_Ambar_WS">http://localhost:8040/services/bB_Ambar_WS</a><br><b>WSDL :</b> <a href="http://www.bestboon.com/service/bB_Ambar_WS/bB_Ambar_WS">http://www.bestboon.com/service/bB_Ambar_WS/bB_Ambar_WS</a><br><b>Target namespace:</b> <a href="http://www.bestboon.com/service/bB_Ambar_WS/">http://www.bestboon.com/service/bB_Ambar_WS/</a> |
| <b>bB_FTP_WSPortType</b> <ul style="list-style-type: none"> <li>• crear_tabla_log</li> <li>• descargar</li> <li>• listar_archivos</li> <li>• evento</li> <li>• cargar</li> <li>• listar_archivos_nuevos</li> <li>• historial</li> </ul>               | <b>Endpoint address:</b> <a href="http://localhost:8040/services/bB_FTP_WS">http://localhost:8040/services/bB_FTP_WS</a><br><b>WSDL :</b> <a href="http://www.bestboon.com/service/bB_FTP_WS/bB_FTP_WS">http://www.bestboon.com/service/bB_FTP_WS/bB_FTP_WS</a><br><b>Target namespace:</b> <a href="http://www.bestboon.com/service/bB_FTP_WS/">http://www.bestboon.com/service/bB_FTP_WS/</a>                 |
| <b>bB_SFTP_WSPortType</b> <ul style="list-style-type: none"> <li>• evento</li> <li>• cargar</li> <li>• historial</li> <li>• descargar</li> <li>• crear_tabla_log</li> </ul>   | <b>Endpoint address:</b> <a href="http://localhost:8040/services/bB_SFTP_WS">http://localhost:8040/services/bB_SFTP_WS</a><br><b>WSDL :</b> <a href="http://www.bestboon.com/service/bB_SFTP_WS">http://www.bestboon.com/service/bB_SFTP_WS</a><br><b>Target namespace:</b> <a href="http://www.bestboon.com/service/">http://www.bestboon.com/service/</a>   |

Gráfico No.3 – Representación del repositorio de los servicios y operaciones disponibles. Fuente: Víctor Archila (2015)

#### 4.3.2 Especificación de servicios.

A continuación, se da una descripción detallada de cada uno de los servicios expuestos y sus operaciones:

##### a) ICGInditex

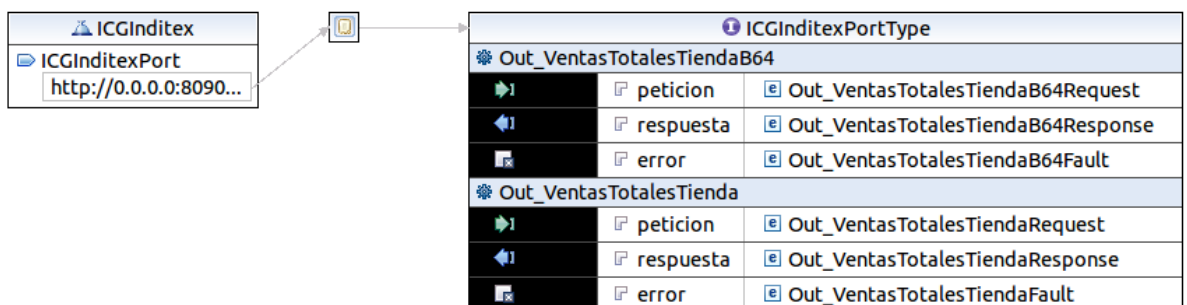


Gráfico No.4 – Vista general del servicio web ICGInditex. Fuente: Víctor Archila (2015)

- **Descripción:** Servicio que permite consultar las ventas totales por caja de las tiendas (Zara, Beshka, Pull&Bear) a nivel nacional y reportar la data generada al servidor SFTP de Inditex España.
- **Operaciones del servicio:**
  - **Out\_VentasTotalesTienda:**
    - **Descripción:** Esta operación permite consultar el total de tickets emitidos por caja o el total de ventas por caja
    - **Vista general de la operación:**

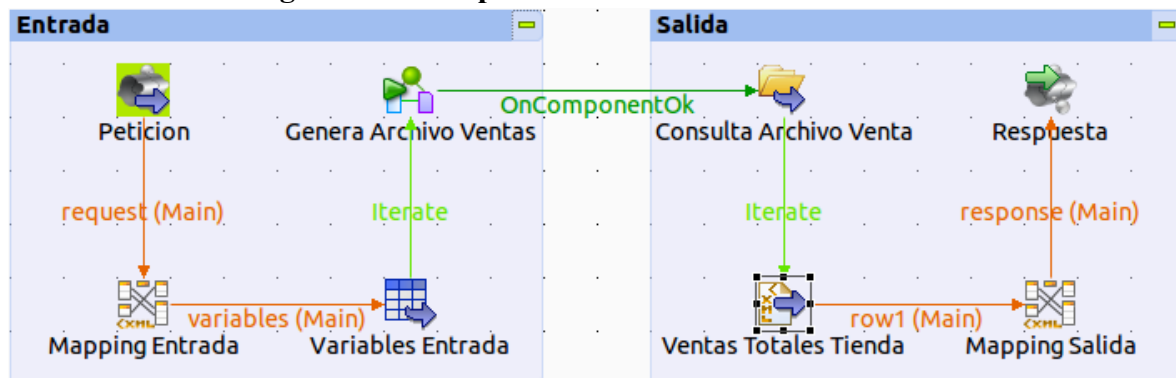


Gráfico No.5 – Flujo principal de la operación **Out\_VentasTotalesTienda**. Fuente: Victor Archila (2015)

- **Out\_VentasTotalesTiendaB64**
  - **Descripción:** Esta operación realiza la misma consulta que la operación anterior solo que la respuesta devuelve el contenido como una cadena de caracteres codificado en Base64 y está especialmente optimizado para ser utilizado en la cola de mensajes del intermediario de mensajería

▪ Vista general de la operación:

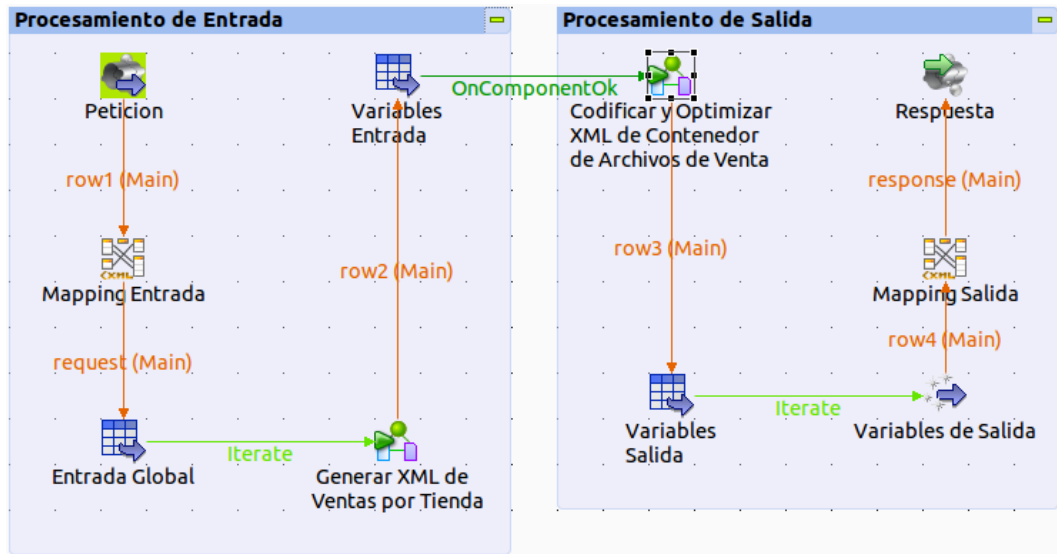


Gráfico No.6 – Flujo principal de la operación Out\_VentasTotalesTiendaB64.  
Fuente: Victor Archila (2015)

b) bB\_Ambar\_WS

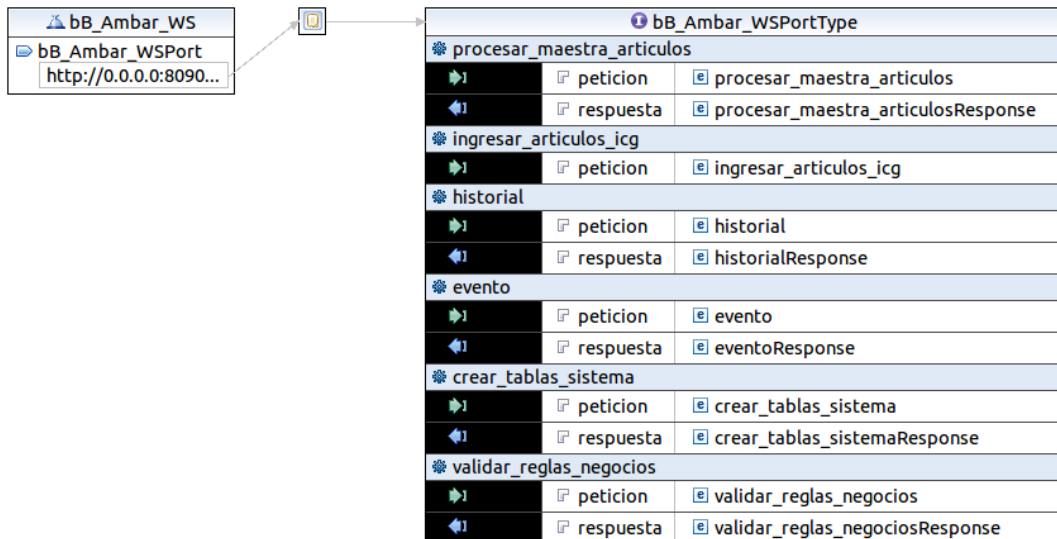


Gráfico No.7 – Vista general del servicio web bB\_Ambar\_WS. Victor Archila (2015)

- **Descripción:** Servicio que permite procesar las actualizaciones de los artículos recibidas desde Inditex España y validarlos contra los existentes en la base de datos del sistema legado de ICG y adicionalmente permite

ingresar la data validada al sistema ICG de forma segura utilizando la aplicación “DataExchange”.

- **Operaciones del servicio:**

- **Procesar\_maestra\_articulos:**

- **Descripción:** Recibe los archivos de data maestra y de detalle de artículos de Inditex España y los fusiona en un solo archivo.
    - **Vista general de la operación:**

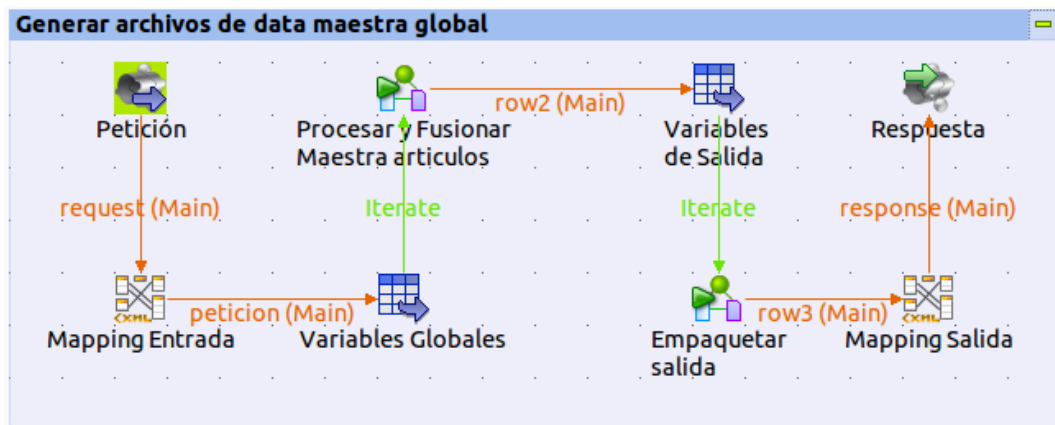


Gráfico No.8 – Flujo principal de la operación **Procesar\_maestra\_articulos**. Fuente:Victor Archila (2015)

- **Ingresar\_articulos\_ICG**

- **Descripción:** Invocación al DataExchange como un subproceso para el ingreso seguro de artículos a las base de datos de empresas de ICG.
    - **Vista general de la operación:**

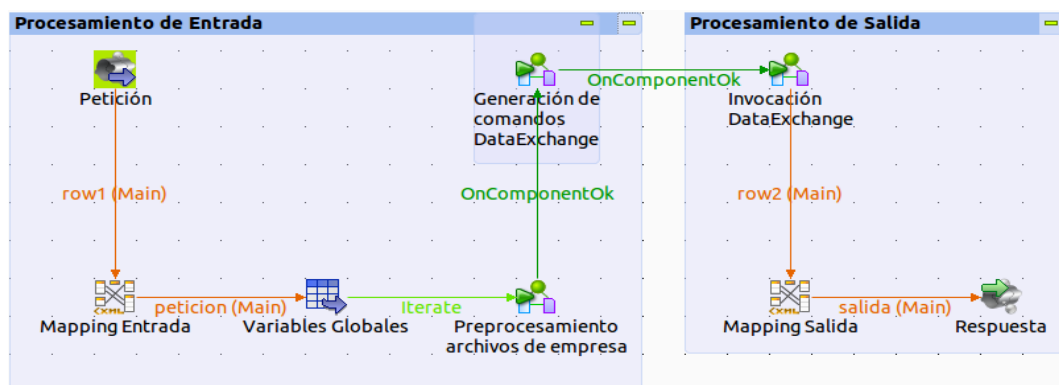


Gráfico No.9 – Flujo principal de la operación **Ingresar\_articulosICG**. Fuente: Victor Archila (2015)

- **Historial**

- **Descripción:** Permite consultar los registros del log de auditoria del servicio.
- **Vista general de la operación:**

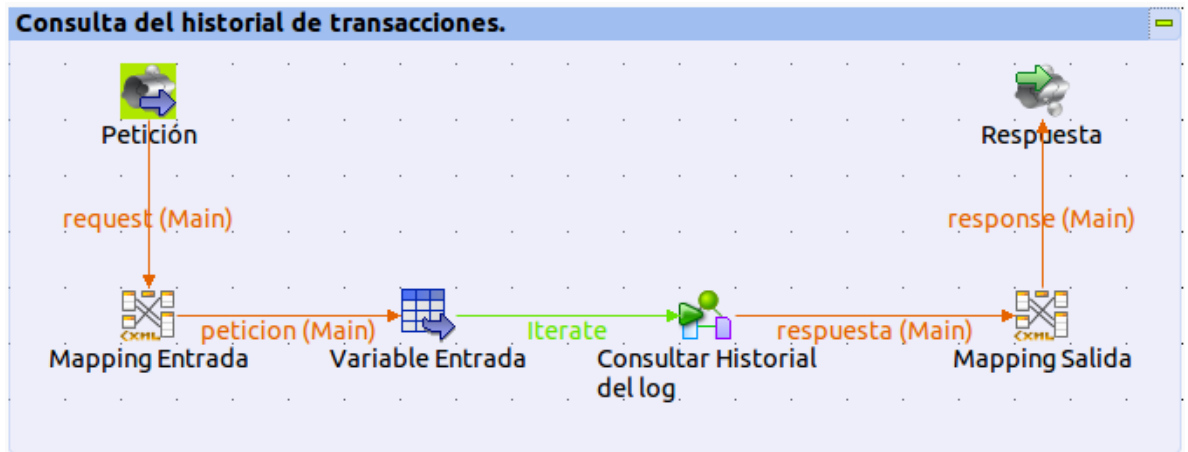


Gráfico No.10 – Flujo principal de la operación Historial. Fuente: Victor Archila (2015)

- **Evento**

- **Descripción:** Permite escribir un registro en el log de auditoria del servicio.
- **Vista general de la operación:**

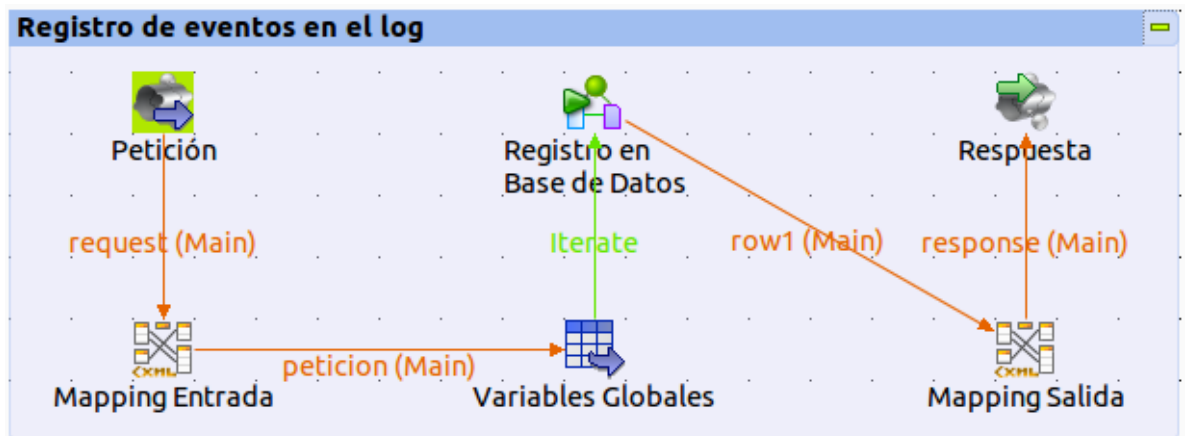


Gráfico No.11 – Flujo principal de la operación Evento. Fuente: Victor Archila (2015)

- **Crear\_tablas\_sistema:**

- **Descripción:** Permite crear las tablas necesarias para el servicio incluyendo la tabla de log de auditoria.

- Vista general de la operación:

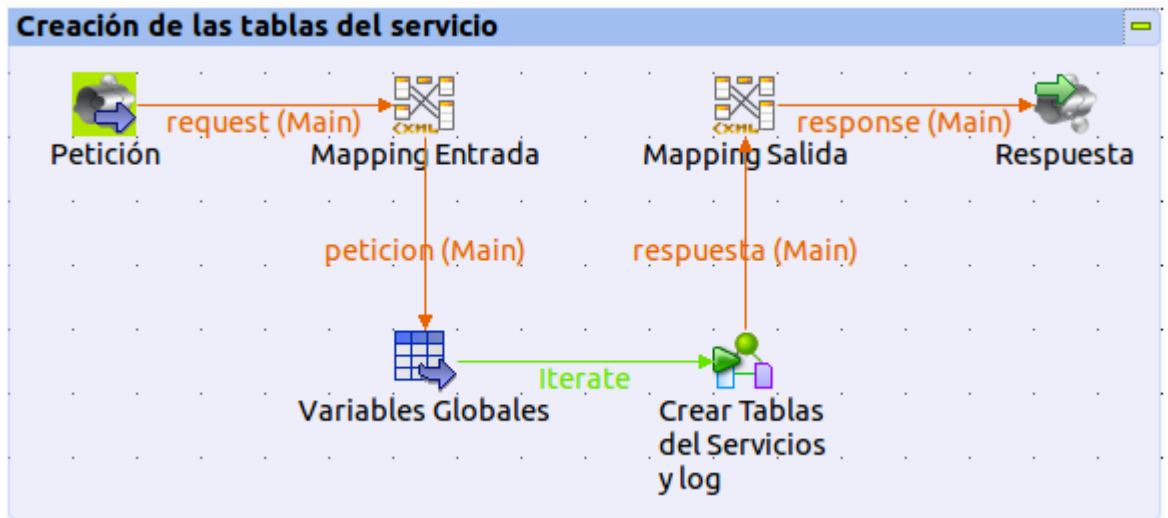


Gráfico No.12 – Flujo principal de la operación Crear\_tablas\_sistema.  
Fuente: Victor Archila (2015)

- Validar\_reglas\_negocios

- Descripción: Aplica las reglas de negocio de Inditex España sobre la maestra de artículos recibida para depurar la información para su posterior ingreso a ICG.
- Vista general de la operación:



Gráfico No.13 – Flujo principal de la operación Validar\_reglas\_negocios.  
Fuente: Victor Archila (2015)

c) bB\_SFTP\_WS

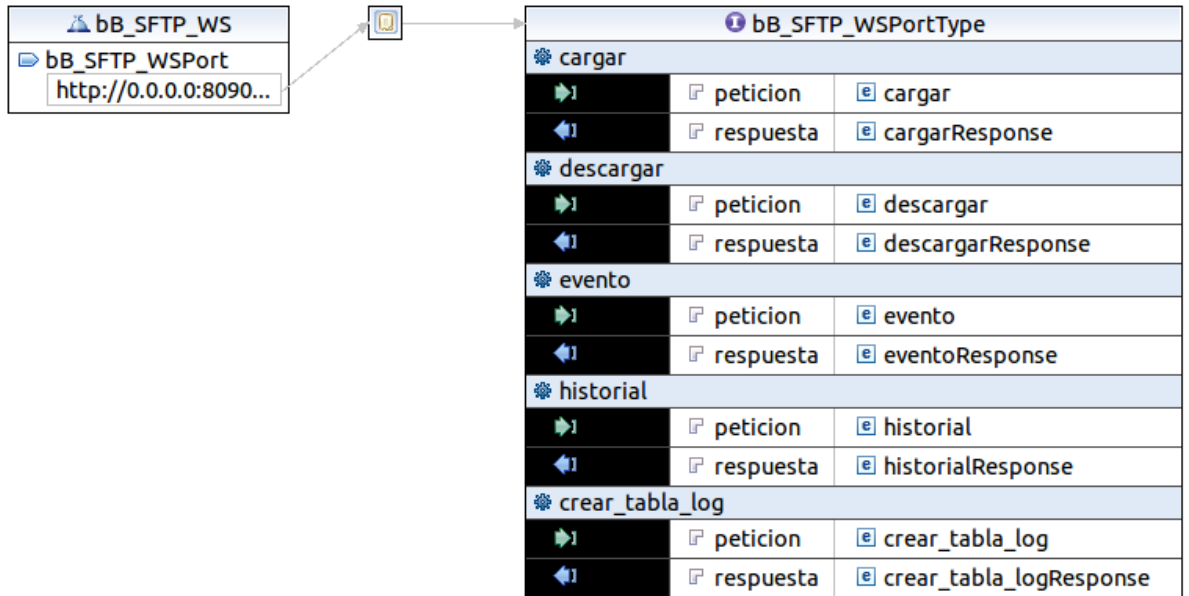


Gráfico No.14 – Vista general del servicio web SFTP. Fuente: Victor Archila (2015)

- **Descripción:** Servicio que permite operar sobre un servidor SFTP, Se configuro para establecer un canal con el servidor SFTP de Inditex España para la subida de los reportes de tienda.
- **Operaciones del servicio:**
  - **Cargar**
    - **Descripción:** Permite subir un archivo al servidor SFTP configurado.
    - **Vista general de la operación:**

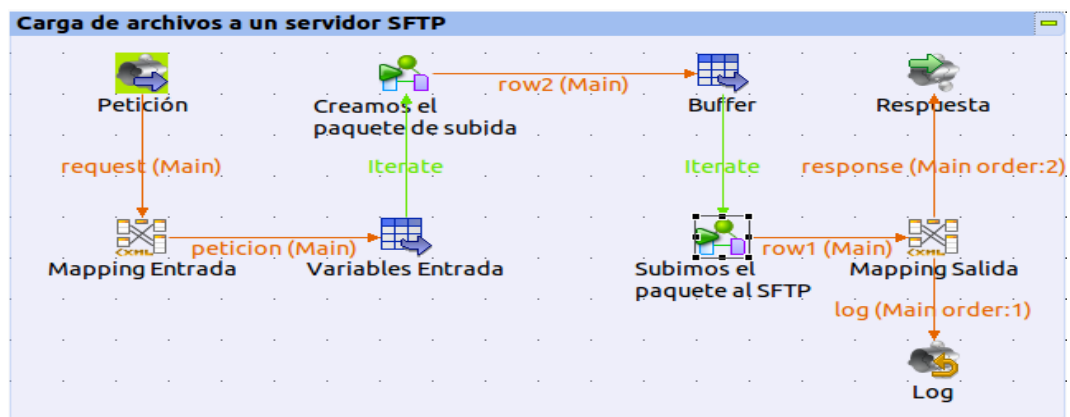


Gráfico No.15 – Flujo principal de la operación Cargar. Fuente: Victor Archila (2015)

- **Descargar**

- **Descripción:** Permite descargar un archivo desde el servidor SFTP configurado.
- **Vista general de la operación:**

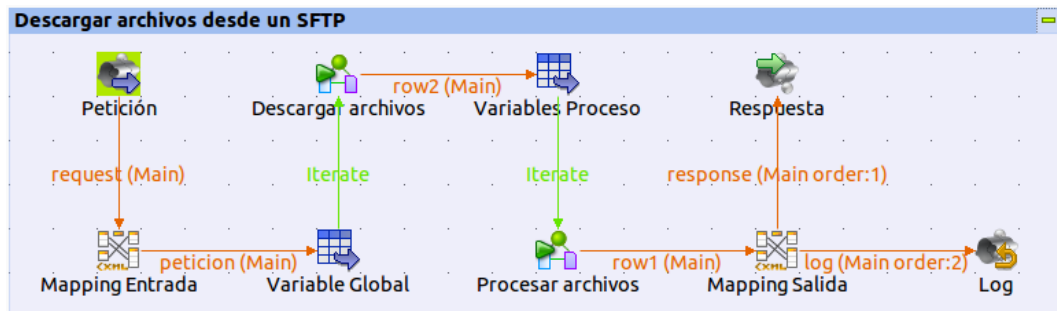


Gráfico No.16 – Flujo principal de la operación Descargar. Fuente: Victor Archila (2015)

- **Evento**

- **Descripción:** Permite escribir un registro en el log de auditoria del servicio.

- **Historial**

- **Descripción:** Permite consultar los registros del log de auditoria del servicio.

- **Crear\_tablas\_sistema:**

- **Descripción:** Permite crear las tablas necesarias para el servicio incluyendo la tabla de log de auditoria.

#### d) bB\_FTP\_WS

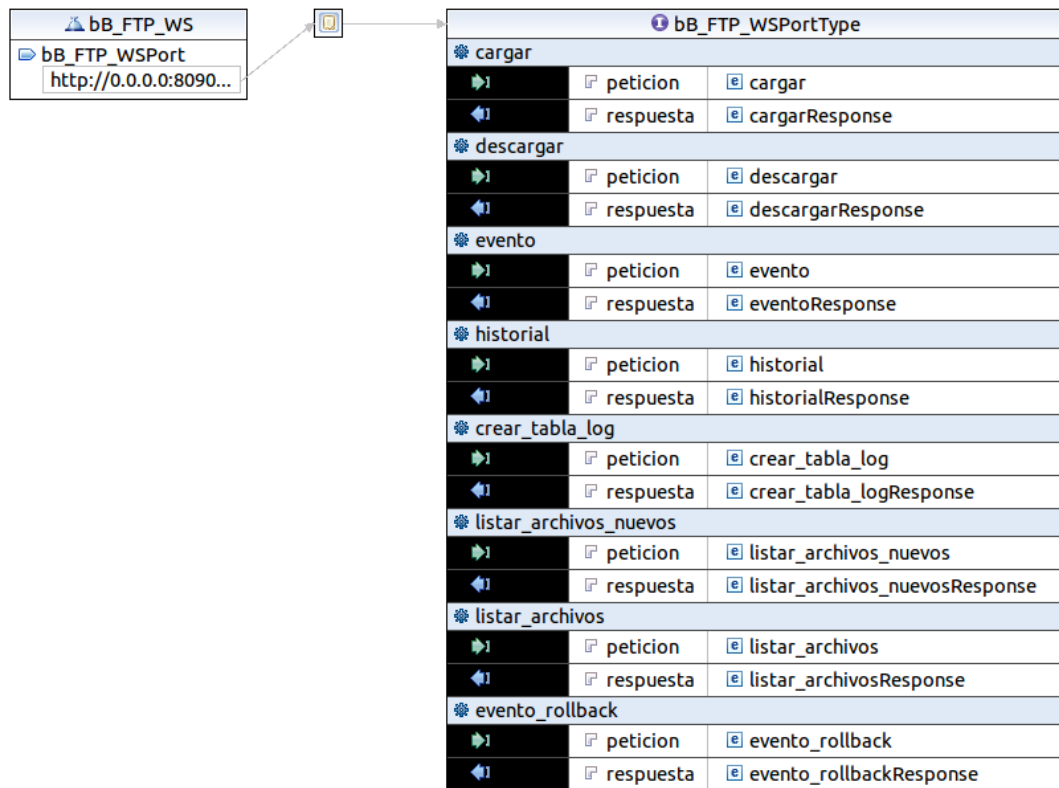


Gráfico No.17 – Vista general del servicio web FTP. Fuente: Victor Archila (2015)

- **Descripción:** Servicio que permite operar sobre un servidor FTP, que se configurò para establecer un canal con el servidor FTP de Inditex España para la subida de los reportes de tienda
- **Operaciones del servicio:**
  - **Cargar**
    - **Descripción:** Permite subir un archivo al servidor FTP configurado.
  - **Descargar**
    - **Descripción:** Permite descargar un archivo desde el servidor FTP configurado.
  - **Evento**
    - **Descripción:** Permite escribir un registro en el log de auditoria del servicio.
  - **Historial**
    - **Descripción:** Permite consultar los registros del log de auditoria del servicio.

- **Crear\_tablas\_sistema:**
  - **Descripción:** Permite crear las tablas necesarias para el servicio incluyendo la tabla de log de auditoria.
- **Listar\_archivos\_nuevos**
  - **Descripción:** Permite obtener un listado de archivos del servidor FTP comparando contra los marcados como descargados en la tabla de log del servicio para determinar el conjunto de los archivos nuevos.
  - **Vista general de la operación:**

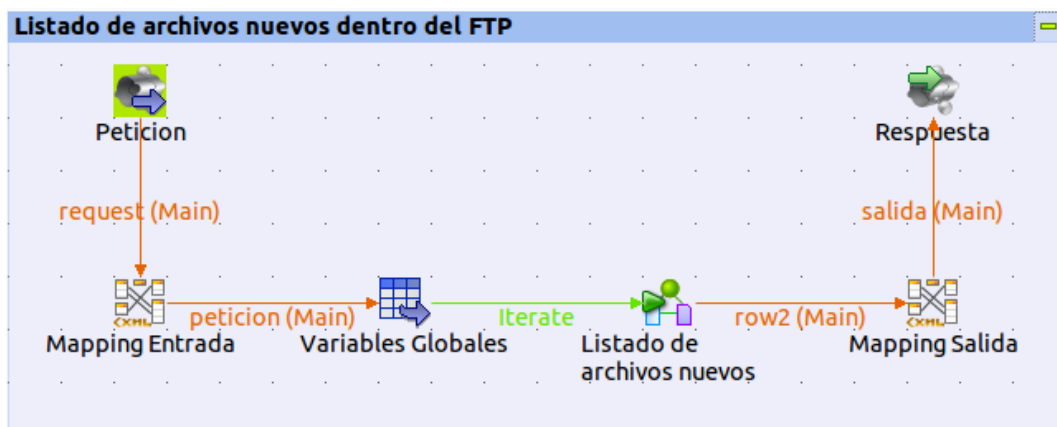


Gráfico No.18 – Flujo principal de la operación Listar\_archivos\_Nuevos. Fuente: Victor Archila (2015)

- **Listar\_archivos**
  - **Descripción:** Permite obtener un listado de archivos del servidor FTP.
  - **Vista general de la operación:**

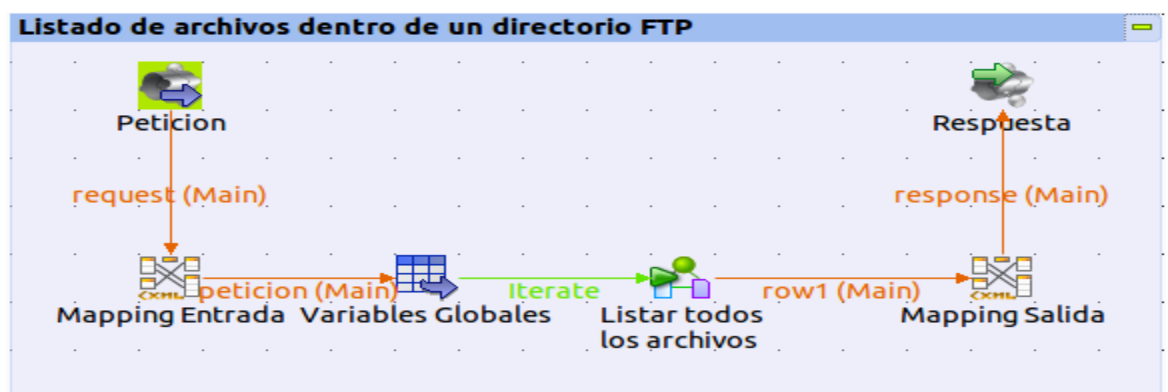


Gráfico No.19 – Flujo principal de la operación Listar\_archivos. Fuente Victor Archila (2015)

- **Evento\_rollback**

- **Descripción:** Permite eliminar un archivo de la tabla de log que haya sido descargado para su reprocesamiento.
- **Vista general de la operación:**

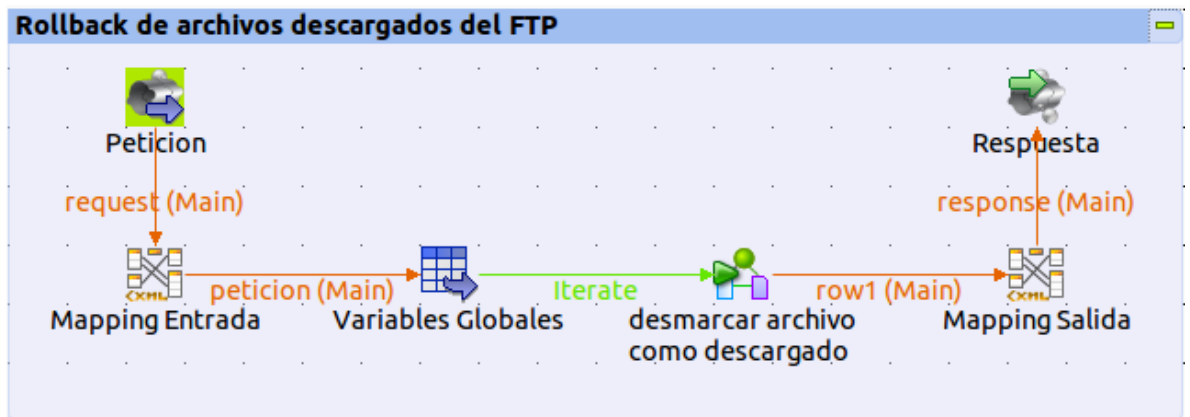


Gráfico No.20 – Flujo principal de la operación Evento\_rollback. Fuente: Victor Archila (2015)

#### 4.3.3 Pruebas a menor escala

Con el objetivo de optimizar el portafolio de servicio, cada servicio fue analizado mediante una herramienta llamada SOAPUI. La cual permite consumir individualmente las operaciones de los servicios expuestos y función de una serie de criterios tanto técnicos como funcionales se determinó la factibilidad de los servicios candidatos.

#### 4.3.4 Documento arquitectónico del proyecto.

En el documento arquitectónico del proyecto se estableció como los servicios interactúan a través de rutas de mediación para realizar operaciones más complejas aplicando la funcionalidad específica de cada operación en conjunto con otras.

Para la primera función compuesta encontramos el proceso de reporte hacia Inditex de las ventas del total por tiendas donde a continuación se representa el flujo de orquestación y la lógica implementada en la ruta de mediación para la gestión de la cola de mensajes del canal de subida de archivos al servidor SFTP de Inditex España.

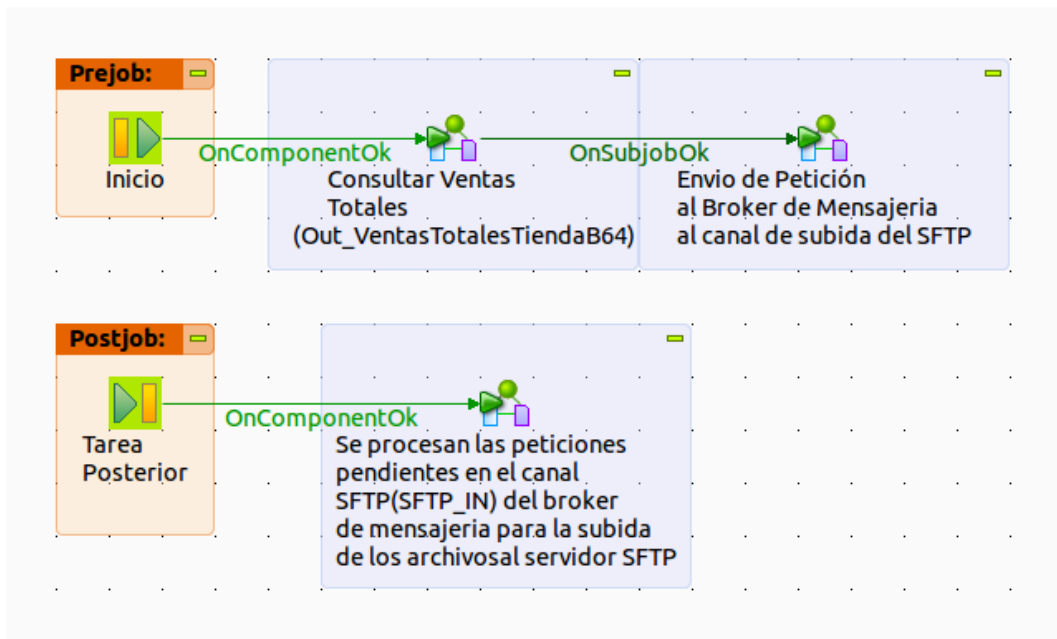


Gráfico No.21 – Flujo principal de la operación envío de reportes Inditex España. Fuente: Victor Archila (2015)

**Ruta de mediación (SFTP\_channel\_with\_log):** Esta ruta de mediación permite consumir los mensajes que se encuentren en cola en el canal de mensajería SFTP\_IN para procesar las peticiones de subida de archivos al servidor SFTP de Inditex España.

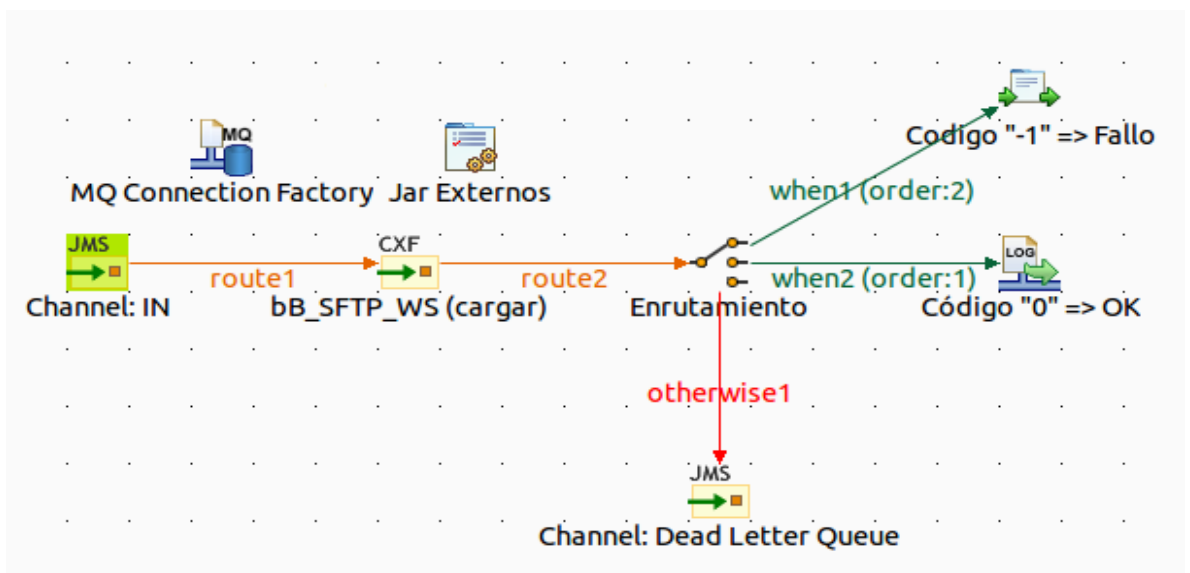
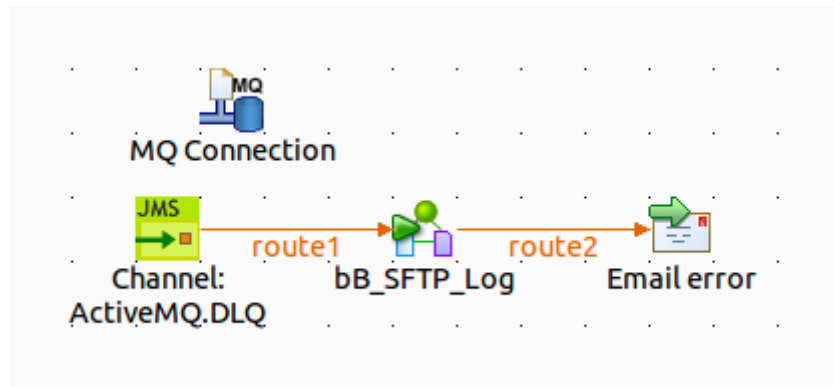


Gráfico No.22 – Flujo principal de la ruta de mediación SFTP\_channel\_with\_log. Fuente: Victor Archila (2015)

**Ruta de mediación (sftp\_dead\_letter\_queue\_log):** Esta ruta de mediación se encarga de capturar los mensajes que no pudieron ser procesados de la cola del canal de mensajería SFTP\_IN para notificar por correo que los mensajes no pudieron subir al servidor SFTP de Inditex España y así no interferir con el procesamiento de los mensajes en la cola y poder tomar las acciones pertinente con los mensajes fallidos ya sea que se procesen manualmente o se reenvíen a la cola para intentar ser procesados nuevamente .



*Gráfico* No.23 – **Flujo principal de la ruta de mediación sftp\_dead\_letter\_queue\_log.** Fuente: Victor Archila (2015)

#### **4.4 Realización e Implementación.**

Durante esta etapa, tanto los servicios como los procesos fueron desarrollados de acuerdo con las decisiones derivadas de la fase de desarrollo y pruebas. Para la puesta en marcha del sistema, en primera instancia, se realizó una prueba de concepto para mitigar riesgos a la hora de realizar la implementación completa del sistema y, en segunda instancia se realizó el desarrollo del proceso completo. A continuación, se describen cada una de las actividades realizadas durante esta fase:

##### **4.4.1 Desarrollo del prototipo para la Prueba de Concepto.**

Para ejecutar la prueba de concepto, se realizó sobre un servidor de pruebas que fue destinado para este fin y así no comprometer el entorno de producción ni la productividad de la empresa, en el servidor de pruebas se implementaron los servicios web con la finalidad de ser consumidos por una aplicación cliente. A continuación, se describen cada una de las actividades realizadas durante esta fase:

- En una primera etapa, se implementó la infraestructura de software que se refiera al bus de servicios empresarial en el cual se exportaron los servicios desarrollados en la etapa anterior de desarrollo y pruebas para medir la

efectividad del ESB con la finalidad de poner a prueba el diseño y la correcta elaboración del proceso a nivel del servidor de procesos.

- En una segunda etapa, se bajó de nivel los servicios que establecen una tarea transaccional, es decir, se colocaron a nivel del intermediario de mensajería, donde estos servicios serán llamados de forma ordenada a través de una cola de mensajes. En primer momento, se generaban mensajes de petición desde el mismo intermediario de mensajería que respondían directamente con la ejecución del servicio asociado al canal de mediación. La finalidad de esto fue establecer el flujo de mediación entre el intermediario de mensajería y el ESB para tener una base robusta al momento de la comunicación definitiva.
- Finalmente, en una última etapa, se unifico el resultado obtenido de las dos etapas anteriores. Para ello, en primer lugar se exportaron las rutas de mediación al ESB las cuales definen el flujo de orquestación y se definió la implementación en donde se generaban mensajes predefinidos manualmente en el intermediario de mensajería, con la lógica necesaria para establecer la comunicación con el sistema legado.

#### **4.4.2 Protocolo de intercambio de mensajes con el sistema legado.**

Para establecer una comunicación efectiva entre el intermediario de mensajería y el sistema legado, se estableció que se utilizaría la aplicación “DataExchange” como puente de comunicación en conjunto con el personal de la entidad. Se acordó que el sistema legado recibiría los comandos de ejecución previamente consolidados, en donde se encuentra especificado que acción aplicara sobre el sistema legado. Adicionalmente se acordó enviar los paquetes de mensajes codificados en base 64 con la finalidad de mantener la integridad del paquete y por consistencia se aplicó esta regla para los demás servicios. A nivel del intermediario de mensaje, al recibir un mensaje en forma de una cadena codificada en base 64, se procesa su decodificación en la operación del servicio, donde se extrae la información del mensaje y se envía al sistema legado en forma iterativa para ser procesada de forma segura. De igual forma, cuando llega una respuesta del sistema legado, se extrae dicha información y se almacena en el log del servicio para su control de auditoría.

#### **4.4.3 Desarrollo completo del modelo de integración para la interoperabilidad de los sistemas mediante patrones de integración empresariales.**

Una vez realizadas todas las pruebas en el servidor destinado para tal función y estas fueron concluyentes, como pruebas puntuales de concepto, se dio por completado el desarrollo del modelo y se procedió a realizar la implementación en el servidor de prueba para esta tarea se realizaron las siguientes etapas.

En primera instancia se importaron e instalaron las herramientas de integración como lo fue el instalador del bus de servicios empresarial y los instaladores de las aplicaciones clientes de escritorio y aplicaciones de consola para la automatización del consumo de los servicios y así a través de la configuración de estas aplicaciones en el programador de tareas del sistema operativo de Window para poder desencadenar de forma automática las actividades descritas en el documento arquitectural del proyecto que se refiera al flujo principal de la operación compuesta por servicios web expuesto.

Una vez instaladas las herramientas, se agregaron los servicios web al bus de servicios empresarial, esto nos generó simplemente un esqueleto del proceso sin la lógica que ofrecen las rutas de mediación. Antes de implementar la lógica que seguirá el flujo de las operaciones se realizó una prueba de los servicios web instalados que garantizó su efectiva instalación y adaptación en el bus de servicios empresarial para poder proceder a agregar la lógica entre las operaciones para eso se implementaron las rutas de mediación que establecen la forma en cómo se van a gestionar el traspaso de mensajes entre operaciones.

Como etapa final se implementó el intermediario de mensajería que es el elemento que permite gestionar colas de mensajes, que son procesadas de forma ordenada y esta es la herramienta final del proceso de implementación para dar por completado modelo de integración para la interoperabilidad de los sistemas mediante patrones de integración empresariales.

A lo largo de todo el desarrollo de implementación en el entorno de producción se hicieron pruebas tanto internas como en conjunto con el personal de sistemas de la empresa. En primera instancia, al no haber conexión con el sistema legado se simulaba el flujo de mensajes con las herramientas de prueba que provee ActiveMQ y se seguía el flujo de dichos mensajes para detectar errores. Una vez establecida la conexión con el sistema legado, se realizaron pruebas en conjunto con el personal técnico y funcional en las cuáles el sistema enviaba una petición al sistema legado y la manejaba y retornaba la respuesta, y de igual forma se retornaba una salida del sistema legado para determinar que el intercambio de mensajes se estuviera llevando de la mejor forma.

## **CAPITULO V**

### **CONCLUSIONES Y RECOMENDACIONES**

#### **5.1 Conclusiones**

Al culminar el proyecto, se desarrolló un middleware el cual sirve como orquestador de las actividades relacionadas al proceso de reportes de ventas por tienda de la entidad Textiles Zanibar C.A de importancia a nivel nacional. A continuación, se presentan las conclusiones derivadas a la elaboración de este proyecto así como las lecciones aprendidas:

- Al ser la metodología SOMA orientada a servicios, esta garantizó el correcto desempeño de los procesos y del sistema.
- La prueba de concepto fue un paso importante en el desarrollo del proyecto, ya que en ella se resolvió una serie de dificultades que fueron cruciales a la hora de hacer el desarrollo.
- El uso de un bus de servicios empresarial facilitó la implementación de este proyecto, ya que los servicios y los procesos de negocio son de fácil implementación. En donde además, la comunicación con los sistemas legados es transparente al usuario, por lo tanto, el personal de la entidad no necesita saber cómo se da ese intercambio de información, sino simplemente enviar los datos necesarios.
- Las herramientas utilizadas en este proyecto resultaron en una gran ventaja para el desarrollo, ya que tiene numerosas funcionalidades para hacer desarrollos en SOA, y en donde además, gran parte del desarrollo se basó en herramientas bajo la fundación apache, lo que redujo en gran medida la complejidad a la hora de acoplar las herramientas por su alto nivel de compatibilidad.
- La documentación adecuada por parte de la entidad permitió llevar una planificación adecuada y un registro de todas las actividades realizadas durante todas las fases del proyecto lo cual fue un punto importante en el cual se pudo contemplar toda la funcionalidad requerida en los servicios propuestos.
- Con el desarrollo de este proyecto se pudo implementar un modelo para la interoperabilidad de sistemas mediante la construcción de interfaces con el uso de patrones de integración empresariales el cual ofrece a la entidad poder estandarizar la forma en como los sistemas interactúan, dándoles una infraestructura de software robusta y altamente escalable. Lo que les permite ser más competitivos debido a que ya establecida la infraestructura de software necesaria es la base para pensar adquirir nuevas tecnologías que ofrezcan un valor agregado a la entidad e inclusive considerar planificar

incluir los procesos más importantes de las áreas de negocios como servicios para así unificar los procesos de la empresa a la arquitectura orientada a servicios.

## **5.2 Recomendaciones**

A continuación se presentan una serie de recomendaciones para futuros desarrollos posteriores a este proyecto, basados en la experiencia adquirida durante el modelado de procesos para el sistema. Al seguir estas recomendaciones, se podrá seguir optimizando los procesos de la entidad con el fin de seguir extendiendo el modelo no solo horizontalmente sino verticalmente adquiriendo más herramientas tecnológicas que complementen el modelo para seguir escalando en funcionalidad con respecto a la entidad ya que el modelo ofrece las bases para desarrollar proyectos posteriores.

- Para agregar nuevos servicios al modelo del proyecto, se recomienda desarrollar los servicios en primer lugar a nivel de diseño de las operaciones a desempeñar y luego enfocarse en implementar la funcionalidad y la lógica correspondiente, ya que, si a nivel de los servicios ya se tiene el archivo WSDL generado, se simplifica en gran medida su desarrollo y la implementación a nivel del ESB.
- A la hora de definir el protocolo de intercambio de mensajes entre servicios, se debe establecer la codificación a ser utilizada (serialización) en base 64, para seguir manteniendo la consistencia en el traspaso de mensajes utilizada a la largo del proyecto y poder evitar generar problemas de comunicación que son difíciles de detectar.
- En temas de tecnología y gestión empresarial, se recomienda implementar un proyecto a futuro de BPM que son las siglas de Business Process Management. siendo BPM una decisión de transformación organizacional, vista como un mecanismo o herramienta de gestión empresarial que permite alinear las diferentes dimensiones (organización, procesos, conocimiento y estrategia), trasladando a la organización de una gestión funcional a una gestión de procesos de negocio exitosa. En el que la adquisición de este tipo de herramienta es factible debido a que BPM se apoya en soap siendo un modelo SOA-BPM, el cual ya hemos implementado la infraestructura de software necesaria para cumplir con la arquitectura SOA que es el modelo implementado en este proyecto que satisface las necesidades de esta herramienta propuesta para un desarrollo a futuro para la entidad.

## REFERENCIAS

- Alves, P., Foti, P. y Scalone, M. (2006). **Proyecto Batuta -Generador de Aplicaciones Orquestadoras**. s.l.: Facultad de Ingeniería - Universidad de la República, 2006. Versión 1.2.1.
- Arsanjani, A. (2004). **Service-oriented modeling and architecture**. [En línea]. Consultado el día 27 de julio de 2015 de la World Wide Web: <http://www.ibm.com/developerworks/library/ws-soa-design1/>.
- Bernstein, P. (1996). **Middleware: A Model for Distributed System Services**. Washington State University
- Bologna, J. y Walsh, A. M. (1997). **The Accountant's Handbook of Information Technology**. Editorial John Wiley and Sons.
- Carrascoso, Y. y Chaviano, V. (2008). **Propuesta de Arquitectura Orientada a Servicios para el Módulo de Inventario del ERP Cubano**. Investigación Concluida. Universidad de las Ciencias Informáticas. Ciudad de la Habana, Cuba.
- Damián Pérez Valdés (2007). **Maestros del Web editorial**. [En línea]. Consultado el día 31 de julio de 2015 de la World Wide Web: <http://www.maestrosdelweb.com/que-son-las-bases-de-datos/>
- Fowler, Martin, y otros. 2002. **Patterns of Enterprise Application Architecture**. s.l.: Addison Wesley, 2002. 0-321-12742-0.
- Frantz, R. (2008). **Integración de Aplicaciones. Un lenguaje específico de dominio para el diseño de solución de integraciones**. Editorial: The Distributed Group

García, A. (2007). **Arquitecturas Empresariales. Orientación a Servicios (SOA) y Gestión de Procesos de Negocio (BPM)**. Recuperado el 4 de agosto de 2015 de la World Wide Web: <http://www.scribd.com/doc/12732986/Arquitecturas-Empresariales-Orientacion-a-Servicios-SOA-y-Gestion-de-Procesos-de-Negocio-BPM>

Hernández, K. (2008). **Desarrollo tecnológico y su vinculación con la educación**. México: Instituto de Ciencias, Humanidades y Tecnología de Guanajuato.

Laudon, K. y Laudon, J. (1996). **Administración de los Sistemas de Información**. Tercera Edición. Prentice Hall Hispanoamericana, S.A. Naucalpan de Juárez.

Mijares, H. y García, L. (Julio del 2007). **Normas para la elaboración y presentación de los anteproyectos, proyectos y trabajos de grado**. [En línea]. Consultado el día 01 de julio de 2015 de la World Wide Web: [http://www.ujap.edu.ve/Universitas/html/Pasantias/NORMAS\\_trabajodegrado\\_Julio\\_2007.pdf](http://www.ujap.edu.ve/Universitas/html/Pasantias/NORMAS_trabajodegrado_Julio_2007.pdf).

Papazoglou, M. y Den Heuvel W. Van (2005). **Services oriented architecture approaches, technologies and research issues**. The VLDB Journal, 16(3), 389-415.

Papazoglou, M. y Den Heuvel W. Van (2006). **Service-oriented design and development methodology**. Int. J. of Web Engineering and Technology (IJWET), 2(4), 412-442.

Pérez, J. **Enfoque orientado a servicios para la reutilización de procesos y herramientas de modelización**. Tesis de doctorado. Universidadde Grenoble. Grenoble, Francia 2010

Pressman, R. (2002). **Ingeniería de Software. Un enfoque practico**. McGraw.Hill/Interamericana de España.,

Reynoso, C. (2006). **MSDN Introducción a la Arquitectura de Software**. Recuperado el 5 de agosto de 2015 de World Wide Web: [http://www.microsoft.com/spanish/msdn/arquitectura/roadmap\\_arq/intro.aspx](http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/intro.aspx)

Reynoso, C. y Kicillof, N. (2004). **MSDN Estilos y Patrones en la Estrategia de Arquitectura de Microsoft**. [En línea] Recuperado el 5 de agosto de 2015 de World Wide Web: [http://www.microsoft.co.ke/spanish/msdn/arquitectura/roadmap\\_arq/style.aspx](http://www.microsoft.co.ke/spanish/msdn/arquitectura/roadmap_arq/style.aspx)

Sabino, C. (2004). **Como Hacer una Tesis**. Quinta Edición. Venezuela: Editorial Panapo.

Sabino, C. (2010), **Diccionario de la lengua española**. Editorial Alfa, Caracas Venezuela.

Shulte, R. y Natis, Y. (1996). **"Service Oriented" Architectures, Part 1**. [Versión electrónica]. Gartner Inc.

UNACE (1996). **United Nations Economic Commission for Europe**. [En línea]. Consultado el día 30 de julio de 2015 de la World Wide Web: <http://www.unece.org/tradewelcome/un-centre-for-trade-facilitation-and-e-business-uncedfact/outputs/standards/unedifact/tradeedifactrules/part-2-uniform-rules-of-conduct-for-interchange-of-trade-data-by-teletransmission-uncid/part-2-uncid-chapter-1-introductory-note.html>.

TIBCO Software Inc. (2008). **The Power of Now, El papel de un bus de servicios empresariales (ESB) en una SOA**. [En línea] Recuperado el 6 de agosto de 2015 de la World Wide Web: <http://www.bpmcenter.cl/index.php/component/content/article/3-noticias/59-tibco-amplia-la-plataforma-activematrix-para-simplificar-la-soa-todavia-mas.html>

