



UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS

COMPUTACIONALES

**TRABAJO DE GRADO, PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SISTEMAS.**

TEMA:

**“COMPARAR DOS ESB J2EE; MULEESB VS GLASSFISHESB /
OPENESB, CON EL PROTOTIPO: DE FACTURACIÓN
ELECTRÓNICA.”**

AUTOR: FABRICIO XAVIER HUERA VINUEZA

DIRECTOR: ING. MAURICIO REA

IBARRA – ECUADOR

2016



Artículo Científico

COMPARAR DOS ESB J2EE; MULEESB VS GLASSFISHESB / OPENESB, CON EL PROTOTIPO: DE FACTURACIÓN ELECTRÓNICA.

FABRICIO HUERA*

MAURICIO REA**

* Egresado de la Carrera de Ingeniería en sistemas computacionales de la Facultad de Ingeniería en Ciencias Aplicadas de la Universidad Técnica Del Norte (Ibarra – Ecuador) - xavys10@yahoo.com

** Profesor Investigador de la Facultad de Ingeniería en Ciencias Aplicadas de la Universidad Técnica Del Norte (Ibarra - Ecuador)

Resumen. Este trabajo de tesis realiza una comparación de los buses de servicios empresariales, con el objetivo de dar a conocer sobre la arquitectura orientada a servicios y los beneficios que se tiene al utilizar dicha arquitectura, el estudio comparativo se lo realizó con el prototipo de facturación electrónica.

Palabras claves:

Arquitectura orientada a servicios, buses de servicios empresariales, mule ESB, openESB.

Summary. This thesis work performs a comparison of bus service and business services, with the aim to raise awareness about the services-oriented architecture and the benefits that you have when using this architecture, the comparative study was it conducted with the prototype of electronic invoicing.

Key Words:

Service Oriented Architecture, mule ESB, openESB, Enterprise Service Bus

Introducción

En la empresa MantizSoft se ha observado que los procesos de facturación se realizan mediante la utilización de documentos físicos en papel, la información esta almacenada en archiveros, esto crea un gran problema porque al momento de solicitar un reporte de las facturas, el proceso es muy lento por la minuciosa búsqueda de documentos en los archivos físicos que posee la empresa.

La inadecuada estructuración y flexibilidad en el intercambio de información entre distintos módulos, los cuales son el de facturación así como el de inventarios, como también la falta total de integración entre la tecnología y los procesos.

En algún punto en el crecimiento de las organizaciones o las empresas, con relación a las soluciones informáticas que proveen sistemas que soportan los procesos de negocios, y cuando el tamaño creciente de la empresa y la complejidad que estas tienen; hay que dar el salto al concepto de arquitectura empresarial.

Debido a estas necesidades se ha optado por implementar la factura electrónica el mismo que será desarrollado con herramientas de software libre, utilizando SOA, lenguaje de programación java.

El estudio propone variedad de ventajas económicas y de tecnología para la empresa porque la arquitectura orientada a servicios (SOA) nos brinda un sin fin de posibilidades para mejorar el funcionamiento de la empresa.

El uso de los bus de servicios empresariales (ESB) es una solución que facilita la integración distribuida, que está basada en mensajes y en estándares.

La función que realiza un ESB es facilitar una comunicación confiable entre los diferentes recursos tecnológicos como plataformas, aplicaciones y servicios, que están distribuidos en diferentes sistemas de toda la empresa. La fortaleza de una solución de ESB es la posibilidad de la comunicación entre sistemas de cualquier protocolo, se encarga de traducir de un lenguaje a otros, el lenguaje que se utiliza se normaliza, empleando lenguaje XML. (Borja, Florez, & Torres Leon, 2011)

Las empresas para ser competitivas requieren arquitecturas, procesos comerciales de TI.

Soporten un alto nivel de desarrollo y un alto nivel de competencia. Las empresas están inmersas en un mundo que exige un desarrollo empresarial y tecnológico, apresurado y competitivo, lo que obliga a que sus procesos deban invariablemente optimizar y mejorar.

Capítulo II

Cuando se introdujeron las tecnologías de información (TI) en todas las industrias, las empresas han construido y desarrollado sus sistemas de información en base a las necesidades que tienen las empresas de negocios, conformando infraestructuras de los datos que se serian transformados en almacenes de información, pobremente cohesionados o mayormente aislados.

Esta situación llegó a un punto crítico cuando las empresas empezaron a enfrentar procesos complejos de fusión o expansión, obligando a ordenar la información y transferirla de una manera más consolidada y eficiente.

Cada vez hay aplicaciones más complejas, con mucho menos tiempo de desarrollo y un presupuesto limitado, y en la mayoría de los casos se requiere re implementar funcionalidades ya existentes.

Tratar de reutilizar todas las funcionalidades resulta una labor complicada de debido a que no fueron diseñadas para poder integrarse entre sí, o bien fueron desarrolladas sobre tecnologías o plataformas desacordes entre sí. El ESB trata a las aplicaciones como servicios, con independencia de cómo sea conexión al bus, permitiendo que las empresas migren gradualmente.

Capítulo III **Mule ESB**

Mule ESB es una de las primeros y exitosos ESB de código abierto. Tienen unas funcionalidades que son similares los demás ESB de código abierto. Mule ESB incluye una instalación sencilla e intuitiva, tiene las herramientas basadas en Eclipse. Por lo general, los ESB de código abierto son soluciones muy ligeras y extensibles. Aparte de la versión de código abierto, existe una versión comercial que está disponible en su página web.

Mule ESB es un bus de servicios basado en Java es ligero y tiene una plataforma de integración que ayuda a los desarrolladores a conectar las aplicaciones de forma rápida y fácil, lo que les permite utilizar el intercambiar de datos.

Mule ESB es de una fácil integración en los sistemas existentes, sin tener en cuenta las variadas tecnologías que pueden utilizar las aplicación, tales como:

- JMS
- Servicios Web
- JDBC
- HTTP.

El bus actúa como mediador entre los diferentes servicios del negocio y se encarga de poder conectarlos entre sí mediante los mensajes de comunicación de estándares. La gran ventaja de un bus es que tenemos que definir la manera de poder integrar servicio con el bus seleccionado, por lo que nos aislamos de la configuración del resto de servicios.

Es encargado de controlar al mensaje recibido para poder direccionarlo a un servicio.

Capítulo IV **OpenESB.**

OpenESB es una herramienta de bus de servicios empresariales, que trabaja con eficiencia en la creación de aplicaciones de integración y hace que SOA sea más fácil. Inicialmente fue diseñado y desarrollado por Sun Microsystems y SeeBeyond, OpenESB se ha mejorado y mantenido por una gran comunidad que colaboran con el proyecto.

OpenESB ofrece un conjunto completo de herramientas que ayudan a diseñar, desarrollar, probar y desplegar aplicaciones de integración y aplicaciones orientadas a servicios. Basándose en JBI (Java Business Integration).

OpenESB propone un proceso de desarrollo único que promueve la migración a un servicio real de desarrollo orientado. Hoy en día, muchas empresas (Telco, Finanzas, Logística, Banco) están utilizando OpenESB para proyectos departamentales y de integración táctica.

Ellos encuentran en este bus de servicios empresariales el mejor equilibrio entre el desarrollo ergonómico, fiabilidad, potencia, escalabilidad y de código abierto. Asimismo, se encuentran en la comunidad, una pila completa de servicios de apoyo para los proyectos

OpenESB: Permite integrar rápidamente aplicaciones empresariales y web services como las aplicaciones compuestas y débilmente acopladas. Esto permite componer y recomponer de una manera fluida y rápida las aplicaciones compuestas, con las ventajas de una Arquitectura Orientada a Servicios.

Capítulo V

Estudio comparativo de los buses de servicio empresariales.

En este capítulo se centrará en establecer parámetros básicos de comparación que permitirán la elección de un bus de servicios empresariales adecuado que permita la futura implementación de una aplicación con la arquitectura SOA, y que utilice la plataforma JavaEE.

Estos parámetros se van a establecer en base al estudio de los temas como son:

Definición de la Arquitectura Orientada a Servicios y las características que son necesarias para la utilización de este paradigma en la implementación de una solución del tipo empresarial.

Definición de la Arquitectura JavaEE y los buses de servicios empresariales existentes en el medio que implementan esta tipo de arquitectura referencial posibilitando la elaboración de aplicaciones SOA.

Hay una gran variedad de Buses de Servicios Empresariales que implementan la arquitectura JavaEE, y tomando como referencia la utilización y popularidad de los ESB se puede distinguir a dos ESB que se utilizan mayoritariamente en el mundo empresarial. Los dos ESB tienen sus versiones libres que son mantenidas por Comunidades de Desarrolladores en Internet, y adicionalmente existen las versiones de paga, que se diferencian de las anteriores porque implementan mecanismos mucho más robustos en lo que tiene que ver a Seguridad.

Los ESB expuestos entonces son:

- Mule ESB v 3.5
- OpenESB v 2.3

Todos estos se someterán al estudio previamente configurado en un equipo con el Sistema Operativo Ubuntu.

Establecimiento de los parámetros para la elección de un ESB para SOA

Las características que se van a exponer aquí en este estudio son estándares de la industria y que fueron propuestos por varios autores y expertos en el desarrollo de aplicaciones SOA para la plataforma JavaEE:

Desarrollo

- Herramientas basadas en Eclipse
- Herramientas basadas en el IDE NetBEANS.
- Creación de Drag-and-Drop
- Protocolos de transporte JDBC, JMS, HTTP
- Arrastrar y soltar Servicios Web y REST
- Experiencia

Capacidades de integración

- Creación y habilitación de los Servicios
- Soporte de patrones de integración Enterprise
- Nube de integración y oferta de plataforma (iPaaS)
- Enrutamiento de mensajes basado en contenido
- Mensajería
- Integración de datos
- Transformación de datos XML del mensaje (XSLT, XQuery...)
- Orquestación de servicios
- Seguridad e identidad de servicios

Despliegue

- Flexibilidad y despliegue del contenedor
- Soporte OSGI
- SOAP (WSDL)
- WS-s*, a través del servicio de token de seguridad (STS)
- Alta Disponibilidad

Administración y Gestión

- Monitoreo JMX
- Supervisión de actividades de servicios
- Soporte y Documentación
- Comunidad
- Documentación
- Ejemplos con código

Para valorar se expone la tabla donde muestra los parámetros los valores que se han puesto a cada parámetro de elección del ESB.

| Detalle | Valor |
|---------|-------|
| Sí | 3 |
| Parcial | 2 |
| No | 1 |

De la tabla anterior si todos los buses de servicios empresariales cumplieran con todos los parámetros de evaluación se lograría la suma de 78 lo que representaría el 100% de efectividad del bus de servicios;

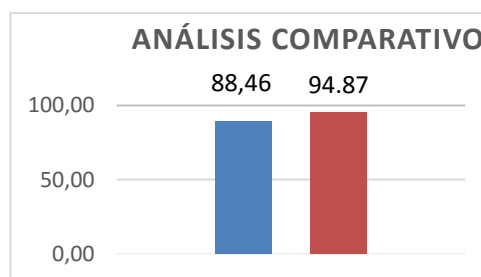
De tal manera para la representar la valoración de cada bus de servicios empresarial se empleara una regla de tres simple, la cual es de la siguiente manera:

$$78 \quad 100\%$$

$$69 \quad X \quad = (100*69)/78 = 88.46\%$$

Obteniendo los siguientes resultados.

| Bus de Servicios Empresarial | Mule ESB | OpenESB |
|------------------------------|----------|---------|
| Porcentaje | 88.46% | 94.87% |



Resultados obtenidos.

OpenESB es el bus de servicios Empresariales que obtiene el mejor resultado con un 94.87%, debido a sus características, facilidad de acoplamiento, integración con herramientas, Con un 89.42% se encuentra Mule ESB y se encuentra ubicado en la segunda posición.

| ESB | Estudio comparativo | Promedio Rendimiento |
|----------|---------------------|----------------------|
| Mule ESB | 88,46% | 2,854 |
| OpenESB | 94.87% | 3,522 |

Se demuestra entonces que ya una vez realizado el estudio comparativo, en el cual el ESB OpenESB obtiene el mayor porcentaje de cumplimiento de todos los estándares, y posteriormente fue sometido a las pruebas de rendimiento, OpenESB obtiene el mejor rendimiento.

Entonces se llegó a comprobar de una manera favorable el problema planteado, ya que OpenESB es el ESB adecuado que asegurará rendimiento y brindará un menor tiempo de respuesta y un acceso rápido a la información.

Capítulo VI

Diseño y aplicación del prototipo utilizando metodología XP

Al evaluar las diferentes alternativas que existen en lenguajes de programación y/o plataformas, la aplicación se desarrolló con el lenguaje de programación Java, dado las características y la sencillez que este lenguaje provee para el trabajo y la gestión en la bases de datos, el cual es el núcleo central de la aplicación.

Fase 1: Exploración

En esta fase el cliente plantea las historias de usuario las cuales son de su interés para la primera entrega del producto.

Historias de usuario

Las historias de usuario son las especificaciones y los requisitos del software, donde se describen brevemente las características que el sistema de facturación debe tener desde la perspectiva y necesidades del cliente.

Fase 2: Planificación

Para poder elaborar la planificación, es necesario poder identificar las historias de usuarios y establecer la prioridad que cada historia, realizando y analizando una estimación del esfuerzo necesario.

Fase 3: Plan de interacciones

En la fase de interacciones se presentan y se describen las historias que se llevaron a cabo en la iteración final, esta fase incluye las pruebas funcionales, la planificación del proyecto y todas las incidencias que ha tenido el proyecto. Finalmente se describe toda la evolución que se ha producido en el equipo que desarrollo el proyecto.

Fase 4: Producción

En esta fase se observa la finalización, las pruebas adicionales y revisar el rendimiento del sistema, para que este sistema pueda ser trasladado al entorno del cliente y sea implementado.

Seguimiento iteraciones.

Para el seguimiento es muy importante la comunicación entre el cliente y los desarrolladores, la finalidad es enfocarse en encontrar y establecer problemas y posibles soluciones de las tareas de desarrollo de la aplicación.

Reporte por iteraciones.

El objetivo principal es controlar las tareas asignadas a cada iteración, aquí se podrá visualizar todo el desarrollo del proyecto.

Fase 5: Cierre del proyecto

Dado que el cliente ya no tiene más historias para poder ser incluidas en el sistema, es decir que se ha satisfecho las expectativas, ahora es necesario poner a prueba el rendimiento, confiabilidad y eficiencia de nuestro sistema de facturación. Para el cual se genera la documentación final que es la especificación de las pruebas del sistema.

Especificación de pruebas:

Las pruebas son una parte importante del sistema, debido a que las pruebas deben ser realizadas la mayor cantidad de veces lo que permitirá corregir errores y poder obtener resultados esperados.

Conclusiones

Al comparar los ESB se observó que cada uno tiene su particularidad y distinta manera de funcionamiento, el estudio comparativo fue para demostrar cual ESB es el más apto para el sistema que se desarrolló de facturación electrónica, resultando como mejor opción el OpenESB el cual se adaptó de la mejor manera al prototipo de Facturación Electrónica funcionando con un alto rendimiento y confiabilidad y es el que adapta a las necesidades para este sistema de facturación electrónica, ya que en la simulación de este ESB observamos que proporciona simplicidad, eficiencia, durabilidad a largo plazo, ofreciendo herramientas para diseño, desarrollo, pruebas y desplegar aplicaciones.

La utilización de Mule ESB con su plataforma de integración de peso ligero ha permitido conectar el sistema de facturación electrónica de una manera rápida y confiable que permite conectar otros sistemas y aplicaciones el cual puede manejar toda la comunicación entre los sistemas, lo que permite seguir y controlar todo lo que sucede.

OpenESB resulto un ESB confiable, rápido y fácil al momento de integrar e implementar los procesos de negocio del sistema de facturación electrónica, OpenESB cumplió con todos los requisitos planteados en el prototipo.

La facturación electrónica en las empresas brinda beneficios ambientales, económicos y administrativos, debido a que ya no se utiliza papel y los costos se reducen.

Recomendaciones

Se recomienda un estudio más exhaustivo de cada uno de los ESB para poder explotar al máximo las capacidades que tienen y no solo sea utilizado en un sistema de facturación electrónica sino en los sistemas de las empresas para un mejor funcionamiento y se recomienda para futuros trabajos de investigación los diferentes ESB que existen en la actualidad ya que nos brindan una variedad de beneficios para las empresas.

En los escenarios fuertemente acoplados se recomienda el uso de un ESB que desacople a todas las aplicaciones, haciendo posible que el uso sea fácil y centralizado.

Se recomienda la constante revisión de leyes y reglamentos de la emisión de comprobantes electrónicos lo que va a permitir realizar una retroalimentación al sistema de facturación electrónica para que pueda cumplir las Normas del SRI.

Se recomienda usar este prototipo desarrollado para ayudar, apoyar a los sistemas de facturación, estableciendo alternativas para colaborar con el medio ambiente reduciendo el uso de papel mediante la utilización de facturación electrónica desarrollada con los buses de servicio que se dio a conocer.

Bibliografía

Borja, E., Florez, M. E., & Torres Leon, P. A. (1 de Febrero de 2011).

ESPECIALIZACIÓN EN PROCESOS PARA EL DESARROLLO DE SOFTWARE. Obtenido de:

http://bibliotecadigital.usbcali.edu.co/jspui/bitstream/10819/351/1/Comparativo_%20Software_Eficacia_Borja_2011.pdf

- Cenatic. (10 de Enero de 2014). Centro Nacional de Referencia de Aplicación de las TIC. Obtenido de Cenatic: <http://www.cenatic.es/laecsp/page21/page22/page22.html>
- Centro de Alto Rendimiento de Accenture. (17 de 09 de 2012). Accenture. Obtenido de <http://www.accenture.com/es-es/Pages/insight-arquitectura-orientada-servicios.aspx>
- Erl, T. (2012). SOA Principles of Service Design. Boston: Pearson Education. Inc.
- Exposito, E., & Diop, C. (11 de Septiembre de 2014). Smart SOA Platforms in Cloud Computing Architectures. California: FOCUS Series. Obtenido de: <http://fergunet.blogspot.com/2007/09/open-esb-vs-servicemix.html>
- Gras, J. C. (20 de Marzo de 2012). Horizontes BPM. Obtenido de modelación, arquitectura... paradigma: http://horizontesbpm.blog.com/?p=111#_ftn2
- Hurwitz, J., Bloor, R., Baroudi, C., & Kaufman, M. (2007). Service Oriented Architecture For Dummies. Indianápolis, Indiana: Wiley Publishing, Inc.
- IDG Communications. (01 de Febrero de 2012). Network World. Obtenido de <http://www.networkworld.es/archive/integracion-de-aplicaciones-en-esb>
- Ingeniería de Software U. Union Bolivariana. (12 de Enero de 2014). Ingeniería de Software. Obtenido de http://ingenieriadesoftware.mex.tl/52753_XP---Extreme-Programing.html
- José, C., Leszek, M., & Joaquim, F. (31 de Mayo de 2012). Expertise Solutions. Wroclaw: Springer. Obtenido de <http://www.expertise-solutions.com.pe/frameworks.html>
- Kendall, K., & Kendall, J. (2011). Análisis y Diseño de Sistemas. Naucalpan de Juárez: Pearson Educación.
- López, W. N. (2009). Mejores Prácticas SOA, Estándares y Gobernabilidad. San Juan: Caribbean White Paper.
- López-Vázquez, M. A.-P. (2012). Fundamentos de las Infraestructuras de Datos Espaciales (IDE). BibliotecaOnline SL. Obtenido de ISBN: 8493919683, 9788493919689
- Borja, E., Florez, M. E., & Torres Leon, P. A. (1 de Febrero de 2011). ESPECIALIZACIÓN EN PROCESOS PARA EL DESARROLLO DE SOFTWARE. Obtenido de http://bibliotecadigital.usbcali.edu.co/jspui/bitstream/10819/351/1/Comparativo_%20Software_Eficacia_Borja_2011.pdf
- Cenatic. (10 de Enero de 2014). Centro Nacional de Referencia de Aplicación de las TIC. Obtenido de Cenatic: <http://www.cenatic.es/laecsp/page21/page22/page22.html>
- Centro de Alto Rendimiento de Accenture. (17 de 09 de 2012). Accenture. Obtenido de <http://www.accenture.com/es-es/Pages/insight-arquitectura-orientada-servicios.aspx>
- Erl, T. (2012). SOA Principles of Service Design. Boston: Pearson Education. Inc.
- Exposito, E., & Diop, C. (11 de Septiembre de 2014). Smart SOA Platforms in Cloud Computing Architectures. California: FOCUS Series. Obtenido de <http://fergunet.blogspot.com/2007/09/open-esb-vs-servicemix.html>.



TECHNICAL UNIVERSITY OF NORTH

FACULTY OF ENGINEERING IN APPLIED SCIENCE

ENGINEERING DEGREE IN COMPUTER SYSTEMS

**GRADE WORK PRIOR TO OBTAINING THE TITLE OF SYSTEMS
ENGINEER**

THEME:

**"COMPARE TWO ESB J2EE; MULEESB VS GLASSFISHESB /
OPENESB, WITH THE PROTOTYPE: ELECTRONIC INVOICING".**

AUTHOR: FABRICIO XAVIER HUERA VINUEZA

DIRECTOR: ING. MAURICIO REA

IBARRA – ECUADOR

2016

Scientific Article

COMPARE TWO ESB J2EE; MULEESB VS GLASSFISHESB / OPENESB, WITH THE PROTOTYPE: ELECTRONIC INVOICING.

FABRICIO HUERA *

MAURICIO REA**

* Graduate of Systems Engineering, Faculty of Engineering Applied Science from the Technical University of the North (Ibarra -Ecuador) –xavys10@yahoo.com

** Researcher Professor of the Faculty of Engineering in Applied Science from the Technical University of the North (Ibarra)

Summary. This thesis makes a comparison of the enterprise service bus, with the aim to raise awareness about service-oriented and the benefits have to use this architecture, architecture comparative study it conducted with the prototype of electronic invoicing.

Keywords:

Service oriented architecture, enterprise service bus, mule ESB, OpenESB.

Introduction

In the company MantizSoft has been observed that billing processes are performed using physical paper documents, the information is stored in file cabinets, and this creates a big problem because when requesting a report of invoices, the process is very slow by the thorough search of documents in physical files owned by the company.

Inadequate structuring and flexibility in the exchange of information between different modules, which are billing as well as inventory, as well as the total.

At some point in the growth of organizations or businesses regarding IT solutions that provide systems that support business processes, and when the growing size of the company and the complexity that these are; we must make the leap to the concept of enterprise architecture.

Because of these needs it has chosen to implement the electronic invoice which will be developed with free software tools, using SOA Java programming language.

The study proposes variety of economic and technology for business advantages for service-oriented architecture (SOA) gives us endless possibilities for improving the functioning of the company.

Using the enterprise service bus (ESB) is a distributed solution that facilitates the integration, which is based on messages and standards.

The function that performs a USB is to provide reliable communication between different technological resources and platforms, applications and services, which are distributed in different systems across the enterprise. The strength of ESB solution is the possibility of communication between systems of any protocol, is responsible for translating from one language to another, the language used is normalized, using XML language. (Borja, Florez, & Torres Leon, 2011).

Companies require to be competitive architectures, business processes.

Support a high level of development and a high level of competition. Companies are immersed in a world that demands business and technology, quick and competitive development, which requires that its processes must invariably optimize and improve.

Chapter II

When information technology (IT) in all industries were introduced, companies have built and developed its information systems based on the needs that companies have business, forming infrastructure of the data would be transformed into information stores poorly cohesive or mostly isolated.

This situation came to a head when companies began to address complex mergers or expansion, forcing sort information and transfer it in a more consolidated and efficient manner.

More and more complex applications with much less development time and a limited budget, and in most cases re required to implement existing functionality.

Try to use all the functions is a complicated work because they were not designed to be integrated with each other, or were developed on technologies or platforms discordant with each other. The ESB treats applications as services, regardless of how it is connected to the bus, allowing companies to migrate gradually.

Chapter III

Mule ESB

Mule ESB is one of the first successful open source ESB. They have some features that are similar the other open source ESB. Mule ESB includes a simple and intuitive installation, you have Eclipse-based tools. Usually, the open source ESB are very light and extensible solutions. Apart from the open source version, there is a commercial version which is available on its website.

Mule ESB is a service bus based on Java is lightweight and has an integration platform that helps developers connect applications quickly and easily, allowing them to use the data exchange.

ESB Mule is easy integration into existing systems, regardless of the various technologies that may use the application, such as:

- JMS
- Web Services
- JDBC
- HTTP.

The bus acts as a mediator between the different services of the business and is responsible to connect with each other through messages communication standards. It is responsible for monitoring the received message to a service address.

Chapter IV

Open ESB.

OpenESB is a tool enterprise service bus, which works effectively in building applications integration and SOA makes it easier. It was initially designed and developed by Sun Microsystems and SeeBeyond, OpenESB has been improved and maintained by a large community that collaborate with the project.

OpenESB offers a complete set of tools to help design, develop, test and deploy applications integration and service-oriented applications. Based on JBI (Java Business Integration).

OpenESB proposes a unique development process that promotes migration to a real service-oriented development. Today, many companies (Telco, Finance, Logistics, Bank) are using OpenESB for departmental and tactical integration projects.

They find in this enterprise service bus the best balance between ergonomic development, reliability, power, scalability and open source. Also they found in the community, a whole stack of support services for projects

OpenESB: Quickly integrate business applications and web services as composite and loosely coupled applications. This allows you to compose and recompose in a fluid and fast composite applications with the advantages of a service-oriented architecture.

Chapter V

Comparative study of enterprise service buses.

This chapter will focus on establishing basic parameters of comparison that allow the selection of a suitable enterprise service bus that enables future implementation of an application with SOA, and use the JavaEE platform.

These parameters will be established based on the study of topics such as:

Definition of Service-Oriented Architecture and features that are necessary for the use of this paradigm in implementing the enterprise-class solution.

Architecture definition JavaEE and existing enterprise services buses in the middle that implement this type of referential architecture enabling development of SOA applications.

There are a variety of Enterprise Service Buses that implement the JavaEE architecture, and with reference to the use and popularity of ESB can distinguish two ESB that are used mainly in the business world. Both ESB have their free versions that are maintained by communities of developers on the Internet, and additionally there are paid versions, which differ from the above because they implement more robust mechanisms in what you have to do to Security.

The ESB then are exposed:

- Mule ESB v 3.5
- OpenESB v 2.3

Setting the parameters for choosing an ESB for SOA

The features to be present here in this study are industry standards that were proposed by various authors and experts in the development of SOA applications for JavaEE platform:

- Development
- Eclipse-based tools
- Tools based on the NetBeans IDE.
- Creating Drag-and-Drop
- Transport protocols JDBC, JMS, HTTP
- Drag and Drop Web Services and REST
- Experience

Integration capabilities

- Creating and Enabling Services
- Support Enterprise Integration Patterns
- Cloud integration platform offering (iPaaS)
- Message routing based on content
- Messaging
- Data Integration
- Transformation of XML data message (XSLT, XQuery ...)
- Service Orchestration
- Security and identity services

Deployment

- Flexibility and deployment of the container
 - Support OSGI
- SOAP (WSDL)
- WS-s * through the security token service (STS)
 - High availability

Administration and management

- Monitoring JMX
- Monitoring service activities
- Support and Documentation

- Community
- Documentation
- Coded Examples

To assess the table shows the parameters where the values that have made each parameter choice of ESB is exposed.

| Detail | Value |
|---------|-------|
| Yes | 3 |
| Partial | 2 |
| No | 1 |

From the above table if all enterprise service buses comply with all evaluation parameters the sum of 104 which would represent 100% effective bus services would be achieved;

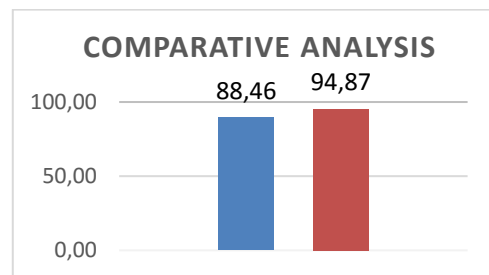
So to represent the valuation of each enterprise service bus one simple rule of three is employed, which is as follows:

$$78 \quad 100\%$$

$$69 \quad X = (100*69)/78 = 88.46\%$$

The following results.

| Enterprise Service Bus | Mule ESB | OpenESB |
|------------------------|----------|---------|
| Percentage | 88.46% | 94.87% |



Results obtained.

OpenESB is the enterprise service bus you get the best result with 94.87%, due to its characteristics, ease of coupling, integration tools, With a 88.46% Mule ESB is and is located in the second position.

| ESB | Stud Comparative study | Average yield |
|----------|------------------------|---------------|
| Mule ESB | 88,46% | 2,854 |
| OpenESB | 94,87% | 3,522 |

And then shows that once the comparative study, in which the ESB OpenESB gets the highest percentage of compliance with all standards, and was subsequently subjected to performance tests, OpenESB get the best performance.

Then he came to check in a favorable manner the problem posed since OpenESB is right ESB will ensure performance and provide a lower response time and quick access to information.

Chapter VI

Design and implementation of the prototype using XP methodology

When evaluating the different alternatives that exist in programming languages and / or platforms, the application was developed with the Java programming language, given the features and simplicity that this language provides for work and management in databases, the is the core of the application.

Phase 1: Exploration

In this phase, the client presents user stories which are of interest for the first delivery.

User Stories

User stories are the specifications and requirements of the software, which briefly describes the features that the billing system should be from the perspective and needs of the client.

Phase 2: Planning

In order to prepare the planning, it is necessary to identify the user stories and prioritize each story, carrying out and analyzing an estimate of the effort required.

Phase 3: Plan interactions

In the present phase interactions and stories that took place in the final iteration, this phase includes functional testing, project planning and all incidents that had the project are described. Finally the whole evolution that has occurred in the team that developed the project described.
essary knowledge.

Phase 4: Production

At this stage completion is observed, additional testing and review system performance, so that this system can be transferred to the customer environment and is implemented.

Monitoring iterations.

For monitoring is very important communication between the client and developers, the purpose is to find and focus on problems and possible solutions of the tasks of application development.

Report by iterations.

The main objective is to control the tasks assigned to each iteration, here you can view the entire project development.

Phase 5: Project Closure

Since the client has no more stories to be included in the system, that has met expectations.

It is now necessary to test the performance, reliability and efficiency of our billing system. To which the final documentation is the specification of system testing is generated.

Test specification:

Testing is an important part of the system, because the tests should be performed as many times allowing correct errors and to obtain expected results.

Conclusions

By comparing the ESB is observed that each has its particularity and different way of operation, the comparative study was to demonstrate that ESB is the most suitable for the system that developed electronic billing, resulting as the best option OpenESB which was adapted the best way to prototype electronic billing working with high performance and reliability and is the one that fits the needs for this electronic billing system, because in simulating this ESB note that provides long-term simplicity, efficiency, durability offering tools for design, development, test and deploy applications.

The use of Mule ESB with its integration platform lightweight has connected the electronic billing system in a quick and reliable way to connect other systems and applications which can handle all communication between systems, enabling tracking and control everything that happens.

OpenESB turned a reliable, fast and easy ESB when integrating and implementing business processes electronic billing system, OpenESB met all the requirements set in the prototype.

Electronic billing companies provide environmental, economic and administrative benefits, because they no longer use paper and reducing costs.

Recommendations

a more comprehensive analysis of each of the ESB study is recommended to fully exploit the capabilities they have and not only be used in an electronic billing system but systems companies for better performance and is recommended for future work research the different ESB that exist today as we provide a variety of benefits for businesses.

In the strongly coupled scenarios using an ESB that decouples all applications it is recommended, enabling easy use and centralized.

The constant revision of laws and regulations for issuing electronic vouchers which will allow feedback to make electronic billing system that can meet the standards of SRI is recommended.

We recommend using this prototype developed to help support billing systems, establishing alternatives to collaborate with the environment by reducing the use of paper by using electronic invoicing developed with the bus service was unveiled.

Bibliography

- Borja, E., Florez, M. E., & Torres Leon, P. A. (1 de Febrero de 2011). ESPECIALIZACIÓN EN PROCESOS PARA EL DESARROLLO DE SOFTWARE. Obtenido de: http://bibliotecadigital.usbcali.edu.co/jspui/bitstream/10819/351/1/Comparativo_%20Software_Eficacia_Borja_2011.pdf

- Cenatic. (10 de Enero de 2014). Centro Nacional de Referencia de Aplicación de las TIC. Obtenido de Cenatic: <http://www.cenatic.es/laecsp/page21/page22/page22.html>
- Centro de Alto Rendimiento de Accenture. (17 de 09 de 2012). Accenture. Obtenido de <http://www.accenture.com/es-es/Pages/insight-arquitectura-orientada-servicios.aspx>
- Erl, T. (2012). SOA Principles of Service Design. Boston: Pearson Education. Inc.
- Exposito, E., & Diop, C. (11 de Septiembre de 2014). Smart SOA Platforms in Cloud Computing Architectures. California: FOCUS Series. Obtenido de: <http://fergunet.blogspot.com/2007/09/open-esb-vs-servicemix.html>
- Gras, J. C. (20 de Marzo de 2012). Horizontes BPM. Obtenido de modelación, arquitectura... paradigma: http://horizontesbpm.blog.com/?p=111#_ftn2
- Hurwitz, J., Bloor, R., Baroudi, C., & Kaufman, M. (2007). Service Oriented Architecture For Dummies. Indianápolis, Indiana: Wiley Publishing, Inc.
- IDG Communications. (01 de Febrero de 2012). Network World. Obtenido de <http://www.networkworld.es/archive/integracion-de-aplicaciones-en-esb>
- Ingeniería de Software U. Union Bolivariana. (12 de Enero de 2014). Ingeniería de Software. Obtenido de <http://ingenieriadesoftware.mex.tl/52753-XP---Extreme-Programing.html>
- José, C., Leszek, M., & Joaquim, F. (31 de Mayo de 2012). Expertise Solutions. Wroclaw: Springer. Obtenido de <http://www.expertise-solutions.com.pe/frameworks.html>
- Kendall, K., & Kendall, J. (2011). Análisis y Diseño de Sistemas. Naucalpan de Juárez: Pearson Educación.
- López, W. N. (2009). Mejores Prácticas SOA, Estándares y Gobernabilidad. San Juan: Caribbean White Paper.
- López-Vázquez, M. A.-P. (2012). Fundamentos de las Infraestructuras de Datos Espaciales (IDE). BibliotecaOnline SL. Obtenido de ISBN: 8493919683, 9788493919689
- Borja, E., Florez, M. E., & Torres Leon, P. A. (1 de Febrero de 2011). ESPECIALIZACIÓN EN PROCESOS PARA EL DESARROLLO DE SOFTWARE. Obtenido de http://bibliotecadigital.usbcali.edu.co/jspui/bitstream/10819/351/1/Comparativo_%20Software_Eficacia_Borja_2011.pdf
- Cenatic. (10 de Enero de 2014). Centro Nacional de Referencia de Aplicación de las TIC. Obtenido de Cenatic: <http://www.cenatic.es/laecsp/page21/page22/page22.html>
- Centro de Alto Rendimiento de Accenture. (17 de 09 de 2012). Accenture. Obtenido de <http://www.accenture.com/es-es/Pages/insight-arquitectura-orientada-servicios.aspx>
- Erl, T. (2012). SOA Principles of Service Design. Boston: Pearson Education. Inc.
- Exposito, E., & Diop, C. (11 de Septiembre de 2014). Smart SOA Platforms in Cloud Computing Architectures. California: FOCUS Series. Obtenido de <http://fergunet.blogspot.com/2007/09/open-esb-vs-servicemix.html>.