

ARQUITECTURA ORIENTADA AL SERVICIO SERVICE ORIENTED ARCHITECTURE (SOA)

Ana Virginia Flores, Edwin Orlando Cholota, David Fabricio Marge

Universidad Técnica de Ambato, Ambato-Ecuador
aflores@uta.edu.ec

RESUMEN:

La creciente complejidad de los procesos de negocio y competitividad entre organizaciones, ha llevado a que se creen nuevas tecnologías ya no enfocadas en procesos sino en servicios, dando origen a la Arquitectura Orientada al Servicio (SOA) como una solución efectiva para las organizaciones que requieran integrar sus datos y lógica de negocio con las TI. El artículo presenta un estudio sobre SOA, revisando la información de libros y artículos técnicos donde se presentan los aspectos más importantes del tema. Inicialmente, se muestra un enfoque general sobre arquitectura de software, su evolución y los diferentes conceptos y definiciones, se explica la importancia del servicio dentro de la arquitectura. Posteriormente, se detallan las diferentes capas y componentes que la conforman, se expone la definición de servicio web sus características, protocolos y los estándares necesarios para su implementación. Por último, se dan a conocer las diferentes ventajas y desventajas de la misma.

Palabras clave: SOA, arquitectura, servicio web, estándares, SOAP.

ABSTRACT:

The increasing complexity of business processes and competitiveness in the organizations, has led to the creation new technologies focused in services not in processes, being SOA an effective solution for organizations that need to integrate their data with IT techniques. Our paper presents a description of Service-Oriented Architecture (SOA), after revising specific literature about this topic. Our initial approach shows a focus on this kind of architecture, its evolution and the different concepts and definitions of SOA, explains the importance of service into architecture. Next, we detail different layers and components that build up the architecture and we define Web service characteristics, protocols and standards required for its implementation. Finally, we expose the different advantages and disadvantages of SOA.

Keywords: SOA, architecture, web service, standard, SOAP.

Artículo Recibido: 1 de julio de 2013

Artículo Aceptado: 25 de septiembre de 2013

1. Introducción

La globalización y el mundo cambiante de los negocios, ha llevado a que las organizaciones adopten las Tecnologías de Información (TI) como una herramienta de apoyo para la administración de sus datos y como medio para alcanzar sus objetivos.

La influencia del internet en los negocios ha obligado a las organizaciones a reconstruir su infraestructura tecnológica y a crear nuevos sistemas de información altamente escalables e interoperables. Para que esto sea posible, los directivos de TI están abandonando gradualmente los sistemas software que han sido desarrollados de forma monolítica. Si bien estos sistemas han sido fundamentales para lograr el incremento económico de las organizaciones, hoy en día no solucionan completamente los requerimientos del usuario.

La construcción de aplicaciones a base de pequeños componentes como servicios de software y servicios web, permite a las organizaciones realizar cambios y adaptaciones en sus sistemas de manera ágil y flexible. Al reutilizar servicios se simplifica el desarrollo de aplicaciones, consiguiendo ahorro de costos, mantenimiento y tiempo.

Dentro del área de la construcción de aplicaciones software un concepto indispensable es la arquitectura de software cuyo propósito es crear productos de calidad y servir como guía para el desarrollo.

El construir sistemas de información sin tener en cuenta la arquitectura de software, puede limitar la calidad del producto final. De acuerdo a Bass, Clements y Kazman [1] la Arquitectura de Software se refiere a “las estructuras de un sistema, compuestas de elementos con propiedades visibles de forma externa y las relaciones que existen entre ellos.”

La ISO/IEC/IEEE 42010:2011 define la arquitectura de un sistema software como “la organización principal de un sistema que involucra sus componentes, las relaciones entre ellos, su entorno y los principios que guían su diseño y evolución.” [2]

Finalmente diremos que “Arquitectura es una disciplina que ayuda a alinear las TI con las estrategias de negocio de la organización.” [3]

En la Figura 1.1 se puede apreciar como la arquitectura de software se ha desarrollado desde sus inicios hasta la actualidad.

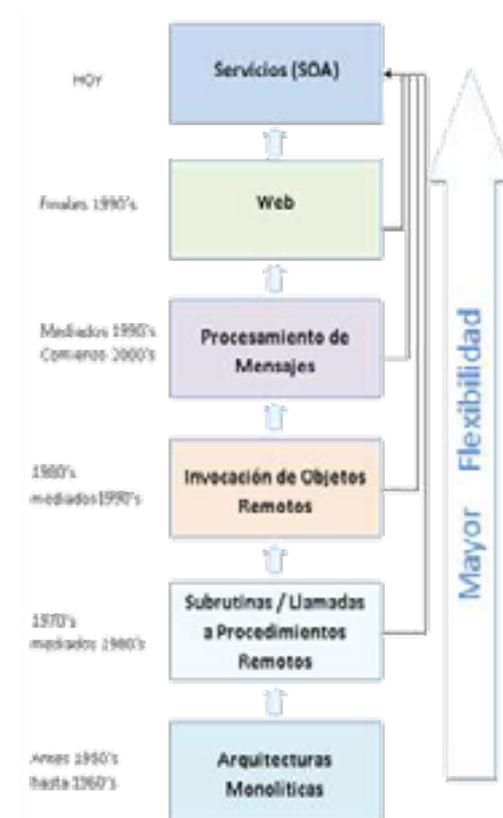


Figura 1. Desarrollo de la Arquitectura de Software. [14]

Como resultado de la evolución de las arquitecturas de software debemos destacar la arquitectura orientada al servicio denominada Service Oriented Architecture (SOA).

Un correcto enfoque de SOA en el diseño y construcción de sistemas software permite que las aplicaciones cuenten con atributos de calidad como: confiabilidad, mantenibilidad, flexibilidad, seguridad, interoperabilidad, escalabilidad, portabilidad, disponibilidad, reusabilidad entre otras, para poder lograr cumplir con dichos atributos es necesario construir aplicaciones a base de servicios.

2. SOA

El concepto de SOA nace a mediados de los años 80, cuando la computación distribuida y las llamadas a procedimientos remotos llegaron al mercado. Entre los años 1980

y 1990, SOA se limitó a proyectos de vanguardia cuyos arquitectos tuvieron la visión, la disciplina y el dinero para invertir en las fases iniciales de la aplicación en desarrollo, sabiendo que entre las recompensas de SOA estarían su escalabilidad, su agilidad y su reutilización. [4] A partir del año 2003 la arquitectura SOA se incorpora en las empresas [4], como una arquitectura que permita la interoperabilidad entre cada módulo o subsistema existente dentro de la organización, o para construir nuevas piezas de software con bajo acoplamiento, utilizando estándares para servicios web y que sean independientes de la tecnología empleada.

A continuación, veremos como es definida la Arquitectura SOA por varios autores:

Brown, define SOA como “un estilo de arquitectura que permite modularizar sistemas de información en servicios”. [5]

Lawler y Howell-Barber, definen SOA como “un framework para integrar los procesos de negocio y soportar la infraestructura tecnológica de la información de forma segura estandarizando componentes y servicios que pueden ser reusados y combinados para hacer frente a los cambios del negocio”. [6]

La Organization for Advancement of Structured Information Standards (OASIS), se refiere a SOA como un “Paradigma para organizar y utilizar capacidades distribuidas que pueden estar bajo el control de diferentes dominios. Provee una manera uniforme de ofrecer, descubrir, interactuar con ellos y sus capacidades de uso para producir el efecto deseado consistente en precondiciones y expectativas medibles.” [7]

IBM define SOA como “una forma de construir sistemas distribuidos que permiten ofrecer las funcionalidades de una aplicación como servicios tanto para aplicaciones de usuario final como para otros servicios”. Desde el punto de vista de IBM el objetivo principal de SOA es “alinearse al mundo de los negocios con el mundo de las TI de forma que los hace a la vez más efectivos”. [8]

En base a las definiciones antes comentadas se puede decir que: SOA es un modelo arquitectónico basado en estándares que sirven para diseñar y construir piezas de software

granulares, débilmente acoplados y altamente interoperables e independientes de la tecnología. Sus aplicaciones hacen uso o se basan en servicios para satisfacer las necesidades de las empresas integrando los procesos del negocio.

3. Conceptos Técnicos de la Arquitectura SOA

Según Nicolai M. Josuttis [9], una SOA se basa en tres principales conceptos técnicos: servicios, interoperabilidad a través de un bus de servicios empresariales, y bajo acoplamiento.

3.1 Servicios

Lawler y Howell-Barber, define los servicios en SOA como “módulos de negocio o funcionalidades técnicas con interfaces expuestas a la funcionalidad.” [6]

Un servicio es una pieza de funcionalidad de negocio autónomo, a los cuales se pueden acceder a través de la red. La funcionalidad puede ser simple (almacenar o recuperar los datos del cliente) o complejo (un proceso de negocios para un pedido). [9]

Los servicios se pueden utilizar de forma individual e independientemente de la aplicación de la que forman parte y de la plataforma informática sobre las que se ejecutan. Al disponer de servicios individuales, las empresas podrán compartirlos, reutilizarlos, y agruparlos formando así servicios compuestos que brinden una respuesta rápida, oportuna e inmediata a los requerimientos empresariales.

3.1.1 Principios de SOA basados en la orientación a servicios

Thomas Erl [13] define ocho principios divididos en dos categorías, de implementación y regulación.

Los principios de implementación definen las características físicas que debería tener en cuenta para diseñar servicios:

- Contrato de servicio estandarizado.-Por medio de un contrato los servicios expresan su funcionalidad.

Todo servicio desarrollado, debe proporcionar un contrato en el cual debe figurar el nombre del servicio, su forma de

acceso, las funcionales que ofrece y los datos de entrada de cada una de las funcionalidades y los datos de salida.

El consumidor del servicio accederá a este mediante el contrato, logrando así la independencia entre el consumidor y la implementación del propio servicio.

- Servicios reutilizables.-El diseño y construcción de servicios debe estar basado en su reutilización dentro de la misma aplicación, dentro del dominio de aplicaciones de la empresa o incluso dentro del dominio público para su uso masivo.
- Servicios autónomos.-Un servicio debe tener su propio entorno de ejecución.
- Servicios sin estado.-Un servicio no debe guardar ningún tipo de información.
- Servicios descubribles.-Los servicios deben poder ser descubiertos para ser utilizados, evitando así la creación accidental de servicios que proporcionen las mismas funcionalidades.

Los principios de regulación aseguran que los principios de implementación sean aplicados de manera coordinada y apropiada.

- Bajo acoplamiento de servicios.-Independencia entre servicios.
- Abstracción de servicios.-El contrato de un servicio contiene la información esencial, ocultando la lógica al mundo exterior.
- Componibilidad de servicios.-Los servicios deben ser construidos de tal manera que pueda ser utilizado para construir servicios genéricos de más alto nivel, el cual estará compuesto de servicios de más bajo nivel.

El principal objetivo de SOA a través de los servicios es construir software altamente reutilizable para lo cual se basa en un conjunto de técnicas y principios.

3.2 Bus de Servicios Empresariales (Enterprise Service Bus -ESB)

El ESB es un middleware basado en estándares, que proporciona la comunicación entre los consumidores y los proveedores de un servicio por medio de mensajes.

El objetivo principal del ESB es proporcionar la interoperabilidad entre diferentes plataformas de hardware y softwa-

re. Debido a que los servicios pueden estar implementados en diferentes lenguajes de programación e integrados en diferentes plataformas es un rol fundamental del ESB permitir la transformación de datos.

Funcionalidades del ESB.

Según Dikmans y Van Luttikhuizen [3] las cuatro funcionalidades comunes que deben incluir los ESBs son: validación, composición de servicios, enrutamiento y transformación.

- Validación.-El ESB puede validar las solicitudes entrantes y/o las respuestas salientes a la interfaz especificada.
- Composición de servicios.-Servicios elementales se pueden agregar a servicios compuestos utilizando el ESB.
- Enrutamiento.-Los servicios están ocultos detrás del ESB y no visibles directamente para los consumidores. Cuando se recibe una solicitud de un consumidor de servicios, el ESB encamina al proveedor de servicio correcto. De esta manera, los cambios en los servicios pueden implementarse de manera transparente para los consumidores. A veces, más de un servicio está disponible para cumplir con la solicitud de servicio.

El ESB puede aplicar enrutamiento basado en el contenido para determinar qué solicitud de servicio necesita ser enrutada. El enrutamiento se basa en el contenido del mensaje, de ahí el nombre. Un caso especial de enrutamiento es dar soporte a múltiples versiones de un servicio, mientras que el proveedor de servicios solo ofrece la última versión del servicio.

- Transformación.-Un ESB expone los servicios en un formato de datos y protocolo apropiado para los consumidores del servicio. Por ejemplo, proporciona la transformación de protocolos y datos entre los consumidores del servicio y proveedores de servicios, para la integración con sistemas heredados, message-oriented middleware (MOM) y COTS, utilizando adaptadores exponiendo las interfaces propietarias como interfaces estandarizadas.

En la Figura 2 se representa el flujo básico del ESB que conecta a los consumidores de servicios con el proveedor de servicio.

El flujo ESB primero valida los mensajes de petición de entrada enviada por los consumidores, transforma el mensaje en un formato que es entendido por el proveedor de

servicios (por ejemplo, transforma de una estructura XML a otra) y luego rutea el mensaje al proveedor del servicio apropiado.

Desde la perspectiva del consumidor el servicio es ofrecido por el ESB.

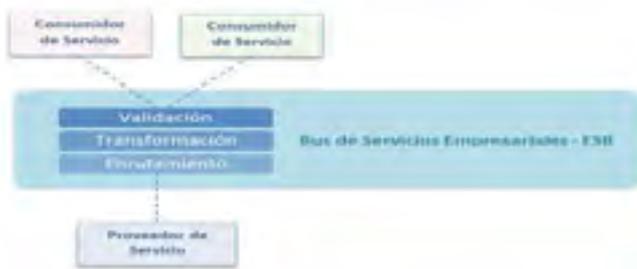


Figura 2. Conexión a servicios a través de un Bus de Servicios Empresariales - ESB [9]

3.3 Bajo acoplamiento

Es el concepto de la reducción de las dependencias del sistema. Debido a que los procesos de negocio se distribuyen a través de múltiples backends, es importante minimizar los efectos de las modificaciones y fallos. Las modificaciones se hacen demasiado arriesgadas, y las fallas podrían romper el panorama general del sistema. Sin embargo, hay un precio para el bajo acoplamiento: la complejidad. Los sistemas distribuidos con bajo acoplamiento son más difíciles de desarrollar, mantener y depurar. [9]

Al hablar de bajo acoplamiento de servicios nos referimos a que los servicios deben vincularse con otros servicios sólo en tiempo de ejecución.

4. SOA y servicios web /

Uno de los aspectos más importantes dentro de SOA son los servicios web (web services -WS).

La W3C define servicio web como “un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la web y que intercambian datos entre sí con el objetivo de ofrecer servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la Web. Estos servicios proporcionan mecanismos de comunicación estándar entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario.” [10]

Los servicios web permiten integrar aplicaciones creadas en lenguajes y plataformas diferentes a través de internet o una intranet.

Los servicios web se basan en un conjunto de estándares de comunicación, como son XML para la representación de datos, SOAP para el intercambio de datos y el lenguaje WSDL (Web Services Description Language) para describir las funcionalidades del WS [11]

“El uso de los servicios web es una buena práctica en la comunicación con los socios de negocios externos.” [12]

De las definiciones antes mencionadas se puede decir que los Servicios Web son el medio de comunicación que permiten la interoperabilidad entre aplicaciones a través del uso de estándares abiertos. De esta manera, aplicaciones desarrolladas en diferentes lenguajes de programación y ejecutadas en diferentes plataformas pueden intercambiar datos y presentar información dinámica al usuario.

4.1 Características de los servicios web

Los servicios web deben proporcionar las siguientes características:

Funcionalidad de negocio.-Los servicios web deben ser accesibles en remoto a través de la web.

Interfaz bien definida.-Ocultando la implementación real. El servicio puede estar implementado en cualquier tecnología, por ejemplo Java.

Interoperabilidad.-El proveedor del servicio y el solicitante pueden estar en tecnologías distintas, y aun así poder interactuar, desde el punto de vista del nivel de interoperabilidad de los servicios web, se pueden distinguir tres categorías:

- Privados.-El servicio web es consumido y desarrollado en la misma organización.
- Públicos.-El servicio web puede ser consumido por clientes internos y clientes de otras organizaciones, con los que previamente se ha negociado el modo de acceso.
- Globales.-El servicio web puede ser consumido por cualquier cliente en el mundo.

4.2 Componentes principales de los servicios web

Son tres los componentes principales en los servicios web: el registro de servicios, el proveedor de los servicios y el consumidor o cliente de los servicios.

Registro de servicios.-Es el directorio donde se encuentran centralizados los servicios. El registro proporciona un lugar común donde se puede publicar nuevos servicios o encontrar los ya existentes.

Proveedor de servicios.-Es la persona u organización encargada de implementar el servicio y ponerlo a disposición en internet o en una intranet. El proveedor de servicios describe la funcionalidad del servicio usando WSDL que contiene una lista de operaciones soportadas, tipos de datos intercambiados y la localización del servicio.

Consumidor o cliente de los servicios.-Es la persona u organización con necesidad de datos o servicios. El consumidor utiliza el servicio mediante la apertura de una conexión de red y el envío de una solicitud en formato XML.

4.3 Estándares en los servicios web

Hay cinco estándares fundamentales dentro de los servicios web. Dos son estándares generales que han existido previamente y se utilizaron como enfoque a los WS.

- XML.-Es usado como un patrón general para describir modelos, formatos y tipos de datos. La mayoría de estándares se basan en el estándar XML. De hecho, todos los estándares de servicios Web se basan en XML 1.0, XSD (XML Schema Definition).
- HTTP incluido HTTPS (Hypertext Transfer Protocol Security).-Es usado como un protocolo de internet de bajo nivel. Un estándar particular dentro del HTTP es HTTPS que es un protocolo que puede ser usado para enviar servicios web a través de la red usando internet, pero no es el único también podemos utilizar FTP (File Transfer Protocol) o SMTP (Simple Mail Transfer Protocol) para el mismo fin.

Los otros tres estándares especificados para los servicios web son:

- WSDL.-Es usado para definir las interfaces de los servicios. De hecho, se puede definir dos aspectos diferentes

de los servicios: su firma (nombre y parámetros) y los detalles de su implementación (protocolo y ubicación).

El Proveedor de servicios utiliza el archivo WSDL para generar el mensaje en el formato adecuado y para utilizar el protocolo de comunicación apropiado. El servicio solicitante utiliza ese archivo para entender cómo recibir y analizar el mensaje y cómo asignarlo al objeto o programa.

- SOAP.-Es un estándar que define el protocolo de los servicios web. Es un protocolo ligero que proporciona una manera de comunicarse entre las aplicaciones que se ejecutan en sistemas operativos diferentes, con diferentes tecnologías y lenguajes de programación.

SOAP define un formato común para mensajes XML a través de HTTP y otros protocolos de transporte, está diseñado para ser un mecanismo sencillo que se puede ampliar para incluir características adicionales, funcionalidades y tecnologías.

- UDDI (Universal Description, Discovery and Integration).-Es uno de los estándares básicos de los servicios web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros.

En la Figura 3 se representa la arquitectura de un servicio web. La arquitectura de los servicios web consta de tres funciones principales: descubrir, describir y transportar. Para realizar cada una de estas funciones se utiliza estándares basados en XML.

El proveedor de servicios web publica la información del servicio en el registro UDDI usando WSDL.

El consumidor del servicio web busca en el registro UDDI, una vez encontrado el servicio web deseado el UDDI entrega la descripción del servicio al consumidor. Entonces para obtener el servicio web el consumidor del servicio se conecta con el proveedor del servicio a través del protocolo SOAP. Finalmente el servicio web es entregado al consumidor vía SOAP.



Figura 3. Arquitectura de un servicio web.

5. Caso de uso

El Sistema Administrativo Contable de la empresa Besix-plus Cia. Ltda está implementado con SOA utilizando servicios web. Permitiendo integrar diferentes módulos de la aplicación, desarrollados por distintos programadores de manera sencilla y en un corto tiempo. Para el desarrollo y ejecución del proyecto primero se realizó la capacitación del equipo de desarrollo sobre las nuevas tecnologías y estándares de programación.

En base al conocimiento adquirido se han definido las herramientas de software a utilizar, entre éstas podemos destacar la plataforma J2EE, IDE Eclipse Galileo y el servidor de aplicaciones JBoss para el despliegue de la aplicación. Ya definida la tecnología a utilizar se ha procedido al levantamiento de requerimientos, basados en los datos obtenidos. El equipo de desarrollo ha definido los servicios web a implementar detallando la funcionalidad y parámetros de entrada y salida que tendrá cada uno, para así evitar la creación de servicios que tengan la misma funcionalidad.

Ya creados los servicios web, se registran y testean para la encontrar fallas. Una vez validados y corregidos los errores, los servicios web son compilados y empaquetados para luego ser desplegados en el servidor de aplicaciones.

La utilización de SOA en la ejecución del proyecto ha presentado múltiples ventajas entre estas: mejorar el tiempo de desarrollo de las aplicaciones, siempre y cuando los procesos a automatizar estén bien definidos.

También ha permitido optimizar los tiempos de repuesta ya que un servicio web puede atender varias peticiones de distintos cliente desde diferentes aplicaciones y en distinto lugar del mundo.

Además la utilización de estándares en la generación de código ha permitido que este sea entendible y de fácil manipulación para nuevos miembros del equipo de desarrollo.

6. Ventajas y Desventajas

La adopción e implementación de una arquitectura SOA en una organización presenta múltiples ventajas y desventajas tales como las siguientes.

Las aplicaciones basadas en SOA utilizan protocolos y estándares abiertos aumentando así la flexibilidad e interoperabilidad entre aplicaciones heterogéneas siendo esta la principal ventaja.

La arquitectura SOA permite alinear las TI con las estrategias de negocio de la organización, pero no es apto optar por una arquitectura SOA cuando se tiene un ambiente de TI homogéneos.

SOA permite a las organizaciones responder con rapidez a las nuevas exigencias empresariales, pero cuando hay pocas razones para cambiar la lógica de negocio, flujos de datos o procesos no es recomendable optar por SOA.

SOA permite construir aplicaciones a base de pequeños componentes configurados como servicios, generalmente servicios web que pueden estar disponibles en internet y pueden ser utilizadas por otras aplicaciones, lo cual conlleva a problemas de seguridad.

Entre las ventajas más evidentes e inmediatas de la implementación de SOA en una organización se encuentra la reducción de costos y tiempos de desarrollo y costos de integración. No siempre una SOA es la mejor opción arquitectónica ya que requiere de un desarrollo adicional así como de infraestructura tecnológica que soporte la arquitectura, lo que se traduce en aumento de los costos.

7. Conclusiones /

La arquitectura SOA simplifica la alineación de los procesos de negocios con las TI. SOA permite a las organizaciones separar los procesos de negocio y orientarlos a servicios, generalmente servicios web los cuales con creados en base a estándares abiertos e independientes de la plataforma en la que se ejecutan lo cual permite la interoperabilidad entre diferentes aplicaciones.

SOA hace posible la creación de sistemas altamente escalables, ya que está formado por servicios débilmente acoplados y altamente interoperables.

La implementación de la arquitectura SOA en una organización puede ser altamente costosa y llevar mucho tiempo si ésta no tiene bien definidos los procesos de negocio.

Finalmente concluiremos diciendo que SOA y servicios web tiene una relación muy estrecha pero no son lo mismo. Los servicios web son una tecnología estándar con la cual se puede implementar SOA.

Referencias /

- [1] Bass, L.; Clements, P. y Kazman, R. *Software Architecture in Practice*, 2nd Edition, Addison Wesley, 2003
- [2] ISO/IEC 2011 y IEEE 2011. (2011). *International Standard ISO/IEC/IEEE 42010*. (1a ed., 2011-12-01).
- [3] Dikmans, L. y Van Luttikhuisen, R. (2012). *SOA Made Simple*(pp. 19,121). UK: Packt Publishing.
- [4] Gartner, Inc. (2002). *Predicts 2003: SOA Is Changing Software*. Consultado de <http://www.gartner.com/id=380364> el 27 de Junio 2013
- [5] Brown, P. (2008). *Implementing SOA: Total Architecture in Practice*(pp. 16). SN: Addison Wesley Professional.
- [6] Lawler, J. y Howell-Barber, H. (2008). *Service-Oriented Architecture SOA Strategy, Methodology, and Technology* (pp. 31). US: Taylor & Francis Group, LLC.
- [7] OASIS. (2006). *Reference Model for Service Oriented Architecture 1.0*, OASIS Standard. Consultado de <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf> el 27 de Junio 2013
- [8] IBM. (2005). *IBM's SOA Foundation An Architectural Introduction and Overview*. (Versión 1.0). Consultado de <http://public.dhe.ibm.com/software/dw/webservices/ws-soa-whitepaper.pdf> el 27 de Junio 2013
- [9] Nicolai M. Josuttis. (2007). *SOA in Practice*. US: O'Reilly Media, Inc.
- [10] W3C. (2004). *Web Services Glossary*. Consultado de <http://www.w3c.es/Divulgacion/a-z/#servicios> el 5 de Julio 2013
- [11] Nicolás Tedeschi, Microsoft. *Web Services, un ejemplo práctico* Consultado de <http://msdn.microsoft.com/es-es/library/bb972248.aspx#authorbref> el 29 de Junio 2013
- [12] Mark Colan, IMB. (2004). *Service-Oriented Architecture expands the vision of web services, Part 1*.
- [13] Erl, Thomas. (2008). *SOA: principles of service design*. US: SOA Systems, Inc.
- [14] Javier Rosado Carrizo. *Arquitectura Orientada a Servicios* Consultado de <http://www.slideshare.net/jocluis/soa-java-day-uni> el 29 de Junio 2013