

Tipo de artículo: Artículo original  
Temática: Tecnologías de Bases de Datos  
Recibido: 14/11/2014 | Aceptado: 19/01/2015

## Implementación de lenguajes de contrato electrónico en Oracle Service Bus

### *Implementation of electronic contract languages in Oracle Service Bus*

Eva Flores Valdés

Instituto Superior Politécnico José Antonio Echevarría. Calle 114 No 11901 e/ Ciclovía y Rotonda, Marianao, La Habana, Cuba. Correo-e: [eflores@ceis.cujae.edu.cu](mailto:eflores@ceis.cujae.edu.cu)

---

#### Resumen

En el desarrollo e implementación de escenarios de integración han intervenido servicios que provienen de fuentes diversas, donde la atención se ha centrado en las operaciones que brindan y pocas veces se han tenido en cuenta sus requisitos no funcionales que incluyen aspectos como el tiempo de respuesta y la disponibilidad. Esto ha provocado la necesidad de establecer políticas automáticas de negociación entre las partes que interactúan con el servicio para tener en cuenta los aspectos anteriores. Para solucionar este problema se han utilizado los Acuerdos de Nivel de Servicio que son especificaciones no formales de los acuerdos entre las partes relativos a los requisitos no funcionales del servicio. Los lenguajes de contrato electrónico para la gestión de calidad de servicio se especificaron con el objetivo de formalizar estos acuerdos utilizando especificaciones estándares. Por otra parte, la herramienta *Oracle Service Bus* implementa la tecnología Bus de Servicios Empresariales que actúa como mediadora en la comunicación entre aplicaciones y tiene soporte para la definición de Acuerdos de Nivel de Servicio, para los servicios web que se conectan a esta. Este artículo tiene como objetivo principal verificar el soporte explícito de los lenguajes en *Oracle Service Bus* y se llegó a la conclusión de que no lo hace pero se propone una estrategia para implementar los aspectos fundamentales de los lenguajes utilizando componentes de la herramienta.

**Palabras clave:** acuerdos de nivel de servicio, lenguajes de contrato electrónico, servicios web, tecnología Bus de Servicios Empresariales.

#### Abstract

*In the development and implementation of integration scenarios have involved services from various sources, where attention has focused on the operations they provide and have rarely been taken into account its non-functional requirements which include aspects such as response time and availability. This has caused the need for automated policy negotiation between the parties that interact with the service to consider those aspects. To solve this problem we have used the Service Level Agreements that are not formal specification of the agreements between the parties relating to the non-functional service requirements. Electronic contract languages for quality of service management were specified in order to formalize these agreements using standard specifications. Oracle Service Bus implements the Enterprise Service Bus technology that mediates communication between applications technology and has support for defining Service Level Agreements for Web Services that connect to it. This article has the main objective of verifying the explicit support of the languages in Oracle Service Bus and concluded that it does not, but a strategy is proposed to implement the fundamental aspects of the languages used tool components.*

**Keywords:** *electronic contract languages, Enterprise Service Bus technology, service level agreements, web services.*

---

## Introducción

La gestión de la información se ha convertido en un factor fundamental para el desempeño óptimo del sector empresarial. Es muy frecuente que los sistemas implantados deban evolucionar en conjunto con la organización en la que se encuentran funcionando, con el objetivo de satisfacer los nuevos requerimientos que surgen en la misma. En este proceso de crecimiento se ve la necesidad de que estos sistemas compartan información y se trabaje con esta de manera automatizada para dar soporte a los disímiles procesos de negocio definidos en la empresa.

Para que las aplicaciones se comuniquen, existen varios mecanismos, siendo uno de los más utilizados los servicios web, por la facilidad que brindan al estar basados en estándares abiertos y ser soportados por una gran cantidad de lenguajes de programación (Díaz, 2010). En entornos donde se adoptan estas soluciones se tiene que diferentes aplicaciones exponen sus funciones a través de servicios y otras los utilizan de la forma que vienen o los componen con otros con el objetivo de obtener nuevas funcionalidades. Con frecuencia, cuando los sistemas clientes adquieren estos servicios, obtienen solo los requisitos funcionales de este, y no tienen noción de los no funcionales como su tiempo de respuesta, su disponibilidad o el tiempo promedio de recuperación en caso de fallo; como consecuencia, estos sistemas tienen que predecir el comportamiento de los servicios en la medida que los utilicen.

Las SLA (Acuerdos de Nivel de Servicio) constituyen una solución a este problema y no son más que contratos bilaterales entre un proveedor y un consumidor de servicios. En estos acuerdos se definen las propiedades del servicio y las garantías que ofrece el proveedor de entregar el servicio con una calidad definida, capacidades para monitorear las propiedades en un tiempo proporcionado y acciones a realizar en caso de que el servicio no se provea con la calidad acordada (Waeldrich, 2011). Estos contratos se pueden especificar de manera estática mediante un documento o de manera dinámica a través de lenguajes formales basados en estándares.

Por otra parte han evolucionado tecnologías que apoyan el proceso de implementar escenarios de integración, una de estas es ESB (Bus de Servicios Empresariales) definida como una plataforma de integración basada en estándares que combina mensajería, servicios web, transformación de datos y enrutamiento inteligente, conecta de forma fiable y coordinada la interacción de un número importante de aplicaciones diversas a través empresas extendidas con integridad transaccional (Chapell, 2004). *Oracle Service Bus* es una herramienta que pertenece a la compañía Oracle e implementa las capacidades de la tecnología ESB. Al actuar como intermediario entre la comunicación entre sistemas, sería vital evaluar el soporte de esta para gestionar contratos de calidad del servicio (SLA). Este trabajo tiene el objetivo de hacer una valoración acerca del soporte de la herramienta para gestionar calidad de servicio a través de lenguajes de contrato electrónico.

## Materiales y métodos

La investigación realizada se enmarca en la gestión de calidad de servicios vista desde la herramienta *Oracle Service Bus* la que implementa la tecnología ESB. En esta sección se expone el estado del arte de los elementos tecnológicos que constituyen la base de la investigación.

### Contratos electrónicos para gestión de calidad de servicio

En el desarrollo e implementación de escenarios de integración intervienen servicios de varias partes por lo que se hace necesario establecer políticas automáticas de negociación entre estas. Varios autores han acordado que, en ambientes de servicios electrónicos los contratos son importantes para alcanzar la interoperabilidad de los procesos de negocio y reforzar su comportamiento (Fernández, 2007). La idea fundamental de la definición de contratos es la declaración explícita de los términos bajo los cuales los servicios web van a interactuar; estos incluyen:

- La especificación del contexto en el cual el contrato es ejecutado.

- La especificación de los actores y sus roles en el sistema.
- Las especificaciones de las acciones de las partes involucradas, estas son actividades que son ejecutadas y pueden cambiar sin afectar el funcionamiento del sistema
- La especificación de métricas para evaluar la calidad de servicio (QoS) asociada a la ejecución de las acciones correspondientes.
- La especificación de mecanismos para medir las métricas, es decir el establecimiento de vínculos entre la definición de métricas y las acciones para medirlas (Fernández, 2007).

Desde hace años se han propuesto mecanismos para formalizar los acuerdos entre partes que intervienen en la automatización de procesos. En 1988 surgió una idea para declarar, de manera explícita, las relaciones entre las partes con roles de consumidor y proveedor; la que fue propuesta por Meyer para mejorar el diseño de aplicaciones orientadas a objeto y se aplicó en el lenguaje de programación Eiffel. En esta propuesta un contrato se entiende por la especificación de *acciones* comenzando con algunas *precondiciones* y obteniendo resultados definidos en *poscondiciones* (Meyer, 1997).

Una expresión formada por una *precondición* más una *condición* más una *postcondición* es considerada una *afirmación*. Esta representa una expresión que abarca entidades del software y las propiedades que esas entidades deben satisfacer en etapas de la ejecución del software. La utilización de estas tienen varias aplicaciones en:

- Escribir software correcto
- Facilitar la documentación.
- Soporte para hacer pruebas, corregir errores en el código y asegurar su calidad.
- Soporte para la tolerancia a fallos (Meyer, 1997).

Otro enfoque es *Electronic Business XML (ebXML)* que es resultado de un proyecto desarrollado por la Organización para el Avance de los Estándares para la Información Estructurada (OASIS siglas en inglés) y por United Nation's Center for Trade Facilitation and Electronic Business (UN/CEFACT) en el año 1997. ebXML provee una infraestructura estándar para los negocios electrónicos que permite el intercambio de información entre estos. Como parte de su especificación se definió además ebXML Collaboration Protocol Profile and Agreement (CPPA) que define un esquema XML para controlar los acuerdos entre las partes involucradas en los negocios (Schonberger, 2010).

### **Lenguajes de contrato electrónico**

Los servicios web han emergido como el mecanismo más utilizado para hacer que las aplicaciones se comuniquen entre sí y se definen como un componente de software que se comunica con otras aplicaciones codificando los mensajes en lenguaje de marcado extensible (eXtensible Markup Language, XML siglas en inglés) y enviando estos mensajes a través de protocolos estándares de Internet tales como el protocolo de transferencia de hipertexto (Hypertext Transfer Protocol, HTTP, siglas en inglés) (Rosales, 2010).

Los servicios web exponen las funcionalidades de las aplicaciones sobre las que están definidos y son consumidos por otras aplicaciones de acuerdo de sus necesidades, por lo que es de gran importancia que estos servicios estén disponibles y funcionen con la máxima calidad posible. La tecnología de servicios web provee una plataforma de integración basada en el Lenguaje de Descripción de Servicios Web (WSDL, siglas en inglés) para la descripción de estos, en Universal Description, Discovery and Integration (UDDI, siglas en inglés) para su persistencia y descubrimiento; y en el Protocolo Simple de Acceso a Objetos (SOAP, siglas en inglés) como mecanismo de comunicación. Estos proveen la oportunidad de enlazarse dinámicamente con otros servicios en tiempo de ejecución y de establecer o eliminar conexión con un proveedor de servicios (Rosales, 2010). Los contratos electrónicos

especifican la forma en que las interacciones proveedor-cliente son llevadas a cabo y qué partes contractuales intervienen en estas (Jones, 2010).

En términos generales se define la calidad de servicio como “*Cuan bien se ejecutan las operaciones de un servicio*” (Tosic, 2010) además se plantea que este concepto existe aún sin que se especifique o se mida. Por otra parte se define la gestión de calidad de servicio como la capacidad de determinar el estado de un servicio(definido como monitoreo) más la capacidad de mantener este en el estado deseado (definida como control) (Tosic, 2010).

Las especificaciones de calidad de servicio deben establecer el momento, el lugar y cómo van a monitorear y controlar métricas QoS establecidas para este; estas especificaciones se pueden clasificar en:

- Implícitas: Son construidas en la implementación del servicio, por lo que carecen de flexibilidad y no se pueden reutilizar.
- Contratos: Representan acuerdos formales.
- Políticas: Objetivos y/o reglas para la gestión de alto nivel para la seguridad, QoS, etc.

Con el objetivo de formalizar estas ideas se han hecho varias propuestas; una de ellas es extender los lenguajes WSDL, UDDI o WS-BPEL para dar soporte, además, a los contratos para gestionar QoS. Este acercamiento tiene la ventaja de que es simple y se puede asociar el descubrimiento de la gestión QoS con el descubrimiento de servicios web. Por otra parte tiene la desventaja de que al tener un lenguaje con dos objetivos, cualquier cambio en tiempo de ejecución que ocurra relativo a alguno de ellos implica modificar todos los ficheros afectados de uno de esos tipos, lo que constituye un proceso muy complejo.

Otra idea de formalización la constituyen los Acuerdos de Nivel de Servicio (*Service Level Agreement*, SLA siglas en inglés) que constituyen un conjunto de garantías de calidad de servicio y obligaciones de las partes involucradas (Keller, 2007). En un entorno de sistemas orientados a servicios estas pueden adoptarse de forma estática mediante la especificación del contrato SLA en un documento y de forma dinámica; a través de su representación en lenguajes formales.

Los lenguajes de contrato electrónico surgen como alternativa para estandarizar los acuerdos entre las partes que intervienen en la comunicación de servicios web y así tratar el procesamiento SLA de manera dinámica. Actualmente existen varias propuestas de este tipo, en este artículo se van a analizar dos de las más representativas (Tosic, 2010).

### **1. Web Service Level Agreements**

WSLA es una especificación desarrollada por la compañía IBM en 2003 y provee un marco de trabajo para definir y monitorear SLA de servicios web. Esta especificación incluye un lenguaje WSLA basado en XML.

De manera general esta especificación tiene tres componentes fundamentales:

- La descripción de las partes involucradas que incluye los roles que desempeñan, así como las interfaces con las acciones que exponen hacia otras partes del contrato.
- La definición del servicio, las métricas a gestionar y como se va a realizar esta gestión.
- La representación de las obligaciones de las partes de realizar alguna acción en caso de que no se cumpla alguna garantía acordada.

El siguiente diagrama de clases ilustra detalladamente los conceptos y su relación que conforman la estructura de un documento WSLA.

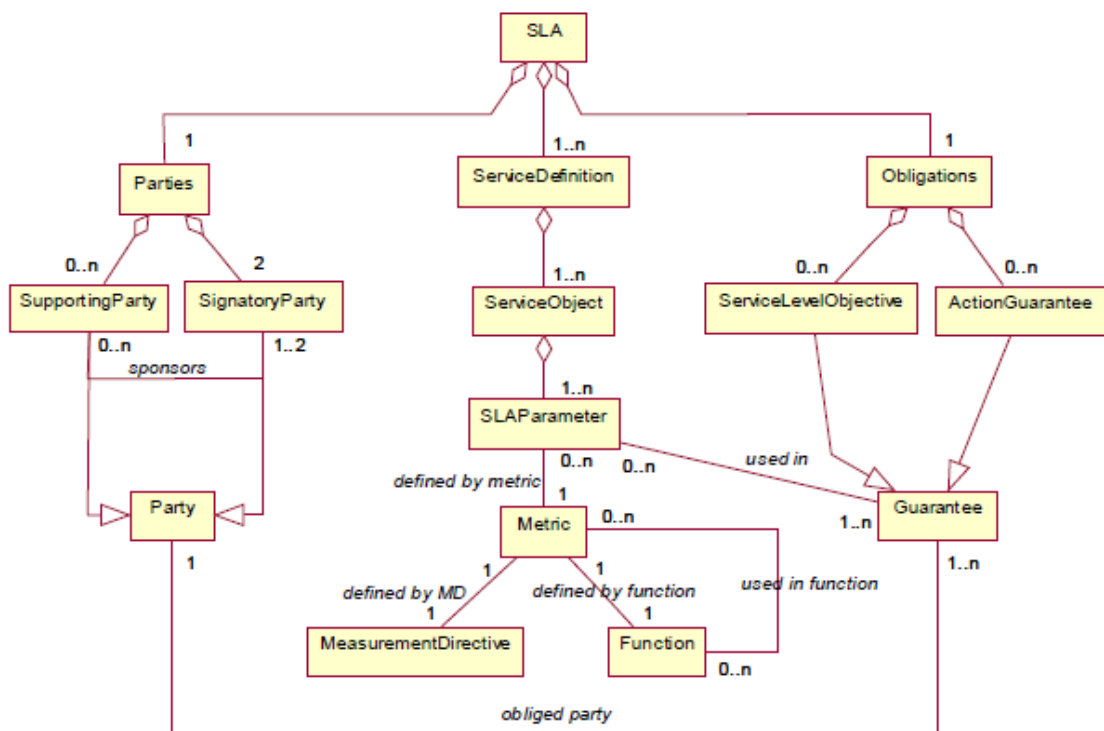


Figura 1. Relación entre los componentes de un documento WSLA (Ludwing, 2003).

Para las partes (*Parties* en la figura) involucradas definidas en WSLA se pueden especificar dos roles; el primero: *SignatoryParty* que representan al proveedor y el consumidor de un servicio; es necesario puntualizar que en este lenguaje se puede definir un contrato entre solo dos partes de este tipo. El otro rol *SupportingParty* lo desempeñan aquellas partes encargadas de medir y computar las métricas para el rendimiento de las operaciones del servicio.

En la definición del servicio (*ServiceDefinition* en la figura) se especifican los parámetros SLA (*SLAParameter* en la figura) que describen una propiedad observable del servicio cuyo valor puede ser obtenido a través de una medición (Ludwing, 2003). Las métricas son definiciones de valores de propiedades del servicio que son medidas desde un sistema u otras métricas o constantes.

En las obligaciones se encuentran los *ServiceLevelObjective* y las *ActionGuarantee*; el primero expresa el compromiso de mantener un estado particular del servicio en un período de tiempo dado. Este estado es definido como una expresión que incluye a un parámetro SLA definido. Por otra parte las *ActionGuarantee* especifican el compromiso de ejecutar un conjunto de actividades si se cumplen algunas precondiciones definidas.

## 2. WS-Agreement

Es una especificación de la Open Grid Forum (OGF) que provee un protocolo para definir acuerdos entre un cliente de servicios y un proveedor. Esta utiliza un lenguaje basado en XML para especificar los acuerdos el cual incluye:

- Un esquema para especificar un acuerdo
- Un esquema para especificar plantillas de acuerdo con el objetivo de descubrir otras partes compatibles
- Un conjunto de *port types* y operaciones para la gestión del ciclo de vida de un acuerdo, el cual incluye la creación, expiración y estados del monitoreo de acuerdos (Tang, 2013).

La estructura general de un acuerdo en la especificación consiste en la descripción del contexto en el que el este es establecido, el propio servicio y los términos de garantía. La siguiente figura ilustra los componentes principales de un documento este tipo.

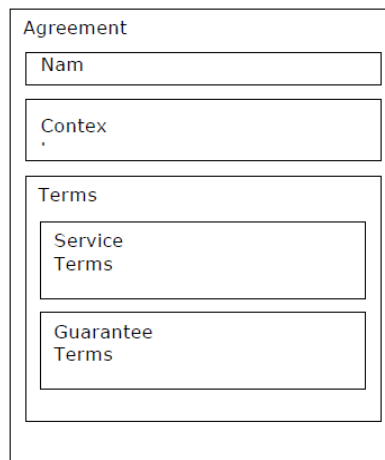


Figura 2. Estructura de un documento WS-Agreement (Fernández, 2007)

En la definición de un contrato con WS-Agreement se especifican tres atributos fundamentales:

- Nombre del contrato.
- Contexto: Se define la información acerca del documento, la que se encuentra conformada por el iniciador del acuerdo, el proveedor del servicio, el tiempo de expiración, entre otras (Waeldrich, 2008).
- Términos: este a su vez está formado por los términos del servicio que describe sus características principales y los Términos de Garantía que expresan acciones y penalizaciones (Fernández, 2007).

WS-Agreement define plantillas de acuerdos para especificar tipos de acuerdos en un dominio específico de aplicación, donde una plantilla de acuerdo no es más que un acuerdo (definición de su nombre, tiempo de terminación, contexto y términos) y la especificación de un conjunto de condiciones a ser cumplidas.

Una vez que se define un acuerdo entre un proveedor y cliente de servicio web, el contrato entra en la etapa de gestión, donde se establece su monitoreo hasta que se acabe el tiempo de vida definido para este (Fernández, 2007).

### **Vistazo de la tecnología Bus de Servicios Empresariales.**

El ESB ocupa la capa de abstracción intermedia entre los distintos sistemas de una o varias organizaciones, proporcionando mecanismos de comunicación y transformación de mensajes basados en estándares. Este debe sustituir toda interacción directa entre aplicaciones con el propósito de que estas se comuniquen a través del bus. Otra de sus características es que proporciona un sistema de mensajería, lo que se realiza a través de adaptadores. Además, el intercambio de mensajes debe ser independiente de la plataforma, lo que permite al ESB integrar aplicaciones que se ejecutan en diversos sistemas operativos (Chapell, 2004; Martiniano, 2009).

En el sitio de ayuda de Microsoft MSDN “El término ESB (Enterprise Service Bus) se usa corrientemente en el contexto de implementación de capacidades de mensajería en una infraestructura orientada a servicios” (Microsoft, 2012).

En libro “Enterprise Service Bus” se refleja cómo “Una plataforma de integración basada en estándares que combina mensajería, servicios web, transformación de datos y enrutamiento inteligente, conecta de forma fiable y coordinada la interacción de un número importante de aplicaciones diversas a través empresas extendidas con integridad transaccional” (Chapell, 2004).

Lo esencial de esta tecnología es que está diseñada para actuar como mediadora en la capa (middleware) entre distintos sistemas o aplicaciones sin importar en qué tecnología se hayan desarrollado y si están dispersas por la red. Para lograr esto utiliza estándares que le garanticen la interoperabilidad y un conjunto de adaptadores para comunicarse con las mismas.

### **Arquitectura funcional y conceptos fundamentales de la herramienta Oracle Service Bus.**

Existen varias maneras de enfrentarse a la implementación de escenarios de integración, una de estas es programando interfaces de comunicación entre las mismas, esta integración conecta punto a punto las aplicaciones; es particular para cada escenario de integración, es decir no se pueden reutilizar el código o los componentes realizados si se desea utilizar las mismas aplicaciones en otro escenario. Otro enfoque es utilizar la tecnología ESB que se basa en la implementación de un bus al cual se conectan las aplicaciones y soluciona los problemas del primer enfoque, para lo que propone varias capacidades. Estas son implementadas por la herramienta *Oracle Service Bus*, que tiene un conjunto de funcionalidades agrupadas en cuatro capas.

Capa de Mensajería: Se proveen las funcionalidades para conectar servicios y/o aplicaciones con OSB, lo que se realiza mediante el aprovechamiento de las normas de transporte de servicio web, de los protocolos tradicionales de mensajería y de la configuración de transportes personalizados para la empresa (ORACLE, 2010).

Capa de Seguridad: Se realizan las configuraciones relacionadas con la seguridad de los mensajes y con los protocolos de transporte relacionados (ORACLE, 2010).

Capa de Composición: Se encuentran un conjunto de componentes que permiten la configuración de servicios y el modelado del flujo de mensajes. Estos permiten enrutar de forma lógica los mensajes entre los diferentes servicios, abstrayéndolos de la complejidad de conocer sus protocolos de comunicación sus requisitos de seguridad y su estructura. Además, en esta capa se incluyen capacidades para el descubrimiento de servicios, la para la importación automática y la sincronización de los servicios con registros UDDI (ORACLE, 2010).

Capa de Gestión: OSB ofrece capacidades de gestión de servicios integrados, lo que posibilita optimizar el control de todos los mensajes. La capa de gestión proporciona un entorno de gestión de servicios que incluye un sistema dinámico de configuración para las políticas y servicios (ORACLE, 2010).

Por su rol como intermediario, la herramienta OSB define dos conceptos fundamentales que intervienen verticalmente en todas las capas de su arquitectura funcional.

Servicio de Negocio: Son la representación de los servicios de las aplicaciones desplegadas con las cuales se desea intercambiar información, en OSB. Este recurso permite realizar configuraciones como el transporte a utilizar con la aplicación destino, políticas de seguridad, entre otras (Jones, 2010).

Servicios Proxy: Constituye uno de los principales elementos dentro de la arquitectura de OSB. Estos se convierten en la interfaz que utilizan los consumidores para conectarse con los proveedores de servicios. Una de las ventajas que posee, es que permite definir flujos de mensajes, en el cual se puede implementar lógica para encaminar los mensajes a múltiples servicios de negocios (Jones, 2010).

De manera general los dos tipo de servicio colaboran entre sí en la implementación de un escenario de integración determinado los primeros representando a los servicios de las aplicaciones que tienen el rol de proveedor y los segundos construyendo, a través de la configuración de un flujo de mensaje donde se invocan a los servicios de negocio, las funcionalidades que requieren las aplicaciones con el rol de clientes. La siguiente figura ilustra esta relación.

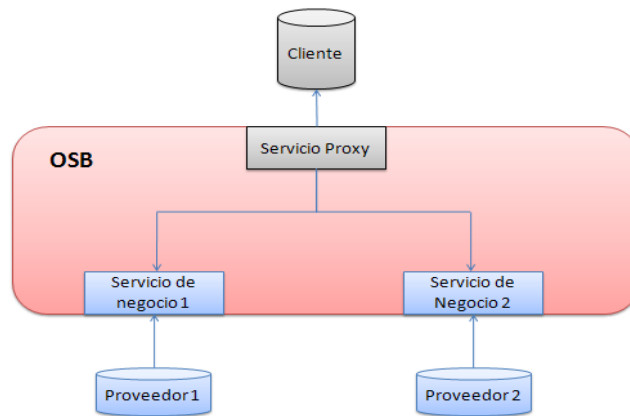


Figura 3. Servicios de negocio y proxy en *Oracle Service Bus*

## Resultados y discusión

Una de las funciones que realiza OSB como intermediario entre los proveedores de servicios y los clientes, es garantizar la definición y gestión de alertas de servicio. Una alerta es una notificación que se envía cuando se cumple una condición determinada, estas alertas se establecen para informar al equipo de operaciones sobre las cuestiones relacionadas con el estado de los servicios proxy y servicios de negocio, o la calidad del servicio prestado. Se pueden configurar con múltiples niveles de relevancia: advertencia, normal, menor, mayor, crítica, y fatal. Esto posibilita que el usuario pueda establecer en una alerta el nivel de atención que debe darle a un servicio determinado. Las reglas de alertas son las condiciones que se definen para dispararlas.

En la reglas de alertas se pueden combinar múltiples condiciones para cada servicio de negocio o servicio proxy. Las condiciones de las alertas se pueden basar en parámetros (tasa de éxito, proporción de éxito, proporción de fracaso, cantidad de mensajes, cantidad de errores, tiempo de respuesta, tiempo mínimo de respuesta, tiempo máximo de respuesta, etc.) para crear las reglas. OSB posibilita establecer dichas reglas basadas en SLA sobre servicios de negocio y servicios proxy. Estas definen el nivel de precisión y calidad que se espera de los servicios de negocio y los servicios proxy y están agrupadas bajo las siguientes categorías:

- Contador: Las reglas de alertas clasificadas bajo este concepto realizan un seguimiento del conteo de los eventos en tiempo de ejecución de OSB, tales como el número de mensajes, el número de fallos, etc.
- Intervalo: Las reglas de alerta con esta clasificación realizan un seguimiento del tiempo transcurrido entre dos eventos bien definidos. En este intervalo se calculan, el total, promedio, mínimo y máximo de tales eventos en OSB.
- Estado de la URI del servicio: Las reglas de alerta asociadas realizan un seguimiento del estado de un servicio, si este está en línea o fuera de línea por ejemplo.

La siguiente tabla muestra las métricas definidas para cada clasificación:



Tabla 1. Métricas y clasificación definidas en Oracle Service Bus.

Categoría	Métrica	Destino
Count	Conteo de Errores (Error Count)	Servicios proxy de negocio
	Proporción de fallos (Failure Ratio)	
	Conteo de mensajes (Message Count)	
	Proporción de logros (Success Ratio)	
	Validación de conteo de errores (Validation Error Count)	
	Conteo de errores de WSS (WSS Error Count)	
Intervalo	Tiempo de respuesta mínimo (Min Response Time)	Servicios de negocio
	Tiempo de respuesta máximo (Max Response Time)	
	Tiempo promedio de respuesta (Average Response Time)	
Estado	Todas las URI fuera de servicio (All URIs offline)	Servicios de negocio
	Todas las URI en servicio (All URIs online)	
	Cualquier URI fuera de servicio (Any URI offline)	
	Cualquier URI (Any URI online)	

Un recurso de gran importancia para la definición de las alertas SLA es el destino de alerta. Este contiene una lista de recipientes que pueden recibir notificaciones de alerta desde Oracle Service Bus. Además de definir los recipientes JMS y Correo electrónico, las notificaciones pueden ser enviadas como SNMP traps y ser procesadas por otros sistemas de monitoreo empresariales; o ser enviadas hacia el módulo de reportes de Oracle Service Bus y ser procesadas por un proveedor de reportes propio desarrollado utilizando las APIs de reporte que brinda OSB.

Un recipiente JMS representa la configuración para enviar el destino de una alerta hacia una cola JMS. Para este se especifican la dirección de la cola JMS hacia donde se va a enviar la alerta, el tipo de cola hacia, el formato del mensaje y el tipo de codificación de caracteres para este. Para el recipiente E-mail se configuran los correos electrónicos de las personas que serán notificadas con un alerta y un servidor SMTP que va a representar la conexión a un servidor de correo que se va a utilizar para enviar la notificación. El siguiente diagrama de clases especifica la relación entre los conceptos descritos anteriormente.

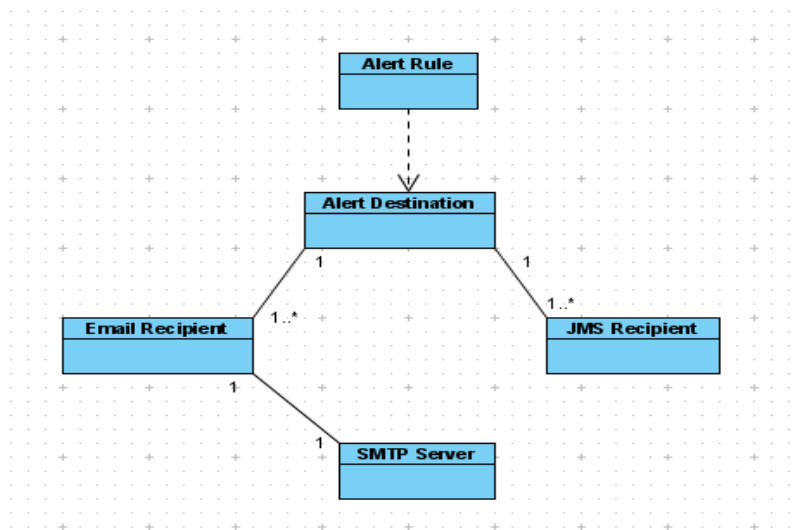


Figura 4. Relación entre los conceptos de monitoreo en OSB.

Para el procesamiento de las alertas SLA que se pueden definir en *Oracle Service Bus* se utiliza la especificación *Java Message Service* (JMS) como posible destino de esta con vistas a un futuro procesamiento. JMS es una especificación que define un API estándar para acceder a sistemas de mensajería. Esta permite a las aplicaciones consumir y enviar mensajes hacia cualquier servidor de mensajería que tenga soporte para esta. JMS presenta dos modelos de mensajería:

Modelo punto a punto: El productor del mensaje lo crea y lo envía a un destino llamado cola. Los consumidores de este mensaje se comunican con esta, obtiene el mensaje y lo procesa. Cada mensaje puede ser procesado por un consumidor y son entregados en el mismo orden en que se almacenaron en la cola siguiendo el patrón Primero en llegar Primero en salir (First In First Out, FIFO siglas en inglés) (Schmutz, 2012).

Modelo publicador-subscriptor: El productor del mensaje lo crea y lo envía a un destino llamado tópico. Estos son entregados a consumidores activos conocidos como suscriptores, los que tienen la opción de suscribirse a un tópico determinado y así buscar solo los mensajes que le interesan. Estas suscripciones pueden clasificarse en durables o no durables; la primeras están siempre vigentes ya sea que el consumidor esté conectado y desconectado al servidor JMS y las no durables solo se establecen mientras el consumidor esté conectado al servidor (Schmutz, 2012).

### Lenguajes de contrato electrónico en Oracle Service Bus.

La capacidad de monitoreo de servicios en *Oracle Service Bus* está basada en la definición de alertas que utilizan el concepto SLA como núcleo. Muchas veces estas se configuran con el objetivo de detectar fallos en los servicios, sin conocer que además de esta función, las SLA permiten ser configuradas para determinar el estado del rendimiento de servicios, es decir, no solo se utilizan para detectar fallos sino que su objetivo tiene un mayor alcance y es el de la prevención de estos.

*Oracle Service Bus* no implementa de manera explícita ninguno de los lenguajes de contrato electrónico analizados en este artículo, aunque se ha visto que estos constituyen una representación formal de las SLA que sí tienen soporte en la herramienta; por lo que la base de estos lenguajes tiene una representación en componentes funcionales de *Oracle Service Bus*. A continuación se hará una propuesta de cómo utilizar estos y su relación con los elementos de los lenguajes WS-Agreement y WSLA para gestionar la calidad de servicio en la herramienta. Suponer que se tiene la premisa de que cada servicio web que exponga una aplicación debe tener definido un contrato utilizando uno de los lenguajes vistos.

- Crear servicios de negocio: Para cada servicio de las aplicaciones proveedoras se crea un servicio de negocio en OSB con el objetivo de hacer las configuraciones de monitoreo sobre este.

- Establecer las configuraciones operacionales: En estas se definen las propiedades generales de monitoreo para el servicio de negocio como, la habilitación de este, el intervalo de agregación en el que se van a chequear las condiciones de las alertas, entre otros.
- Determinar las métricas: Estas se obtienen a través del análisis de los contratos WSLA o WS-Agreement definidos por las aplicaciones proveedoras. En el caso del primero las métricas se determinan por los parámetros SLA (sección SLAParameter en el contrato) y en WS-Agreement por los Términos de Garantía (GuaranteeTerms en el documento de contrato).
- Determinar las reglas de alerta: Estas especifican las condiciones acordadas para el comportamiento de propiedades del servicio. En los contratos WSLA y WS-Agreement se determinan a través de los SLO ya que es un concepto presente en ambos.
- Determinar Acciones: Las acciones especifican qué hacer en caso de que no se cumplan las condiciones acordadas. En el contrato WSLA se determinan a través de las Acciones de Garantía (ActionGuarantee en el documento) y en WS-Agreement a través de las Penalizaciones (Penalty en el documento) o Recompensa (Reward en el documento del contrato).
- Crear Destino de Alerta: En este componente se configuran los destinos de una alerta en caso que se disparen. Si en las acciones determinadas solo se incluye establecer una notificación por correo electrónico, solo se configura para el destino de alerta recipientes de correo electrónico; información que se obtiene contactando a las partes definidas en los contratos WSLA y WS-Agreement; si no está especificado en el documento. En caso de que se incluyan acciones que puedan ser procesadas por un flujo de mensaje, tal como la invocación a un servicio web, se configuran recipientes JMS, estableciendo la dirección de la cola JMS hacia dónde va a ser enviada la alerta en caso de que se dispare.
- Crear alerta SLA: Después de obtenida toda la información del contrato, se procede a crear la alerta, donde se configuran sus propiedades generales como el intervalo de agregación, la severidad de la alerta, el destino de alerta que va a utilizar, entre otras. Con las métricas definidas se determina y configura que propiedad se va a monitorear y se determinan las expresiones basándose en las reglas de alerta obtenidas de los contratos
- Crear servicio proxy de escucha: si se configuró un recipiente JMS en el destino de alerta utilizado entonces se debe crear un servicio proxy que esté monitoreando la cola JMS especificada para capturar las alertas que entren a la cola y procesarlas.
- Configurar acciones en el flujo de mensaje: Se configura de acuerdo a las pautas de las acciones determinadas en el contrato.

La siguiente imagen muestra un diagrama con las actividades descritas:

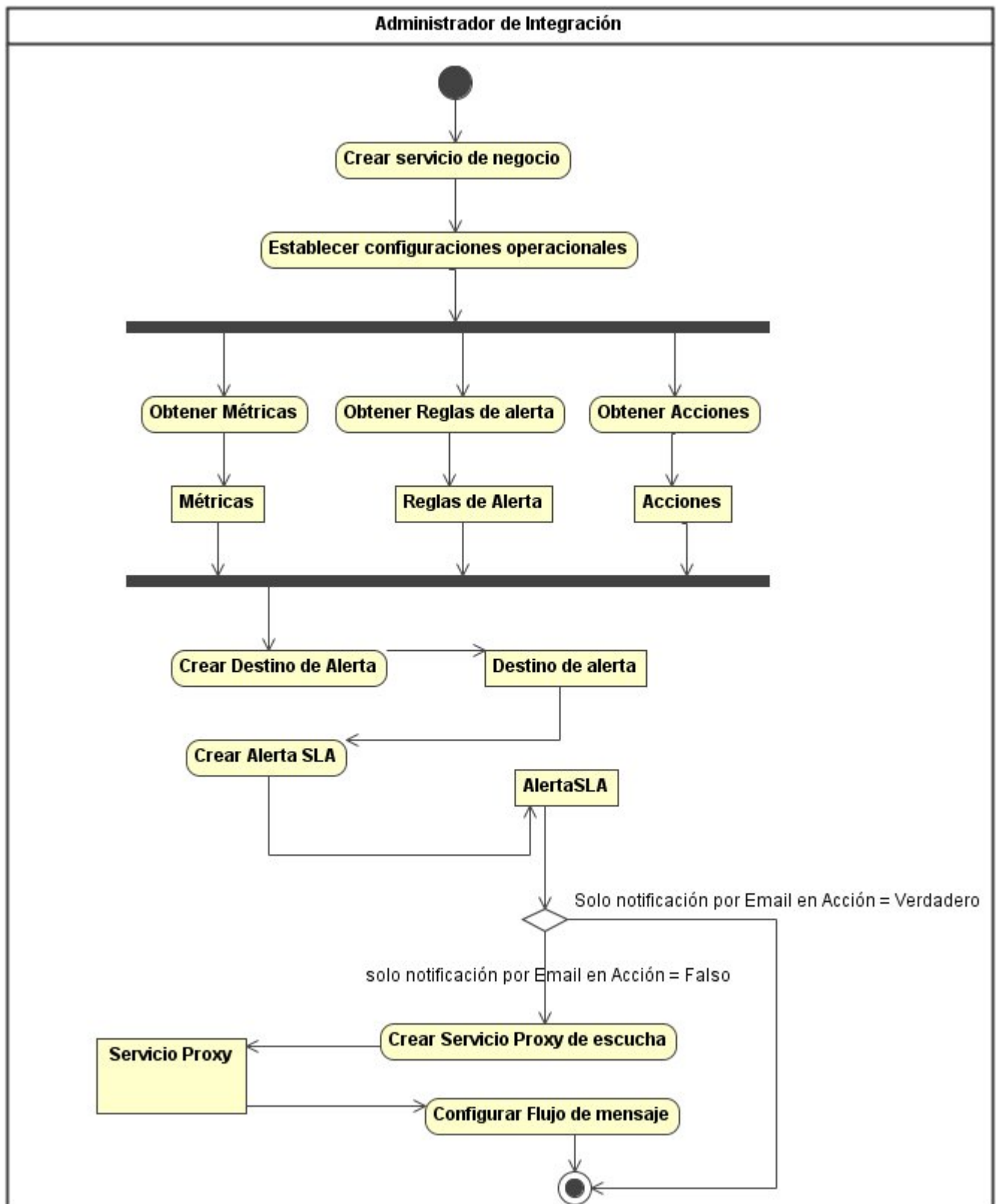


Figura 5. Diagrama de actividades de la estrategia propuesta para la gestión de calidad de servicio en OSB utilizando los elementos fundamentales de los lenguajes de contrato electrónico.

En el Complejo de Investigaciones de Tecnologías Integradas (CITI) se está desarrollando un proyecto de investigación llamado Integración de Información basada en la Descripción Semántica de Fuentes de Datos Heterogéneas (SEMANTINFO), donde se utiliza la tecnología ESB para integrar aplicaciones utilizadas en el ámbito de la CUJAE. Específicamente, se integraron los sistemas SIGENU, Copérnico y Directorio CUJAE, mediante la composición de varios de los servicios web que exponen. Estos no tienen definida ninguna garantía acerca de su comportamiento. Específicamente es de interés, en el proyecto, monitorear el tiempo de respuesta de los servicios;

por lo que, en vistas de que no había información al respecto, se decidió estimar los valores de tiempo de respuesta utilizando herramientas de prueba y aplicar el flujo anterior con los resultados. Como acción solo se tomó la notificación por correo electrónico al administrador de integración debido a que se estaba trabajando con valores estimados. Una vez configurados los componentes, siguiendo la propuesta anterior, se pudo monitorear el comportamiento de esos servicios respecto a la métrica definida. Durante el proceso hubo cambios en los valores asociados al tiempo de respuesta debido a que el comportamiento no fue uniforme, hecho provocado, en algunas ocasiones, por inestabilidad en la red y en otras, por el aumento del flujo de información a través de esta. Durante el intervalo de tiempo que se estuvieron monitoreando los servicios, se observó de forma indirecta, que estos estuvieron fuera de servicio por algunos días, debido a fallos en los servidores en que están publicados, hecho que se repitió con posteridad, por lo que se decidió incorporar la métrica disponibilidad de servicios al proceso de monitorización con lo que se logró identificar en tiempo este tipo de fallo.

## Conclusiones

En un ambiente de desarrollo de escenarios de integración donde las aplicaciones intercambian información utilizando como interfaces de comunicación a los servicios web; es común brindar más importancia a los requisitos funcionales del servicio. Este ambiente al estar enteramente basado en tecnologías de redes y en arquitecturas distribuidas es más vulnerable a fallos externos por lo que es de suma importancia tomar en cuenta estos aspectos para brindar estos servicios con una calidad bien definida.

Los lenguajes de contrato electrónico vistos en este trabajo son intuitivamente aplicados a arquitecturas de conexión punto a punto, es decir que una aplicación cliente se conecte a una proveedora de servicios directamente. La tecnología ESB forma parte de una arquitectura de integración más compleja donde actúa como intermediaria entre la comunicación, eliminando las desventajas de la conexión punto a punto. Debido a que los lenguajes de contrato electrónico permiten especificar cuán buena será la ejecución de las operaciones de este, a través de la definición formal de acuerdos de garantía y calidad; en este artículo se definieron los lenguajes de contrato electrónico como vía para lograr una gestión de calidad de servicio y cómo podrían ser implementados por la herramienta *Oracle Service Bus* basada en la tecnología ESB; con el objetivo de valorar la utilización de estos contratos a arquitecturas más complejas. Del estudio realizado se puede concluir que la herramienta Oracle Service Bus permite la gestión de calidad de servicio por tener soporte a la creación de Acuerdos de Nivel de Servicio (SLA), donde se especifican las propiedades que se van medir y las acciones a realizar si estas propiedades no se cumplen. Por otra parte la herramienta no tiene soporte explícito para adoptar contratos realizados en los lenguajes vistos aunque sí cuenta con componentes que tienen la misma responsabilidad que los elementos del contrato. Además, sólo se pueden adoptar en su totalidad aquellos que especifiquen las mismas métricas que vienen definidas en la herramienta y definan acciones que puedan ser procesadas automáticamente, como son notificaciones por correo electrónico o ejecución de un servicio web como penalización o recompensa.

Otro aspecto a considerar es que, como se ha visto en este trabajo, en un contrato para gestionar *QoS*, las propiedades y acciones a realizar en caso que no se cumplan las garantías son definidas por el proveedor de un servicio. Oracle Service Bus al actuar como intermediario; adquiere el rol de proveedor y aunque la herramienta permite establecer alertas SLA sobre los servicios proxy, que son el componente utilizado para la comunicación con el cliente, esta no tiene soporte para formalizar los acuerdos establecidos entre OSB como proveedor de servicio y los clientes de estos, utilizando lenguajes de contrato electrónico. Por último es importante resaltar que las experiencias con respecto al proceso descrito no son muy abarcadoras debido a que, en el entorno en que se aplicó, el conocimiento acerca de la

contratación electrónica para gestionar calidad de servicio es muy poco. Por parte de los proveedores de servicios no existía ningún tipo de conclusión acerca del comportamiento de estos, por lo que, para probar el modelo, se tuvieron que simular, dentro del equipo de trabajo, las partes que intervinieron, las garantías con respecto al comportamiento de los servicios y las acciones acordadas en caso de que las garantías no se cumplan.

## Referencias

- ANDRIKOPOULOS, V. QoS Contract Formation and Evolution. En: Buccafurri, F. y Semeraro, G.E-Commerce and Web Technologies. Berlin: Springer Berlin Heidelberg, 2010, 61, p.119-130.
- CHAPPELL, D. Enterprise Service Bus. Sebastopol, O'Reilly Media Inc, 2004.276
- DÍAZ, M. Comparación de tecnologías de servicios web Axis, Xfire y Jax-WS. En: 15 Convención Científica de Ingeniería y Arquitectura. Memorias de la 15 Convención Científica de Ingeniería y Arquitectura. La Habana: Comité Científico del evento, 2010, 10.
- FERNÁNDEZ, F. Towards a contract based interoperation model. [En línea] 2007. [Consultado el: 12 de mayo de 2014]. Disponible en: [<http://www.lsi.upc.edu/~techreps/files/R07-35.zip>].
- JONES, F. Oracle Fusion Middleware Developer's Guide for Oracle Service Bus. [En línea] 2010. [Consultado el: 12 de diciembre de 2014]. Disponible en:[ [http://docs.oracle.com/cd/E28280\\_01/doc.1111/e15866.pdf](http://docs.oracle.com/cd/E28280_01/doc.1111/e15866.pdf)]
- KELLER, A. The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. [En línea] Journal of Network and Systems Management, 2007. [Consultado el: 10 de marzo de 2014]. Disponible en: [http://clip.dia.fi.upm.es/Projects/S-CUBE/papers/keller03:wsla\\_framework.pdf](http://clip.dia.fi.upm.es/Projects/S-CUBE/papers/keller03:wsla_framework.pdf)
- LUDWING, H. Web Service Level Agreement (WSLA) Language Specification. [En línea] IBM, 2003. [Consultado el: 12 de febrero de 2014]. Disponible en: <http://www.research.ibm.com/people/a/akeller/Data/WSLASpecV1-20030128.pdf>
- MARTINIANO, D. Implementando un Bus de Servicios Empresariales. [En línea] 2009. [Consultado el: 16 de febrero de 2014].
- MEYER, B. Object Oriented Software Construction SECOND EDITION. Santa Bárbara, ISE INC, 1997.1254
- MICROSOFT. MSDN [En línea] 2012. [Consultado el: 15 de febrero de 2014]. Disponible en: [<http://msdn.microsoft.com/es-ES/library/dd257004%28v=bts.10%29.aspx>].
- ORACLE. Concepts and architecture for Oracle Service Bus. [En línea] 2010. [Consultado el: 13 de mayo de 2014]. Disponible en: [[http://docs.oracle.com/cd/E28280\\_01/doc.1111/e15020.pdf](http://docs.oracle.com/cd/E28280_01/doc.1111/e15020.pdf)]
- ROSALES, M. D. Comparación de tecnologías de servicios web Axis, XFire y JAX-WS En: 15 Convención Científica de Ingeniería y Arquitectura. Memorias de la 15 Convención Científica de Ingeniería y Arquitectura. La Habana: Comité Científico del Evento, 2010, 9.
- SCHMUTZ, G. Oracle Service Bus 11g Development Cookbook. Birmingham, Packt Publishing, 2012. 620.
- SCHONBERGER, A. Translating shared state based ebXML BPSS models to WS-BPEL. International Journal of Business Intelligence and Data Mining, 2010, 5(4): 398-442.
- TANG, L. SLA-Aware Enterprise Service Computing. [En línea] 2013. [Consultado el: 16 de mayo de 14]. Disponible en: [<http://www.servicetechmag.com/system/application/views/I75/0813-4.pdf>].
- TOSIC, V. Contract-Based Quality of Service (QoS) Monitoring and Control of XML Web Services. [En línea] 2010. [Consultado el: 14 de mayo de 2014]. Disponible en: [<http://www.nicta.com.au/pub?doc=1628>].

- WAELDRICH, O. From WS-Agreement to SLA negotiation. [En línea] 2008. [Consultado el:13 de mayo de 2014]. Disponible en: [[http://coregrid.ercim.eu/mambo/images/stories/Meetings/wp6\\_march-2008\\_waeldrich.pdf](http://coregrid.ercim.eu/mambo/images/stories/Meetings/wp6_march-2008_waeldrich.pdf)].
- WAELDRICH, O. WS-Agreement Negotiation. [En línea] 2011. [Consultado el: 16 de mayo de 2014]. Disponible en: [<http://www.ogf.org/documents/GFD.107.pdf>].