

# UN MARCO COMPARATIVO PARA LA CLASIFICACIÓN DE PLATAFORMAS DE DESARROLLO DE SOFTWARE BASADO EN SERVICIOS WEB

Gustavo Alejandro Bartoli, Alejandra Cechich

Departamento de Ciencias de la Computación - Universidad Nacional del Comahue  
Buenos Aires 1400, 8300 Neuquén, Argentina  
E-mail: {g\_bartoli@yahoo.com}{acechich@uncoma.edu.ar}

## RESUMEN

Los Servicios Web conforman una tecnología emergente que tiene como objetivo permitir la interoperabilidad e integración de negocios a través de la Web de una forma transparente al usuario e independiente de las plataformas. Se caracterizan por brindar un modelo de computación distribuida y por hacer uso tanto de los estándares de Internet como de su infraestructura. En este artículo, presentamos un marco comparativo que sirve como herramienta para contrastar las características que ofrecen las diferentes plataformas de desarrollo de software basado en esta nueva tecnología. A modo de ejemplo, aplicamos el marco para clasificar tres de las plataformas actualmente más difundidas.

**PALABRAS CLAVES:** Servicios Web – Plataformas de Desarrollo – Calidad del producto software.

## 1. INTRODUCCIÓN

A partir del surgimiento de Internet, y en particular de la WWW han comenzado a nacer nuevas tendencias de mercado, junto con nuevos requerimientos tales como servicios complejos a través de la Web, comercio electrónico, capacidad de integración B2B, etc. Los Servicios Web, buscan aprovechar el modelo de programación Web y dicha estructura para permitir construir servicios a través de aplicaciones distribuidas. El principal objetivo, es el de proveer de una *plataforma* que permita el desarrollo de aplicaciones, las cuales puedan interoperar sin importar el lenguaje de programación en el cual fueron desarrolladas, ni el sistema operativo o el hardware en el cual estén ejecutando, permitiéndose invocar a los servicios de distintos sistemas en forma distribuida y de manera similar a la invocación de una función en un lenguaje tradicional. De esta manera, se busca desacoplar el servicio o aplicación en sí del resto de los componentes (hardware, lenguajes de programación, etc.). La idea de los Servicios Web es pasar a un auténtico entorno de computación distribuida basada en Protocolos Web.

Los Servicios Web permiten la interoperabilidad entre software escritos en diferentes lenguajes de programación, desarrollados por distintos vendedores, o corriendo en diferentes sistemas operativos o plataformas [1]. La multiplicidad de características que actualmente ofrecen los diversos entornos de desarrollo y plataformas comerciales, hacen que la tarea de evaluación y selección de esta nueva tecnología no sea fácil. Con el fin de facilitar su uso y/o adopción, en este artículo describimos un marco comparativo que permite clasificar y valorar alternativas.

Desde un punto de vista de desarrollo de software hemos identificado tres roles principales, que forman la base de nuestro marco comparativo [2][3][4]:

- **Desarrollador de Servicios Web:** Es la entidad encargada de concebir, modelar, desarrollar e implementar el Servicio Web. Está asociada de forma directa con el ciclo de vida del desarrollo del Servicio Web.
- **Proveedor de Servicios Web:** Es la entidad encargada de publicar los Servicios Web y de realizar todas las tareas asociadas al seguimiento y gestión de los mismos.
- **Consumidor de Servicios Web:** Es la entidad que busca, encuentra y hace uso posterior del Servicio Web.

Estos roles constituyen tres puntos de vista o perspectivas desde las que se analizarán las plataformas de desarrollo. Los requerimientos se adecuarán a cada perspectiva – desarrollador, proveedor y consumidor – dando por resultado un *marco comparativo basado en roles*.

Para definir los requerimientos, se ha tomado como base la jerarquía provista por el estándar ISO-9126 [5] para especificación de calidad de productos software. Aún cuando los requerimientos pueden repetirse entre distintas perspectivas, su definición varía de acuerdo a las características de cada rol. Es por eso que el mismo requerimiento puede ser analizado más de una vez dentro del marco propuesto.

En la sección 2 de este artículo, presentamos los elementos que constituyen nuestro marco comparativo desde las tres perspectivas mencionadas previamente. Luego, la sección 3 clasifica tres plataformas de desarrollo de servicios Web ampliamente difundidas. Además, se agrega un valor o puntuación que podría utilizarse en estudios comparativos. Finalmente, la sección 4 enuncia conclusiones y trabajos futuros.

## 2. DEFINICIÓN DEL MARCO COMPARATIVO

Las secciones siguientes presentan los requerimientos que definen el marco comparativo para cada perspectiva.

### 2.1 Marco Comparativo desde la Perspectiva del Desarrollador

- **Soporte para Desarrollo Distribuido:** Se refiere al proceso de realizar proyectos en múltiples centros de desarrollo, con el objeto de disminuir costos o mejorar el tiempo de entrega del producto [6]. La plataforma debe fomentar el desarrollo distribuido de una manera colaborativa y paralela.
- **Manejo de Requerimientos:** Los requerimientos son la piedra angular en el desarrollo de un producto de software. A partir de los requerimientos identificados se definirán restricciones y condiciones que irán dando forma al proyecto. Es de vital importancia disponer de un entorno que permita dar soporte a la identificación, organización, documentación y gestión de cambios de requerimientos.
- **Lenguaje de Modelado de Componentes:** Es importante poseer un método o herramienta de modelado para representar de una manera no ambigua los componentes del Servicio Web y la relación entre los mismos. Se han de permitir tanto una representación lo suficientemente precisa como para poder generar a partir de ella el código asociado a los componentes, así como también una representación lo suficientemente abstracta como para ser independiente de la tecnología de implementación.

- **Lenguajes de Programación:** Flexibilidad que el entorno de desarrollo ofrece respecto a los lenguajes de programación que pueden ser utilizados para implementar los Servicios Web. Ha de brindar al desarrollador la posibilidad de optar dentro de un conjunto de lenguajes por aquel que se ajuste mejor a sus necesidades o aquel al cual éste se encuentre más familiarizado.
- **Soporte para Modelado de Orquestación - Composición - Workflow:** Por medio de la *Orquestación* se describe cómo los Servicios Web pueden interactuar con los demás a nivel de mensaje, teniendo en cuenta la lógica del negocio y el orden de ejecución de las interacciones [7]. En la ejecución de un proceso, transacción o negocio, puede intervenir más de un Servicio Web. La secuencia de mensajes que intervienen en dicha transacción o proceso debe de estar bien definida. A partir del modelo de workflow, se han de poder automatizar las interacciones entre dos o más Servicios Web.
- **Soporte para Modelado de Coreografía:** Por *Coreografía* se entiende a la realización de procesos de negocio de manera colaborativa por medio de la integración de Servicios Web. Se refiere a la descripción tanto de la interacción dinámica como del flujo de mensajes entre Servicios Web en el contexto de un proceso [8]. Tanto la Orquestación como la Coreografía, permiten modelar la manera en que se realizan las invocaciones entre servicios dentro de un proceso para alcanzar una meta.
- **Modelado de Seguridad:** Se ha de permitir establecer un modelo en el que se especifiquen las restricciones y condiciones necesarias para dar seguridad en diferentes niveles o ámbitos. Es necesario el soporte a través de especificaciones preferentemente formales. El modelo de seguridad debe mantener independencia con respecto a las tecnologías de implementación.
- **Disponibilidad de Patrones:** Los patrones juegan un papel importante en el diseño de Servicios Web que deben crearse a partir de un conjunto de componentes para proveerse como un solo Servicio Web. A través del uso de patrones se facilita en gran medida la comprensión del diseño de los servicios, permitiendo la adopción de soluciones bien conocidas.
- **Soporte para Testing:** Se debe poder probar un Servicio Web por medio de herramientas integradas dentro del entorno de desarrollo. Dicha herramienta ha de permitir realizar diversos tipos de tests: generación de carga, monitoreo de performance, testing a nivel de mensajes, de orquestación, etc.
- **Soporte para Reuso:** Es importante poder crear de manera semiautomática a partir de sistemas heredados (procedurales, orientados a objetos, etc) Servicios Web que brinden la funcionalidad para la que originalmente fueron construidos. Tomar componentes existentes (componentes EJB, objetos COM, componentes CORBA, clases JAVA, etc) y “convertirlos” en un Servicio Web, facilita el reuso de código resultando en una disminución de costos en el re-desarrollo de todo/parte de la aplicación.
- **Coexistencia del Entorno de Desarrollo:** Grado en que el entorno de desarrollo de Servicios Web permite una buena coexistencia con la plataforma de software donde ha de

ser utilizado. La independencia de plataformas permite una mejor integración y una mayor flexibilidad a la hora de elegir un entorno de desarrollo.

- **Integrabilidad:** Grado de soporte que los proveedores de entornos de desarrollo ofrecen para facilitar la integración de sistemas y aplicaciones dispares. La integración puede llevarse a cabo dentro del proceso de negocios y hacer uso de las facilidades ofrecidas por los Servicios Web para tal fin. Este aspecto generalmente es cubierto a través de servidores o “brokers” de integración.
- **Apertura:** La apertura que un entorno de desarrollo o una plataforma de software brindan son de una importancia destacable y tienen efectos desde el punto de vista económico (costos). La apertura de una plataforma se puede apreciar en los protocolos (propietarios versus no propietarios), lenguajes utilizados, código abierto, etc.
- **Facilidad de instalación:** Se han de tener en cuenta las facilidades de instalación del entorno de desarrollo que la firma ofrece. Éste se ha de poder instalar de una manera sencilla y clara, seleccionando los componentes a medida. Ha de brindar la posibilidad de ampliar entornos preexistentes por medio de plug-ins, capacidades de personalización y optimización del entorno, etc.

## 2.2 Marco Comparativo desde la Perspectiva del Proveedor

- **Soporte para Gestión de Servicios Web:** Grado de soporte para el descubrimiento de la existencia, disponibilidad, integridad y nivel de uso así como para el control y la configuración de Servicios Web [9]. Los Servicios Web suministrados por un proveedor de servicios, deben de poderse administrar de una manera amigable y sencilla para el proveedor. Entre las tareas que se han de facilitar se encuentran las de administración de usuarios, perfiles, monitoreo de servicios, etc.
- **Facilidad de Publicación:** Para la publicación de servicios es necesario disponer de un repositorio donde se alojará la descripción de los Servicios Web (UDDI Registry). Dicho repositorio será consultado por las aplicaciones de los potenciales usuarios del servicio. Es conveniente que se provea de un método que facilite la publicación de dichos servicios, de una manera sencilla y amigable para el desarrollador.
- **Framework de Ejecución:** Se necesita disponer de un entorno de software que actúe a modo de “Web Service Container” con soporte para Servicios Web. En éste se alojarán y ejecutarán los servicios propiamente dichos (Servidores de Servicios Web). Éstos han de proveer facilidades tales como balance de carga, disponibilidad, “clustering” y “failover” entre otras [10].
- **Soporte para Seguridad:** Se refiere a la protección contra lecturas no autorizadas, modificación o destrucción de la información así como a los problemas relacionados de autenticación y no repudiación. Los aspectos de seguridad de los Servicios Web son uno de los condicionantes más importantes en la adopción de esta nueva tecnología. Se ha de proveer un marco que de soporte a dichos aspectos en diferentes niveles, tales como transporte, red, aplicación, DBMS, etc [11].

- **Soporte para Interoperabilidad:** El Framework de Ejecución ha de permitir un nivel de interoperabilidad con diferentes proveedores de tecnologías. Esta característica facilita a los proveedores que ofrecen Servicios Web ejecutar el framework en diferentes plataformas o entornos, soportar otras implementaciones de protocolos, intercambiar datos a través de herramientas, etc.
- **Soporte para Coreografía/Orquestación/Workflow:** En tiempo de ejecución, los Servicios Web pueden estar formados por otros Servicios Web que brindan su funcionalidad para obtener en forma conjunta una funcionalidad más compleja. Esto da lugar a complejas interacciones entre servicios, las cuales se han de poder especificar a través de protocolos y lenguajes [12].
- **Composición de Servicios:** Una de las características de los Servicios Web es la capacidad de composición: un Servicio Web ha de poder proveer funcionalidad a partir de Servicios Web más simples. La relación de composición, junto con cualquier otra información necesaria para tal propósito se debe definir por medio de lenguajes o protocolos.
- **Adecuación a Estándares:** Compatibilidad o adopción de las últimas versiones de los estándares propuestos para la mensajería, publicación y descripción de Servicios Web (SOAP, UDDI, WSDL, etc). Cumplimiento de las especificaciones definidas para el manejo de interoperabilidad entre Servicios Web implementados por diferentes proveedores (WS-I.org).
- **Descripción de Características de Servicio no Funcionales:** El proveedor del Servicio Web debe disponer de capacidades que le permitan describir todos los aspectos no funcionales que caractericen al Servicio Web en cuestión. Sirve como base para la definición de acuerdos a nivel de servicio y como indicador cualitativo para los potenciales usuarios en el momento de búsqueda y selección de servicios.

## 2.3 Marco Comparativo desde la Perspectiva del Consumidor

- **Posibilidad de especificar Calidad de Servicio (QoS):** Es necesario establecer un mecanismo por medio del cual, el Consumidor de Servicios especifique requerimientos no funcionales, asociados al Servicio Web en cuestión. También son necesarios mecanismos para calificar los Servicios Web que ofrecen la misma funcionalidad pero con distinta calidad. Algunos aspectos no funcionales de importancia son la disponibilidad, latencia, nivel de seguridad, etc [13].
- **Soporte para Mensajería:** La seguridad asociada a la mensajería es uno de los aspectos que mayor atención recibe de parte de los Consumidores. Está motivada por la infraestructura de Internet que es inherentemente insegura, la cual hace susceptible a los mensajes a una gran variedad de ataques contra la confidencialidad, autenticidad, e integridad.
- **Manejo de Adjuntos:** Posibilita al Consumidor intercambiar con sus pares (proveedores, socios, clientes, etc.) no solamente datos XML “planos” sino también datos mutimediales de una forma confiable y segura. Se debe brindar soporte a este requerimiento a través de los protocolos de mensajería [14].

- **Soporte para Transacciones:** Los procesos pueden involucrar una gran cantidad de servicios y actividades que pueden llegar a tener un considerable grado de complejidad. Se debe poseer un framework en el que se permita coordinar y dar soporte a nivel de protocolo a los procesos de transacciones, apuntando a aspectos tales como atomicidad, manejo de estado, etc.
- **Soporte para Dispositivos Móviles:** Brinda la capacidad de que los Servicios Web ejecuten en plataformas ligadas a dispositivos móviles tales como PDAs, teléfonos celulares, etc. La flexibilidad y portabilidad ofrecidos por estos dispositivos hacen de ellos un candidato crucial para dar soporte a los Servicios Web.
- **Soporte para Descubrimiento e Invocación:** La búsqueda que debe hacer el Consumidor puede hacerse exhaustiva si la cantidad de Proveedores y registros de servicios a inspeccionar es considerable. Por otro lado, es necesario proveer de un framework que permita de manera dinámica (en “runtime”) establecer ligaduras o enlaces entre el Cliente y el Servicio que éste necesita.
- **Soporte de Autenticación de Usuarios y Proveedores de Servicios:** Se han de suministrar mecanismos de autenticación de usuarios, para que éstos accedan a la funcionalidad especificada para su perfil. Esta característica le da al Proveedor y al Cliente mayor seguridad de “quién es la otra parte”, para cuestiones tales como las de pago del servicio.
- **Performance:** En ciertos dominios, la performance brindada por un servicio es de crucial importancia, tanto o más que la confiabilidad o la seguridad que éste provee. Por ello, es necesario que se tomen medidas de optimización en los distintos componentes que participan en el suministro del Servicio Web.

### 3. APLICACIÓN DEL MARCO COMPARATIVO A DIFERENTES PLATAFORMAS DE DESARROLLO

En esta sección, clasificaremos tres de las principales plataformas de desarrollo de software ofrecidas por conocidas firmas comerciales, de acuerdo a las características del marco presentado en la sección anterior. El marco cubrirá aspectos de las diferentes plataformas incluyendo los entornos de desarrollo así como también las distintas especificaciones que cada firma incorpora a sus arquitecturas por capas para dar soporte a los Servicios Web.

Se hará uso del método DESMET [15] para puntuar el nivel de soporte que cada plataforma ofrece sobre las característica consideradas. Para ello se define una escala ordinal que abarca las siguientes puntuaciones:

- 0 (*No soportado*),
- 1 (*Mínimamente Soportado*),
- 2 (*Parcialmente Soportado a través de herramientas externas al entorno de desarrollo*),
- 3 (*Parcialmente Soportado por medio de herramientas propias del entorno de desarrollo*),
- 4 (*Completamente Soportado a través de herramientas externas al entorno de desarrollo*),
- 5 (*Completamente Soportado con herramientas propias del entorno*).

Cada característica será clasificada según su nivel de importancia como: Obligatoria (**M**), Altamente Deseable (**HD**), Deseable (**D**) o Alternativa (**N**).

Las tres firmas comerciales que se considerarán son Sun Microsystems® con su entorno de desarrollo Sun One SDK® sobre la plataforma J2EE [16][17], Microsoft® quien ofrece el entorno Visual Studio SDK® corriendo sobre la plataforma .NET® [18][19] y finalmente IBM® quien provee del entorno WebSphere Studio SDK® y la plataforma J2EE [20][21].

En las secciones siguientes, se resume en forma tabular la clasificación desde las tres perspectivas del marco comparativo. La puntuación de acuerdo a la escala ordinal se ha ubicado en cada celda al finalizar la explicación que la fundamenta. El nivel de importancia de la característica se ubica a continuación de su nombre.

### 3.1 Aplicación del Marco desde el punto de vista del Desarrollador de Servicios

	PROVEEDOR DE PLATAFORMA		
	Sun Microsystems®	Microsoft®	IBM®
<b>Soporte para desarrollo distribuido</b>  (D)	Manejo de Versiones entre equipos de desarrollo. (1)	Uso de Plantillas. Permite control de versiones, compartir documentación entre grupos de trabajo. Guías y helps dinámicos para desarrolladores. Cubierto por herramientas externas. (3)	Incluido en el entorno Rational ClearCase® que da soporte para desarrollo en paralelo desde el diseño hasta el testing. ClearCase® da soporte a grupos de desarrollo geográficamente distribuidos, control de versiones, etc. (4)
<b>Manejo de requerimientos</b> (HD)	No incluido en el entorno. (0)	No incluido en el entorno. Sólo a través de herramientas externas. (2)	No incluido en el entorno. Integrable a través de IBM Rational Requisite Pro®. (3)
<b>Lenguaje de Modelado de Componentes</b> (HD)	A través de herramientas externas (Iopsis®, etc). (2)	Incluye herramientas gráficas para modelado UML y para bases de datos (modelado conceptual, lógico y físico). (3)	Propone UML extendido. Herramientas no incluidas en el entorno. Integrable a través de Rational Rose® (herramienta gráfica basada en UML). (2)
<b>Lenguajes de Programación</b> (D)	Basado exclusivamente en Java. (1)	Visual Basic, NET®, C#, C++ y J#, Perl, Cobol, Fortran, Eifel, Mondrian, Pascal, Etc. (4)	Basado exclusivamente en Java. (1)
<b>Soporte para Modelado de Orquestación/ Composición/ Workflow</b> (HD)	Herramienta gráfica incluida en el entorno y herramientas de otros proveedores (Sybase®, Iopsis®, etc.) (3)	Incluye herramienta gráfica para modelar la orquestación y workflow de procesos de negocio. (3)	Incluye herramienta ( <i>choreographer</i> ) para modelado visual de orquestación, composición y workflow de procesos. Ofrece plataforma BPEL4WS para validación y ejecución de procesos. Herramientas de terceros (Ej: BindSystem®). (3)

<b>Soporte para Modelado de Coreografía (HD)</b>	No suministrada por el entorno. Provista por la firma en forma separada. (2)	No provista por el entorno. (1)	Incluida en el entorno ( <i>choreographer</i> ). (3)
<b>Modelado de Seguridad (M)</b>	No se ofrece en el entorno ninguna herramienta. (0)	Programación declarativa basada en WS-Policy y WS-SecurityPolicy (Requiere Kit de herramientas adicionales) (1)	No se ofrece en el entorno ninguna herramienta. (0)
<b>Disponibilidad de Patrones (HD)</b>	Patterns J2EE Java. (2)	Patrones de arquitectura, de diseño y de implementación. (3)	Patterns Java Herramienta separada (Rational Rose Modeller®) para la automatización y asistencia en el uso de patrones. (2)
<b>Soporte para Testing (D)</b>	Prueba de operaciones ofrecidas por el Servicio Web con asistente. Análisis de Performance, Monitoreo, Registro UDDI interno. (1)	Incluye ACT (Application Center Test) para testing de aplicaciones y Servicios Web. Herramientas de terceros (IBM Rational PurifyPlus®). (3)	El entorno incluye soporte mínimo para testing. Ofrece soporte a través de herramientas separadas (IBM Rational Suite TestStudio® e IBM Rational PurifyPlus®). (2)
<b>Soporte para Reuso (HD)</b>	Exposición de Componentes EJB, funciones en C y objetos de datos como Servicios Web. Basada en arquitectura de conectores J2EE JCA. (5)	Componentes COM pueden exponerse como Servicios Web (usa lenguaje propietario Web Services Meta Language). (5)	Exposición de Componentes EJB, funciones en C, procedimientos DB2, etc como Servicios Web. Basado en arquitectura de conectores J2EE JCA. Apoyo de herramientas gráficas. (5)
<b>Coexistencia del Entorno de Desarrollo (D)</b>	Solaris 9 y 8 (Sparc®) Windows® 2000 (Intel®). (3)	Windows® Millennium, Windows® XP y Windows® 2000. Procesadores Intel® – Compatibles. (1)	Windows® NT, Windows® 2000 Windows® XP Linux® (Red Hat® – SuSe®) Procesadores Intel®-Compatibles. (2)
<b>Integrabilidad (HD)</b>	Broker de Integración incluido en el entorno de desarrollo Sun ONE® con capacidad de gestión de procesos. (3)	Host integration Server®, forma parte del entorno (integración a nivel de hosting y de aplicación con sistemas heredados). Provee como herramienta externa el BizTalk® Server (integración de componentes COM basado en estándares). (3)	No incluido en el entorno. A través de WebSphere® Business Integration Server provee herramientas para gestión e integración de procesos, entre ellas el WebSphere® Business Integration Adapters que posee adaptadores para integrar procesos de negocio por medio de Servicios Web. (2)
<b>Apertura (D)</b>	Plataforma abierta (Basado en J2EE) Protocolos y estándares no propietarios. Código abierto. (3)	Protocolos y estándares propietarios. No se basa en código abierto. (1)	Plataforma abierta (Basado en J2EE) Protocolos y estándares propietarios. (2)
<b>Facilidad de instalación (N)</b>	Mediante asistentes. Plug-ins preinstalados facilitan ampliar el entorno de una manera simple. (2)	Por medio de asistente. Instalación selectiva de componentes y librerías. (2)	Por medio de asistente. Selección de componentes a instalar y de la interfaz para control de versiones. (2)

### 3.2 Aplicación del Marco desde el punto de vista del Proveedor de Servicios

	PROVEEDOR DE PLATAFORMA		
	Sun Microsystems®	Microsoft®	IBM®
<b>Soporte para Gestión de Servicios Web</b>  <b>(HD)</b>	Herramientas de Terceros. Parcialmente cubierto a través de servidores incluidos en Sun ONE®. (3)	No ofrece funcionalidades específicas para la gestión de Servicios Web. (0)	Incluido en la pila de protocolos propuesta (WS-Manageability) Soporte en tiempo de <i>deployment</i> limitado. Amplio soporte por herramientas separadas (Tivoli® Application Performance Management y Tivoli® Web Services Manager). (4)
<b>Facilidad de Publicación</b>  <b>(D)</b>	En registro UDDI-ebXML por APIs Java y Asistentes. No es operador UDDI. Ofrece registro UDDI interno. (3)	Operadora de nodos UDDI. Ofrece registro de prueba. Asistentes de Publicación. (3)	Operadora de nodos UDDI. Ofrece registro de prueba. Asistentes de Publicación. (3)
<b>Framework de Ejecución</b>  <b>(M)</b>	Sun ONE® Application Server. Variedad de Servidores de Aplicación J2EE (iPlanet® Application Server, Cape Clear® Server, Systinet® WASP Server, etc). (4)	Servidores de aplicación Microsoft.NET® enterprise server y otros de la firma Microsoft®. (4)	Integrabilidad de servidores de terceros. WebSphere® Application Server Network Deployment ofrece mejores características (No incluido en el entorno ). (4)
<b>Soporte Para Seguridad</b>  <b>(HD)</b>	Motor de Políticas de Seguridad, LDAP, SAML, Seguridad del Servidor de Aplicaciones basada en Roles, Privilegios, Etc. (Especificación J2EE) Variedad de Servidores para Seguridad. (3)	Se basa en la extensión WSE y el servidor BizTalk®. Protocolo propietario WS-Security, Protocolo WS-Policy y otros en desarrollo. (3)	Seguridad del Servidor WebSphere® Studio Application Server acorde a especificaciones J2EE. Protocolos propietarios WS-Security, WS-Policy y otros en desarrollo. (3)
<b>Soporte para Interoperabilidad</b>  <b>(HD)</b>	Interoperabilidad con .NET® y con variedad de Servidores conforme a J2EE de terceros. (4)	Sólo Servidores Microsoft .NET® Servers. Interoperabilidad con SUN® e IBM® por medio de herramientas de terceros. (2)	Interoperabilidad con .NET® Runtime Provee variedad de Servidores de la firma IBM® para tal fin. (3)
<b>Soporte para Coreografía/ Orquestación/ Workflow</b>  <b>(HD)</b>	WSCI (Web Services Choreography Interface) BPML (Business Process Management Language) BPSS (Business Process Specification Schema) para ebXML. (4)	Business Process Execution Language for Web Services (BPEL4WS). (3)	Business Process Execution Language for Web Services (BPEL4WS) Herramienta WebSphere® MQ Workflow (no incluida en el entorno) da apoyo al diseño y control de workflow. (3)

<b>Composición de Servicios</b>  (M)	WS-CAF (Web Services Composite Applications Framework). (3)	No ha definido especificación . Soportado parcialmente por BPEL4WS, WS-Coordination y WS-Transaction. (2)	No ha definido especificación . Soportado parcialmente por BPEL4WS, WS-Coordination y WS-Transaction. (2)
<b>Adecuación a Estándares</b>  (M)	Cumple con WS-I Basic Profile y ebXML. (5)	Cumple con WS-I Basic Profile 1.0. (5)	Cumple con WS-I Basic Profile 1.0. (5)
<b>Descripción de Características de Servicio no Funcionales</b>  (D)	No ha definido especificación. Parcialmente soportado por CPP y CPA del entorno ebXML. (1)	No soportado. (0)	Contemplado en la arquitectura. Especificación aún en desarrollo. (0)

### 3.3 Aplicación del Framework desde el punto de vista del Proveedor de Servicios

	PROVEEDOR DE PLATAFORMA		
	Sun Microsystems®	Microsoft®	IBM®
<b>Posibilidad de especificar Calidad de Servicio</b>  (D)	No ha definido lenguaje/protocolo Sólo a través de CPP (Collaboration Protocol Profiles) y CPA (Collaboration Protocol Agreement) de ebXML.(2)	No ha definido especificación. (0)	Por medio de WSLA (Web Services Level Agreement) permite establecer acuerdos , su control y monitoreo de los mismos. (3)
<b>Soporte para transacciones</b>  (M)	Por medio de la definición Web Services Transaction Management (WS-TXM). (2)	WS-Transaction conjuntamente con WS-Coordination y WS-AtomicTransaction. (2)	WS-Transaction conjuntamente con WS-Coordination y WS-AtomicTransaction. (2)
<b>Soporte para Mensajería</b>  (M)	WS-Reliability 1.0 ebXML Messaging Service (ebMS) JMS (Java Message Service). (3)	WS-Reliablemessaging (Necesita instalación de WSE) WS-SecureConversation (Posible adopción en 2005). (3)	WS-Reliablemessaging. IBM® no detalla sobre su implementación ni el soporte de esta especificación en las herramientas o servicios ofrecidos por la firma comercial WS-SecureConversation (Posible adopción en 2005). (1)

<b>Manejo de Adjuntos</b> <i>(HD)</i>	A través del uso del API SAAJ (SOAP with Attachments API for Java) y ebMS del contexto ebXML. (5)	Por medio de WS-Attachments. (5)	Por medio de WS-Attachments. (5)
<b>Soporte para Dispositivos Móviles</b> <i>(D)</i>	No incluido en el entorno de desarrollo, pero sí provee como IDE separada el Sun ONE Studio, Mobile Edition®. (2)	Incluido en el entorno. (3)	No incluido en el entorno de desarrollo. Provisto a través de WebSphere® Studio Device Developer. (2)
<b>Soporte para Descubrimiento – Invocación</b> <i>(D)</i>	Registros UDDI y ebXML Basado sólo en el uso de SOAP Herramientas para búsqueda de servicios incluidas en el entorno (Ej: navegador de registros ebXML). No es operador de nodos UDDI. (2)	Registros UDDI. WS-Inspection (Inspección y búsqueda de Servicios). Navegador en su nodo de Registro UDDI. (3)	Incluye un navegador para búsqueda en registros UDDI y documentos WSIL en el entorno. API Java para WS-Inspection. Ligadura dinámica de servicios a través de WSIF. Registros UDDI. WS-Inspection (Inspección y búsqueda de Servicios). Navegador en su nodo de Registro UDDI. (4)
<b>Soporte de Autenticación de Usuarios/Proveedores de Servicios Web</b> <i>(M)</i>	No ha definido especificaciones para autenticación de Usuarios/Proveedores. (1)	Se basa en el uso de WS-Security (autenticación de mensajes SOAP, uso certificados X.509) Necesita la instalación de WSE. (3)	Se basa en el uso de WS-Security. (3)
<b>Performance</b> <i>(D)</i>	Ejecución de aplicaciones y Servicios Web no optimizada para una plataforma particular. Trabajo en desarrollo para mejora de transmisión de mensajes. (1)	Plataforma optimizada al Sistema Operativo Windows®. (2)	Ejecución de aplicaciones y Servicios Web no optimizada para una plataforma particular. (1)

## 4. CONCLUSIONES

En este artículo, se han mostrado de una manera sintética, cuáles son las características y requerimientos que proponemos considerar a la hora de clasificar y/o adoptar una plataforma de software para desarrollar Servicios Web.

El marco comparativo ha sido utilizado a modo de ejemplo, permitiendo observar que las ventajas ofrecidas por la firma SUN Microsystems® están relacionadas a la Apertura, la Coexistencia de Entorno de Desarrollo, el Soporte para la Interoperabilidad, para la Coreografía y Orquestación, Composición de Servicios Web y finalmente la Descripción de Características no Funcionales. Por el lado de Microsoft®, se destacan los aspectos de Lenguaje para Modelado de Componentes, Lenguajes de Programación, Disponibilidad de Patrones, Soporte para Testing, Soporte para Dispositivos Móviles y Performance. Por último, IBM® muestra ventajas en aspectos de Soporte para Desarrollo Distribuido, Manejo de Requerimientos, Soporte para Modelado de Coreografía,

Gestión de Servicios Web, Posibilidad de Especificar Calidad de Servicios y Soporte para Descubrimiento e Invocación de Servicios Web.

Adicionalmente, el marco nos ha permitido identificar puntos que necesitan mayor investigación y/o desarrollo. Por ejemplo, uno de los aspectos a tener en cuenta es la relativa falta de madurez de las especificaciones y estándares para soportar aspectos de Seguridad o Descripción de Características no Funcionales.

Sin embargo, hay algunas limitaciones que debemos resaltar: para algunas de las características del marco comparativo se ha podido recabar suficiente información, mientras que para otras la información dada por las firmas comerciales es muy limitada. Cabe mencionar que como consecuencia de esto es necesario realizar un análisis más profundo en algunos de los puntos definidos tales como *Integrabilidad*, *Soporte para Seguridad* y *Soporte para Interoperabilidad*.

## REFERENCIAS

- [1] D. Johnson. An Introduction to Web Services Enabled with PHP. *NuSphere Co. White Paper*, 2002.
- [2] Stevens M. Service-Oriented Architecture. *Java Web Services Architecture*, 2003.
- [3] Champion M., Ferris C., Newcomer E., Orchard D. Web Services Architecture. *W3C Working Draft*, 2003.
- [4] Jaideep R. and Anupama R. Understanding Web Services. *IEEE Computer Society. IT Pro*, 2002.
- [5] ISO/IEC. Software Product Evaluation – Quality Characteristics and Guidelines for their Use. International Standard ISO/IEC 9126, International Standards Organization, Geneva, Switzerland, 2001.
- [6] Fukushige S. and Trainor H. A Primer on Distributed Development Processes. *Symbio White Paper*, 2002.
- [7] Peltz C. Web Services Orchestration: A review of emerging technologies, tools, and standards. *Hewlett Packard, Co. Paper*, 2003.
- [8] Yendluri P. Web Services Choreography. *webMethods White Paper*, 2003.
- [9] Sun Microsystems. Web Services Life Cycle: Managing Enterprise Web Services – *White Paper*, 2003.
- [10] Goetz B. The Web Services Tools Landscape Part 3 – Deployment Tools. *Intel Technical Article*, 2003.
- [11] DataPower Technology Inc. Essential XML Web Services Security Best Practices. *Technical Whitepaper*, 2003.
- [12] O’Riordan D. Business Process Standards For Web Services. *Tect White Paper*, 2002
- [13] Mani A. and Nagarajan A. Understanding quality of service for Web services. *IBM Technical Article*, 2002.
- [14] Bosworth A., D, Box D., Gudgin M., Nottingham M., Orchard D., Schlimmer J. XML, SOAP and Binary Data. *XML.COM White Paper*, 2003.
- [15] Kitchenham B. DESMET: A method for evaluating Software Engineering Methods and Tools. *Technical Report TR96-09*, 1996.
- [16] Sun Microsystems. Delivering Services on Demand. *Sun ONE Architecture Guide* <http://www.sun.com/software/sunone/docs/arch/>.
- [17] Mahmoud Q. Deploying Web Services on J2EE. *Sun Microsystems Technical Article*. <http://java.sun.com/developer/technicalArticles/WebServices/wsJ2ee>, 2002.
- [18] Microsoft MSDN Web Site. Product Overview for Visual Studio .NET 2003 Enterprise Architect. <http://msdn.microsoft.com/vstudio/productinfo/overview/eaoverview.aspx>.
- [19] Saunders K. Servicios Web .NET. *Servicios Web XML*. Anaya Multimedia/Wrox., 2002
- [20] Flurry G. Support for J2EE Web Services in WebSphere Studio Application Developer V5.1 -- Part 1: The Server Environment. *IBM Technical Article*. [http://www-106.ibm.com/developerworks/websphere/library/techarticles/0310\\_flurry/flurry1.html](http://www-106.ibm.com/developerworks/websphere/library/techarticles/0310_flurry/flurry1.html).
- [21] Knutson J. and Kreger H. Web Services for J2EE, Version 1.0. *IBM Proposed Final Draft v 0.9*, 2002.