

Universidad de las Ciencias informáticas

Facultad 6



Título: “Procedimiento para la integración de sistemas legados en organizaciones empresariales con arquitecturas orientadas a servicios”

Trabajo de diploma para optar por el título de ingeniero en ciencias informáticas

Autor (es): José Andrés Cuan Céspedes

Geovanis Manzano Chacón

Tutor (es): Ing. Marbys Marante Valdivia

Ing. Iskael Diaz Marquez

Co-Tutor: Ing. Asnay Guirola González

Ciudad de la Habana, Junio de 2010

DECLARACIÓN DE AUTORÍA

Declaro que somos los únicos autores de este trabajo y autorizo a la Facultad 6 y a la Universidad de las Ciencias Informáticas para que haga el uso que estime pertinente con el mismo, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ de 2010.

Firma del Autor

José Andrés Cuan Céspedes

Firma del Autor

Geovanis Manzano Chacón

Firma del Tutor

Ing. Marbys Marante Valdivia

Firma del Tutor

Ing. Iskael Diaz Marquez

Firma del Co-Tutor

Ing. Asnay Guirola González

DATOS DE CONTACTO

Tutor:

Ing. Marbys Marante Valdivia.

- Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas 2005
- Se ha desempeñado en el campo del Desarrollo de Software y la Gestión de Proyectos.
- Profesor Instructor

E-mail: mmarante@uci.cu

Tutor:

Ing. Iskael Diaz Marquez

- Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas 2008.
- Se ha desempeñado en el campo del Desarrollo de Software.
- Profesor Adiestrado.

E-mail: imarquez@uci.cu

Co – Tutor:

Ing. Asnay Guirola González.

- Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas 2005.
- Se ha desempeñado en el campo del Desarrollo de Software, la Gestión de Proyectos y como Profesor.
- Profesor Instructor.

E-mail: aguirola@uci.cu

RESUMEN

La Arquitectura Orientada a Servicios (SOA) establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios. La forma más habitual de implementarla es mediante Servicios Web, una tecnología basada en estándares e independiente de la plataforma, con la que SOA puede descomponer aplicaciones monolíticas en un conjunto de servicios e implementar esta funcionalidad en forma modular.

Un servicio es una funcionalidad concreta que puede ser descubierta en la red y que describe tanto lo que puede hacer como el modo de interactuar con ella.

La adopción de una solución de diseño basada en SOA no exige implantar servicios Web. No obstante los servicios Web son la forma más habitual de implementar SOA. Los servicios Web son aplicaciones que utilizan estándares para el transporte, codificación y protocolo de intercambio de información. Los servicios Web permiten la intercomunicación entre sistemas de cualquier plataforma y se utilizan en una gran variedad de escenarios de integración, tanto dentro de las organizaciones como con socios de negocios.

Debido al dominio que tienen las empresas privadas del sector SOA, se hace difícil utilizar y aplicar de manera gratuita sus soluciones. Utilizando como base esta problemática, es que tiene lugar el desarrollo de la presente investigación, la cual tiene como objetivo desarrollar un procedimiento de integración de aplicaciones legadas con una arquitectura orientada a servicios.

Los elementos fundamentales de la propuesta son los artefactos, roles, técnicas y herramientas, subprocesos y actividades que conforman el flujo de trabajo a seguir durante la aplicación del procedimiento para la integración de sistemas legados en organizaciones empresariales con arquitecturas orientadas a servicios. Para validar la propuesta fue utilizado el Método Delphi, a través del cual se utilizaron como elementos de certificación los criterios aportados por un panel de expertos en el tema.

Palabras Claves

Arquitectura Orientada a Servicios, Servicio, Servicio Web, WSDL, procedimiento.

Tabla de contenido

INTRODUCCIÓN	1
CAPÍTULO 1: MARCO TEÓRICO	5
INTRODUCCIÓN	5
1.1 ESTADO DEL ARTE.....	5
1.2 ARQUITECTURA DE SOFTWARE.....	7
1.2.1 ARQUITECTURAS MÁS COMUNES	7
1.2.2 OTRAS ARQUITECTURAS MENOS CONOCIDAS.....	8
1.3 ARQUITECTURA ORIENTADA A SERVICIOS (SOA)	8
1.3.1 ¿QUÉ ES UN SERVICIO?	9
1.3.2 SERVICIOS WEB	9
1.3.3 SERVICIOS WEB REST.....	9
1.3.4 SOA. RETOS	10
1.3.5 SOA. SOLUCIONES	11
1.3.6 ¿QUÉ ES UN WSDL?	11
1.3.6.1 Elementos que conforman un WSDL	11
1.4 LENGUAJES DE PROGRAMACIÓN IDENTIFICADOS PARA PROPONER EL PROCEDIMIENTO	12
1.4.1 JAVA.....	12
1.4.1.1 La plataforma Java	13
1.4.1.2 Máquina virtual de Java	13
1.4.2 JAVASERVER PAGES	14
1.4.2.1 HTML	14
1.4.2.2 XML	14
1.4.2.3 Script	14

1.4.3 .NET	15
1.4.3.1 Lenguajes de programación desarrollados para el marco de trabajo .NET. C Sharp (C#)	15
1.5.4 PHP	16
CONCLUSIONES PARCIALES.....	16
CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA	18
INTRODUCCIÓN	18
2.1 ESTRUCTURA DEL PROCEDIMIENTO	18
2.2 ALCANCE DEL PROCEDIMIENTO	18
2.3 PRINCIPIOS DEL PROCEDIMIENTO	19
2.4 PREMISAS PARA LA APLICACIÓN DEL PROCEDIMIENTO	20
2.5 ROLES QUE INTERACTÚAN CON EL PROCEDIMIENTO	21
2.5.1 ARQUITECTO SOA.....	21
2.5.1.1 Artefactos que realiza el rol Arquitecto SOA	22
2.5.2 DESARROLLADOR	22
2.5.2.1 Artefactos que realiza el rol Desarrollador.....	22
2.5.3 IMPLEMENTADOR.....	22
2.5.3.1 Artefactos que realiza el rol Implementador.....	22
2.6 REPRESENTACIÓN DEL PROCEDIMIENTO	22
2.7 DESCRIPCIÓN DEL PROCEDIMIENTO	24
2.7.1 OBTENER EL CÓDIGO FUENTE	25
2.7.2 ELABORAR EL PLAN DE TRANSICIÓN	26
2.7.3 DEFINIR FUNCIONALIDADES A EXPORTAR	27
2.7.4 EXPORTACIÓN DE SERVICIOS EN C#	28
2.7.4.1 Analizar el código fuente.....	30

2.7.4.2 Organizar funcionalidades en una clase.....	31
2.7.4.3 Generar el WSDL	32
2.7.4.3 Exportar servicios	33
2.7.5 EXPORTACIÓN DE SERVICIOS EN JAVA	34
2.7.5.1 Analizar el código fuente.....	36
2.7.5.2 Organizar funcionalidades en una clase.....	37
2.7.5.3 Generar el WSDL	38
2.7.5.3 Exportar servicios	39
2.7.5 EXPORTACIÓN DE SERVICIOS EN PHP.....	40
2.7.5.1 Analizar el código fuente.....	42
2.7.5.2 Organizar funcionalidades en una clase.....	43
2.7.5.3 Generar el WSDL	44
2.7.5.4 Exportar servicios	46
2.7.6 PROBAR LOS SERVICIOS.....	46
CONCLUSIONES PARCIALES.....	48
CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA.....	49
INTRODUCCIÓN	49
3.1 LOS MÉTODOS DE EXPERTOS	49
3.1.1 VENTAJAS DE LOS MÉTODOS DE EXPERTOS.....	49
3.1.2 DESVENTAJAS DE LOS MÉTODOS DE EXPERTOS.....	49
3.2 EL MÉTODO DELPHI	50
3.2.1 CARACTERÍSTICAS DEL MÉTODO DELPHI.....	51
3.3 APLICACIÓN DEL MÉTODO	52
3.3.1 ELECCIÓN DE LOS EXPERTOS.....	52

3.3.1.1 Cálculo del coeficiente de competencia	53
3.3.2 Elaboración del cuestionario, para validación de la propuesta.....	54
3.3.3 Establecimiento de la concordancia de los expertos mediante el uso del coeficiente de Kendall.....	54
3.3.4 Desarrollo práctico y explotación de resultados	56
CONCLUSIONES PARCIALES.....	59
CONCLUSIONES GENERALES	60
RECOMENDACIONES	61
REFERENCIAS BIBLIOGRÁFICAS	62
BIBLIOGRAFÍA.....	65
ANEXOS	68
GLOSARIO DE TÉRMINOS.....	71

Tabla de Figuras

FIGURA 1: EMPRESAS LÍDERES EN EL MERCADO SOA	7
FIGURA 2: ELEMENTOS QUE FORMAN PARTE DE LA PLATAFORMA JAVA	13
FIGURA 3: PRINCIPIOS DEL PROCEDIMIENTO.....	20
FIGURA 4: PREMISAS PARA LA APLICACIÓN DEL PROCEDIMIENTO	21
FIGURA 5: REPRESENTACIÓN GENERAL DEL PROCEDIMIENTO DE EXPORTACIÓN DE SERVICIOS.....	23
FIGURA 6: FLUJO DE ACTIVIDADES DEL SUBPROCESO EXPORTACIÓN DE SERVICIOS EN C#.....	30
FIGURA 7: FLUJO DE ACTIVIDADES DEL SUBPROCESO EXPORTACIÓN DE SERVICIOS EN JAVA.....	36
FIGURA 8: FLUJO DE ACTIVIDADES DEL SUBPROCESO EXPORTACIÓN DE SERVICIOS EN PHP	42
FIGURA 9: PETICIÓN EN LA HERRAMIENTA SOAPUI	48
FIGURA 10: FRECUENCIAS ACUMULADAS	57
FIGURA 11: PUNTOS DE CORTE.....	58
FIGURA 12: RESULTADOS FINALES DE LA VALIDACIÓN	59

Introducción

Las empresas necesitan poder interconectar los procesos, personas e información tanto con la propia organización como atravesando sus fronteras. La falta de integración entre los componentes de las Tecnologías de Información (TI) –sistemas, aplicaciones y datos– hace difícil obtener una respuesta rápida y efectiva ante los cambios que afectan de forma natural a los negocios. La inflexibilidad genera costes, reduce la capacidad de respuesta ante los clientes, compromete el cumplimiento con las normativas legales y afecta negativamente a la productividad de los empleados. Es por ello que una deficiente integración es uno de los problemas más importantes a los que las organizaciones deben hacer frente para mantener su competitividad y garantizar su crecimiento.

Desde los últimos años del siglo XX y principios del presente han irrumpido en la escena mundial las Arquitecturas Orientadas a Servicios convirtiéndose en una de las tecnologías más populares de esta década.

La Arquitectura Orientada a Servicios (SOA) supone una estrategia general de organización de los elementos de las TI, de forma que una colección abigarrada de sistemas distribuidos y aplicaciones complejas se pueda transformar en una red de recursos integrados, simplificada y sumamente flexible. Un proyecto SOA bien ejecutado permite alinear los recursos de las TI de forma más directa con los objetivos de negocio, ganando así un mayor grado de integración con clientes y proveedores, proporcionando una inteligencia de negocio más precisa y más accesible con la cual se podrán adoptar mejores decisiones, y ayuda a las empresas a optimizar sus procesos internos y sus flujos de información para mejorar la productividad individual. El resultado neto es un aumento muy notable de la agilidad de la organización.

La Arquitectura SOA establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios.

[1]

Un servicio es una funcionalidad concreta que puede ser descubierta en la red y que describe tanto lo que puede hacer como el modo de interactuar con ella. Los servicios también pueden acoplarse dentro de una aplicación completa que proporcione servicios de alto nivel, con un grado de complejidad muy superior que puede involucrar varias aplicaciones de negocio.

La meta de la SOA es lograr cambios enfocados en el negocio rápidamente con un menor costo y lograr diferencias competitivas. Cada empresa tiene sus particularidades y prioridades, por lo que es importante fijar una estrategia de adopción de una SOA desde el principio y posteriormente definir proyectos individuales que permitan ser implementados en forma incremental, logrando resultados exitosos en poco tiempo.

Cuba no está ajena a las tendencias que brindan facilidades para la competencia entre las empresas y el desarrollo de las mismas en todos los sentidos. Es por ello que en el año 2002 se crea la Universidad de las Ciencias Informáticas (UCI) con el objetivo de lograr una vinculación entre la universidad y las empresas, introduciendo el concepto de Universidad Productiva. La UCI tiene la producción como un problema económico, político y social en cual incluye la participación de la mayoría de sus integrantes. Esto proporcionaría una mejora para la economía del país y un avance para la industria del software.

El surgimiento de la UCI provocó en Cuba una explosión en el campo de la informática. Comienzan a realizarse software para todas las necesidades, no solo del país, sino de otros países. Esto ha traído consigo grandes beneficios económicos porque muchas empresas han sido informatizadas tanto en Cuba como en otras partes del mundo.

A partir de la necesidad del país y más concretamente de la UCI de insertarse en el mercado con productos y servicios de software de alta calidad, debido a la novedad de la rama BPM/SOA en el campo de la informática, el hecho de que las empresas líderes poseedoras de los conocimientos para enfrentar proyectos de este tipo cobran por sus servicios altas sumas de dinero y tienen un dominio privado en el sector de BPM/SOA se hace difícil utilizar y aplicar de manera gratuita esta arquitectura. Es por ello que surge la idea de crear en la UCI el Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE) el cual poseería el personal capacitado para dar respuesta a cualquier situación. Este centro le está dando una principal atención a los proyectos BPM/SOA por su creciente aceptación mundial y su comprobada eficiencia.

En la actualidad muchas empresas cuentan con sistemas informáticos complejos, independientes, heterogéneos y muchas veces redundantes. Estos sistemas informáticos tienen varias aplicaciones hechas en distintas plataformas y estándares. En ocasiones algunos de los sistemas han perdido su valor pues con el transcurso del tiempo los procesos de negocio y la lógica de negocio embebida en ellos ha cambiado y no han podido adaptarse a ese cambio, por lo que el personal que los opera, no utiliza todas las funcionalidades para los que fueron concebidos inicialmente, y sin dudas los sistemas en ese

momento comienza a devaluarse, muchas empresas ofrecen cierto rechazo a la migración total de un sistema a otro aunque sean más rápidos y más eficientes, pues esa migración lleva consigo un alto costo.

Otra situación adversa son las políticas de uso de las aplicaciones existentes en las organizaciones que desean adoptar SOA, en muchos casos no brindan el código fuente, y en otros, temen perder la estabilidad de su funcionamiento, por otra parte a los arquitectos SOA les conviene aprovechar las funcionalidades ya desarrolladas y garantizar de igual modo la estabilidad y complacencia del cliente. Es por ello que surge la necesidad de crear una red de recursos interoperables, es decir, unir en una sola capa de arquitectura todos los servicios que van a ser usados del sistema de información empresarial.

Por todo lo antes expuesto surge como **problema científico**: ¿Cómo lograr la integración entre sistemas legados existentes en las organizaciones con una arquitectura orientada a servicio?, dando lugar al **objeto de estudio**: Proceso de integración de sistemas legados, en el cual se enmarca el **campo de acción**: Proceso de migración o transición de sistemas legados con una Arquitectura Orientada a Servicios. Por lo que se plantea como **objetivo general**: Desarrollar un procedimiento de integración de aplicaciones legadas con una arquitectura orientada a servicios. En correspondencia con el mismo se derivan los siguientes **objetivos específicos**:

- ✓ Realizar una investigación de los procedimientos de integración existentes.
- ✓ Desarrollar un procedimiento de integración de aplicaciones legadas con una arquitectura orientada a servicio para el lenguaje de programación Java.
- ✓ Desarrollar un procedimiento de integración de aplicaciones legadas con una arquitectura orientada a servicio para el lenguaje de programación C#.
- ✓ Desarrollar un procedimiento de integración de aplicaciones legadas con una arquitectura orientada a servicio para el lenguaje de programación PHP.
- ✓ Validar el procedimiento propuesto.

Para dar cumplimiento a estos objetivos fueron trazadas las siguientes **tareas**:

- ✓ Búsqueda de las tecnologías existentes en el entorno empresarial a acatar por la UCI en los próximos años.
- ✓ Descripción de perfiles asociados a esas tecnologías.
- ✓ Estudio de procedimientos de exportación de funcionalidades que poseen esas tecnologías.

- ✓ Proposición de procedimientos para una adecuada integración de los sistemas legados con SOA, con el fin de desarrollar mediante estos las capas subyacentes (o de aplicaciones) de la SOA.
- ✓ Validación de la propuesta mediante el método Delphi.

De la presente investigación, se esperan como **resultados**:

Documentación de las tecnologías más usadas en los sistemas de información empresariales en los que la UCI o la industria de software cubana tiene como objetivo público o campo de acción.

Procedimiento de exportación como servicio que presentan las tecnologías seleccionadas.

Conjunto de técnicas o mecanismos a utilizar para integrar sistemas legados encontrados en los diagnósticos en las organizaciones con arquitecturas orientadas a servicios desarrolladas para las mismas.

El presente Trabajo de Diploma posee la siguiente **estructura**:

Capítulo 1:

En este capítulo se analizan los principales conceptos y definiciones asociados al dominio del problema que sirven de apoyo durante el desarrollo de la investigación. Como aspecto medular se ofrece un enfoque de los términos fundamentales relacionados con las Arquitecturas Orientadas a Servicios (SOA). Además se realiza un estudio del arte de la rama SOA.

Capítulo 2:

El presente capítulo está dedicado a describir y explicar el procedimiento para la integración de sistemas legados en organizaciones empresariales con una arquitectura orientada a servicios, durante la fase de implementación en un proyecto SOA. Lo cual es la solución resultante de la investigación; se definen los artefactos involucrados, los roles que intervienen y las responsabilidades asociadas a los mismos. La solución se basa en un procedimiento que puede ser usado en empresas que vayan a adoptar SOA.

Capítulo 3:

En este capítulo se valida el procedimiento propuesto en el capítulo anterior mediante el Método Delphi. Para ello se decidió conformar un Panel de Expertos que emitiera su criterio acerca de la adecuación del procedimiento.

CAPÍTULO 1: MARCO TEÓRICO

Introducción

En este capítulo se analizan los principales conceptos y definiciones asociados al dominio del problema que sirven de apoyo durante el desarrollo de la investigación. Como aspecto medular se ofrece un enfoque de los términos fundamentales relacionados con las Arquitecturas Orientadas a Servicios (SOA). Además se realiza un estudio del arte de la rama SOA.

1.1 Estado del arte

Una de las necesidades actuales de las empresas es poder interconectar los procesos, personas e información tanto con la propia organización como atravesando sus fronteras. Es por ello que desde los últimos años del siglo pasado y principios del presente han irrumpido en la escena mundial las Arquitecturas Orientadas a Servicios convirtiéndose en una de las tecnologías más populares de esta década.

En el mundo existen organizaciones que están liderando el mercado de SOA, de las cuales se realiza un breve estudio:

IBM

International Business Machines (IBM), también conocida coloquialmente como **el Gigante Azul**, es una empresa multinacional que fabrica y comercializa herramientas, programas y servicios relacionados con la informática. [2]

En estos momentos IBM es la organización líder de SOA. Este liderazgo se debe a que la estrategia SOA de IBM no sólo respalda los estándares abiertos de servicios Web, sino que además brinda soporte a organizaciones de todos los tamaños, durante todas las etapas de implementación de una SOA, desde la evaluación inicial hasta el despliegue y la gestión. [3]

Microsoft Corporation

Microsoft Corporation es una empresa multinacional estadounidense dedicada al sector de la informática. Microsoft desarrolla, fabrica, licencia y produce software y equipos electrónicos. [4]

Microsoft mantiene su compromiso de fomentar el desarrollo de un rico ecosistema para la creación y gestión de sistemas interconectados. Microsoft ha realizado cuantiosas inversiones en servicios Web, basando por completo su plataforma de desarrollo de última generación en los servicios Web con Microsoft .NET.

Para la integración y orquestación de procesos Microsoft dispone de tecnologías habilitadoras para el diseño, automatización, gestión y optimización de los procesos de negocio de una forma flexible. Microsoft ha realizado grandes inversiones en su estrategia de “Procesos preparados para las personas”, a fin de permitir a las personas dirigir y controlar los resultados de negocio haciéndoles capaces de generar impactos positivos directamente sobre aquellos procesos críticos que hacen progresar a las empresas. [5]

Oracle SOA Suite

Oracle SOA Suite incluye un conjunto completo de componentes de infraestructura de servicios para crear, implementar y administrar arquitecturas orientadas a servicios. Oracle SOA Suite permite que los servicios se creen, administren y diseñen en aplicaciones compuestas y procesos comerciales. Con los componentes de funcionamiento permanente de Oracle SOA Suite, las organizaciones pueden extender y desarrollar fácilmente sus arquitecturas en lugar de reemplazar las inversiones existentes. [6]

A partir del estudio realizado de las empresas líderes en el mercado de SOA se puede arribar a la conclusión de que dicho mercado está ampliamente liderado por empresas desarrolladoras de software propietario las cuales poseen la mayor cantidad de información, experiencia, metodologías y las técnicas de soporte de las arquitecturas orientadas a servicios. También existen compañías en el campo del Open Source las cuales poseen un excelente nivel y que poco a poco van alcanzando un prestigio en este campo.

El Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE), el cual le está dando una principal atención a los proyectos BPM/SOA por su creciente aceptación mundial y su comprobada eficiencia, ya cuenta con varios trabajos aprobados que pueden servir de guía para los trabajadores del CDAE a la hora de implantar SOA en cualquier empresa que esté interesada en este servicio, pero no cuenta con una guía que encierre un procedimiento para la integración de sistemas legados en organizaciones empresariales con arquitecturas orientadas a servicios. Esta guía sería de gran

importancia para los desarrolladores del centro pues los proveería de técnicas de integración de sistemas probadas en distintos lenguajes de programación. Esto reduciría grandemente el tiempo de desarrollo de los proyectos, debido a que los desarrolladores del centro dedican gran cantidad de tiempo buscando técnicas y mecanismos de integración en diferentes bibliografías y en Internet.

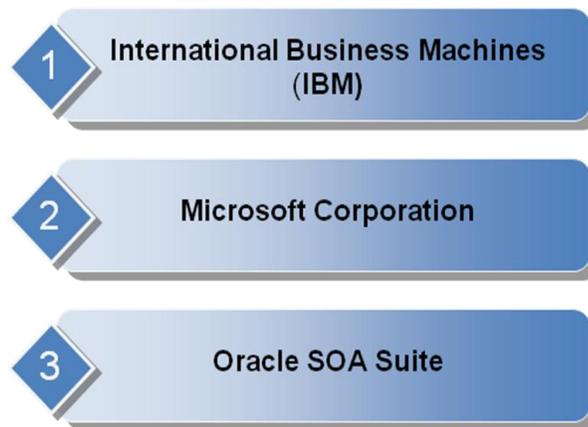


Figura 1: Empresas líderes en el mercado SOA

1.2 Arquitectura de software

La definición oficial de Arquitectura del Software es la IEEE Std 1471-2000 que dice: “La Arquitectura del Software es la organización fundamental de un sistema formada por sus componentes, las relaciones entre ellos y el contexto en el que se implantarán, y los principios que orientan su diseño y evolución”. [7]

1.2.1 Arquitecturas más comunes

Generalmente, no es necesario inventar una nueva arquitectura de software para cada sistema de información. Lo habitual es adoptar una arquitectura conocida en función de sus ventajas e inconvenientes para cada caso en concreto. Las arquitecturas más universales son:

- **Monolítica.** Donde el software se estructura en grupos funcionales muy acoplados.
- **Cliente-servidor.** Donde el software reparte su carga de cómputo en dos partes independientes pero sin reparto claro de funciones.

- **Arquitectura de tres niveles.** Especialización de la arquitectura cliente-servidor donde la carga se divide en tres partes (o capas) con un reparto claro de funciones: una capa para la presentación (interfaz de usuario), otra para el cálculo (donde se encuentra modelado el negocio) y otra para el almacenamiento (persistencia). Una capa solamente tiene relación con la siguiente.

1.2.2 Otras arquitecturas menos conocidas

- En pipeline.
- Entre pares.
- En pizarra.
- Orientada a servicios.
- Máquinas virtuales.

1.3 Arquitectura Orientada a Servicios (SOA)

La Arquitectura SOA establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios. La forma más habitual de implementarla es mediante Servicios Web, una tecnología basada en estándares e independiente de la plataforma, con la que SOA puede descomponer aplicaciones monolíticas en un conjunto de servicios e implementar esta funcionalidad en forma modular. [8]

La estrategia de orientación a servicios permite la creación de servicios y aplicaciones compuestas que pueden existir con independencia de las tecnologías subyacentes. En lugar de exigir que todos los datos y lógica de negocio residan en un mismo ordenador, el modelo de servicios facilita el acceso y consumo de los recursos de TI a través de la red. Puesto que los servicios están diseñados para ser independientes, autónomos y para interconectarse adecuadamente, pueden combinarse con suma facilidad en aplicaciones complejas que respondan a las necesidades de cada momento en el seno de una organización. Las aplicaciones compuestas (también llamadas “dinámicas”) permiten a las empresas mejorar y automatizar sus procesos manuales, disponer de una visión consistente de sus clientes y socios comerciales y orquestrar sus procesos de negocio para que cumplan con las regulaciones legales y políticas internas. El resultado final es que las organizaciones que adoptan la orientación a servicios

pueden crear y reutilizar servicios y aplicaciones y adaptarlos ante los cambios evolutivos que se producen dentro y fuera de ellas, y con ello adquirir la agilidad necesaria para ganar ventaja competitiva.

1.3.1 ¿Qué es un servicio?

Un servicio es una funcionalidad concreta que puede ser descubierta en la red y que describe tanto lo que puede hacer como el modo de interactuar con ella. Desde la perspectiva de la empresa, un servicio realiza una tarea concreta. Puede corresponder a un proceso de negocio sencillo pero también los servicios pueden acoplarse dentro de una aplicación completa que proporcione servicios de alto nivel, con un grado de complejidad muy superior en un proceso que, desde que comienza hasta que termina, puede involucrar varias aplicaciones de negocio. [8]

1.3.2 Servicios Web

La adopción de una solución de diseño basada en SOA no exige implantar servicios Web, no obstante son la forma más habitual de implementar SOA. Los servicios Web son aplicaciones que utilizan estándares para el transporte, codificación y protocolo de intercambio de información, los mismos permiten la intercomunicación entre sistemas de cualquier plataforma y se utilizan en una gran variedad de escenarios de integración, tanto dentro de las organizaciones como con socios de negocios.

Los servicios Web se basan en un conjunto de estándares de comunicación, como son XML para la representación de datos, SOAP¹ para el intercambio de datos y el lenguaje WSDL² para describir las funcionalidades de un servicio Web. [8]

1.3.3 Servicios Web REST

Los servicios web REST son una alternativa más simple a los servicios basados en SOAP y WSDL. Estos servicios cada vez están ganando más aliados y comenzarán a tomar cada vez más importancia en el escenario Java fundamentalmente.

¹ **SOAP** (del inglés *Simple Object Access Protocol*) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.

² **WSDL** (del inglés *Web Services Description Language*) describe la interfaz pública a los servicios Web.

REST define un set de principios arquitectónicos por los cuales se diseñan servicios web haciendo foco en los recursos del sistema, incluyendo cómo se accede al estado de dichos recursos y cómo se transfieren por HTTP³ hacia clientes escritos en diversos lenguajes. REST emergió en los últimos años como el modelo predominante para el diseño de servicios. De hecho, REST logró un impacto tan grande en la web que prácticamente logró desplazar a SOAP y las interfaces basadas en WSDL por tener un estilo bastante más simple de usar. [9]

No siempre REST es la mejor opción. Está surgiendo como una alternativa para diseñar servicios web con menos dependencia en middleware⁴ propietario (por ejemplo, un servidor de aplicaciones), que su contraparte SOAP y los servicios basados en WSDL. De algún modo, REST es la vuelta a la Web antes de la aparición de los grandes servidores de aplicaciones, ya que hace énfasis en los primeros estándares de Internet, URI⁵ y HTTP.

REST ayuda a cumplir con los requerimientos de integración que son críticos para construir sistemas en donde los datos tienen que poder combinarse fácilmente y extenderse. Desde este punto de vista, los servicios REST se convierten en algo de gran beneficio.

1.3.4 SOA. Retos

Las Arquitecturas Orientadas a Servicios (SOA) se imponen como la opción por defecto a la hora de abordar proyectos estratégicos.

- Actualmente las empresas dependen completamente de sus TI, de las que esperan **flexibilidad, agilidad y eficiencia** en los costes.
- Las TI de las empresas están compuestas por un gran número de **sistemas interdependientes, heterogéneos** y muchas veces **redundantes**.

³ El protocolo de transferencia de hipertexto **HTTP** (del inglés *HyperText Transfer Protocol*) es el protocolo usado en cada transacción de la Web. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.

⁴ **Middleware** es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas.

⁵ **URI** (del inglés Uniform Resource Identifier) es una cadena corta de caracteres que identifica inequívocamente un recurso. Normalmente estos recursos son accesibles en una red o sistema.

- La rapidez con que las TI pueden adaptarse a los cambios en las necesidades del negocio no siempre es suficiente (falta de **agilidad** y **flexibilidad**).
- La estructura actual de las TI puede hacer que los cambios introducidos cuesten más que los beneficios que aportan (baja **eficiencia** en los costes).

1.3.5 SOA. Soluciones

Una solución SOA no es un producto específico ni existe una SOA general. Cada organización tiene un dominio específico y una estructura de relaciones particular, y es respecto a estos que se construye una arquitectura personalizada para la organización.

- Ahorro de dinero, tiempo y esfuerzo mediante la **reutilización de componentes** y gracias a la **flexibilidad** de SOA.
- Elimina frustraciones con las TI gracias a las **soluciones flexibles y los tiempos más cortos de despliegue** de soluciones.
- Permite justificar más claramente las inversiones en TI, ya que éstas están más **alineadas con el negocio**.
- Proporciona a los ejecutivos del negocio una **visión clara de lo que hacen las TI** y su **valor** asociado.
- Permite la **creación y cambio de servicios de forma incremental**, evitando proyectos de larga duración y alto coste.

1.3.6 ¿Qué es un WSDL?

El lenguaje de descripción de servicios Web (WSDL, Web Service Description Language) es un dialecto basado en XML sobre el esquema que describe un servicio Web. Un documento WSDL proporciona la información necesaria al cliente para interactuar con el servicio Web. WSDL es extensible y se puede utilizar para describir, prácticamente, cualquier servicio de red, incluyendo SOAP sobre HTTP e incluso protocolos que no se basan en XML como DCOM sobre UDP. [10]

1.3.6.1 Elementos que conforman un WSDL

Un documento WSDL utiliza los siguientes elementos en la definición de servicios de red: [10]

- Types: contenedor de definiciones del tipo de datos que utiliza algún sistema de tipos (por ejemplo XSD).
- Message: definición abstracta y escrita de los datos que se están comunicando.
- Operation: descripción abstracta de una acción admitida por el servicio.
- Port Type: conjunto abstracto de operaciones admitidas por uno o más puntos finales.
- Binding: especificación del protocolo y del formato de datos para un tipo de puerto determinado.
- Port: punto final único que se define como la combinación de un enlace y una dirección de red.
- Service: colección de puntos finales relacionados.

1.4 Lenguajes de programación identificados para proponer el procedimiento

Un lenguaje de programación es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. La Universidad de las Ciencias Informáticas (UCI) desarrolla diversos software en distintos lenguajes, este procedimiento se enfoca en tres lenguajes específicos producto a la importancia que estos tienen para la producción de software en la UCI, los mismos se detallarán a continuación.

1.4.1 Java

Java es un lenguaje de programación desarrollado por la compañía Sun Microsystems. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito de Internet como en la informática en general.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma. Eso quiere decir que si hacemos un programa en Java podrá funcionar en cualquier ordenador del mercado. Es una ventaja significativa para los desarrolladores de software, pues antes tenían que hacer un programa para cada sistema operativo, por ejemplo Windows, Linux, Apple, etc. Esto lo consigue porque se ha creado una Máquina de Java para cada sistema que hace de puente entre el sistema operativo y el programa de Java y posibilita que este último se entienda perfectamente. [11]

1.4.1.1 La plataforma Java

Una plataforma es el ambiente de hardware o software en el cual se ejecutan los programas. En general, la mayoría de las plataformas pueden ser descritas como una combinación de hardware y sistema operativo. Algunas de las plataformas más populares son Windows, Solaris, Linux y MacOS.

La plataforma Java difiere de las anteriores en que ésta es una plataforma basada únicamente en software que corre por encima de las plataformas basadas en hardware. La plataforma Java consta de dos componentes:

- La Máquina Virtual de Java (JVM)
- La Interfaz de Programación de Aplicaciones de Java (API Java)

A continuación se muestra una representación de los elementos que forman parte de la plataforma Java. [12]

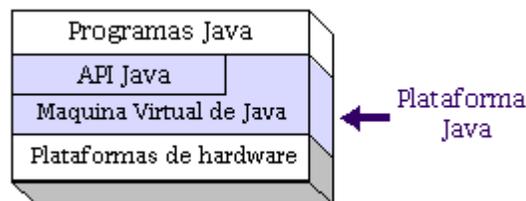


Figura 2: Elementos que forman parte de la plataforma Java

1.4.1.2 Máquina virtual de Java

En informática una máquina virtual es un software que emula a un ordenador y puede ejecutar programas como si fuese un ordenador real.

La JVM es el entorno en el que se ejecutan los programas Java, su misión principal es la de garantizar la portabilidad de las aplicaciones Java. Define esencialmente un ordenador abstracto y especifica las instrucciones (*bytecodes*⁶) que este ordenador puede ejecutar. El intérprete Java específico ejecuta las instrucciones que se guardan en los archivos cuya extensión es .class. [13]

⁶ El **bytecode** es un código intermedio más abstracto que el código máquina. Habitualmente es tratado como un fichero binario que contiene un programa ejecutable similar a un módulo objeto, que es un fichero binario producido por el compilador cuyo contenido es el código objeto o código máquina.

1.4.2 JavaServer Pages

JavaServer Pages (JSP) es una tecnología orientada a crear páginas web con programación en Java. El contenido dinámico para web se genera en forma de documentos HTML, XML o de otro tipo de lenguajes.

Las JSP permiten la utilización de código Java mediante scripts. Además, es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Bibliotecas de Etiquetas (TagLibs o Tag Libraries).

1.4.2.1 HTML

HTML, del inglés **HyperText Markup Language** (*Lenguaje de Marcas de Hipertexto*), es el lenguaje con el que se definen las páginas web. Básicamente se trata de un conjunto de etiquetas que sirven para definir el texto y otros elementos que compondrán una página web. [14]

1.4.2.2 XML

XML (del inglés eXtensible Markup Language ó Lenguaje extensible de marcas) es un conjunto de reglas que sirven para definir etiquetas semánticas para organizar un documento. Además el XML es un metalenguaje⁷ que te permite diseñar tu propio lenguaje de etiquetas. A diferencia del un lenguaje de etiquetas normal (HTML), XML te permite definir tu propio lenguaje. [15]

XML es una tecnología sencilla que tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

1.4.2.3 Script

Un script es un programa que puede acompañar a un documento HTML o que puede estar incluido en él. El programa se ejecuta en la máquina del cliente cuando se carga el documento, o en algún otro instante, como por ejemplo cuando se activa un vínculo. El soporte de scripts de HTML es independiente del lenguaje de scripts. [16]

⁷ **Metalenguaje:** es un lenguaje que se usa para hablar acerca de otro lenguaje

1.4.3 .NET

Microsoft.NET es el conjunto de nuevas tecnologías en las que Microsoft ha estado trabajando durante los últimos años con los objetivos de mejorar sus sistemas operativos, mejorar su modelo de componentes COM⁸, obtener un entorno específicamente diseñado para el desarrollo y ejecución del software en forma de servicios que puedan ser tanto publicados como accedidos a través de Internet de forma independiente del lenguaje de programación, modelo de objetos, sistema operativo y hardware utilizados tanto para desarrollarlos como para publicarlos. Éste entorno es lo que se denomina la plataforma.NET, y los servicios antes mencionados son a los que se denomina servicios web.

Para el desarrollo y ejecución de aplicaciones en este nuevo entorno tecnológico Microsoft proporciona el conjunto de herramientas conocido como .NET Framework SDK, que es posible descargarlo gratuitamente de su sitio web e incluye compiladores de lenguajes como C#, Visual Basic.NET, C++ y JScript.NET específicamente diseñados para él. [17]

La plataforma .NET – que para muchos es una respuesta de Microsoft a la plataforma Java de Sun Microsystems y al incremento de los framework de desarrollo web basados en PHP - propone desarrollar aplicaciones de una forma más rápidas y económica de modo que se pueda lograr una integración más ágil entre empresas y un acceso más sencillo a todo tipo de información desde cualquier dispositivo.

1.4.3.1 Lenguajes de programación desarrollados para el marco de trabajo .NET. C Sharp (C#)

Algunos de los lenguajes de programación desarrollados para el marco de trabajo .NET son: C#, Visual Basic, Delphi (Object Pascal), C++, J#, Perl, Python, Fortran, Prolog, Cobol y PowerBuilder.

En este trabajo haremos énfasis en el lenguaje de programación C# el cual es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft, que después fue aprobado como un estándar por la ISO.

⁸ **COM** (del inglés **Component Object Model**) es una plataforma de Microsoft para componentes de software introducida por dicha empresa en 1993. Esta plataforma es utilizada para permitir la comunicación entre procesos y la creación dinámica de objetos, en cualquier lenguaje de programación que soporte dicha tecnología.

C# (leído en inglés “C Sharp” y en español “C Almohadilla”) es el nuevo lenguaje de propósito general diseñado por Microsoft para su plataforma .NET. Sus principales creadores son Scott Wiltamuth y Anders Hejlsberg, éste último también conocido por haber sido el diseñador del lenguaje Turbo Pascal y la herramienta RAD Delphi.

Aunque es posible escribir código para la plataforma .NET en muchos otros lenguajes, C# es el único que ha sido diseñado específicamente para ser utilizado en ella, por lo que programarla usando C# es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes ya que C# carece de elementos heredados innecesarios en .NET. Por esta razón, se suele decir que C# es el lenguaje nativo de .NET. [18]

1.5.4 PHP

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Fue creado originalmente en 1994 por Rasmus Lerdorf, pero como **PHP está desarrollado en política de código abierto**, a lo largo de su historia ha tenido muchas contribuciones de otros desarrolladores.

Este lenguaje de programación está preparado para realizar muchos tipos de aplicaciones web gracias a la extensa librería de funciones con la que está dotado. La librería de funciones cubre desde cálculos matemáticos complejos hasta tratamiento de conexiones de red, por poner dos ejemplos. [19]

Conclusiones parciales

En este capítulo se realizó un estudio de las principales compañías que están desarrollando soluciones con arquitectura orientada a servicio a nivel mundial, a partir del estudio realizado se arribó a la conclusión de que el mercado SOA está ampliamente liderado por empresas desarrolladoras de software propietario las cuales poseen la mayor cantidad de información, experiencias, metodologías y las técnicas de soporte de las arquitecturas orientadas a servicios. Se abordaron aspectos importantes relacionados con las arquitecturas orientadas a servicios haciendo énfasis en los beneficios que se pueden obtener al implantar este tipo de arquitectura de software. Se recopiló información necesaria relacionada con temas

importantes de la investigación como son: Servicio Web, Servicio, WSDL y además se realizó un estudio de los lenguajes de programación sobre los cuales se va a proponer el procedimiento.

CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Introducción

El presente capítulo está dedicado a describir y explicar el procedimiento para la integración de sistemas legados en organizaciones empresariales con una arquitectura orientada a servicios, durante la fase de implementación en un proyecto SOA. Lo cual es la solución resultante de la investigación; se definen los artefactos involucrados, los roles que intervienen y las responsabilidades asociadas a los mismos. La solución se basa en un procedimiento que puede ser usado en empresas que vayan a adoptar SOA.

2.1 ESTRUCTURA DEL PROCEDIMIENTO

El procedimiento provee los subprocesos, actividades, roles involucrados y responsabilidades asociadas a ellos, artefactos generados y consultados, técnicas y herramientas a utilizar. Durante la especificación de los subprocesos se mantiene una estructura análoga que incluye una breve descripción del mismo, una representación de los roles, herramientas y técnicas que intervienen en el subproceso y las entradas y salidas del mismo. Posteriormente se explican los flujos de actividades que componen los subprocesos y cada uno de sus pasos.

Para cada actividad se especifican los roles que intervienen en la misma, entradas, salidas y las herramientas y técnicas que se utilizarán en su ejecución, para que sirva de guía y orientación a las personas que decidan aplicar el procedimiento.

2.2 ALCANCE DEL PROCEDIMIENTO

El procedimiento es aplicable a cualquier iniciativa corporativa de adoptar una SOA. El mismo ofrece la descripción y guía objetiva para la realización de cada uno de los subprocesos y actividades involucrados en la integración de sistemas legados dentro de cualquier empresa que se aliste a implantar una SOA. De cada uno de los subprocesos y actividades propuestos en el procedimiento se especifican las entradas,

salidas, participantes y la descripción de los elementos que conforman las actividades. Además, se definen técnicas y herramientas a utilizar durante la realización de estas actividades.

2.3 PRINCIPIOS DEL PROCEDIMIENTO

El procedimiento se basa en tres principios fundamentales que se encargan de garantizar su adecuado funcionamiento y a los que tributan todos los subprocesos y actividades de una forma u otra.

- *Todo el personal de la disciplina implementación de servicios está involucrado en este procedimiento:* el procedimiento concibe a todo el personal involucrado en la disciplina implementación de servicios; siendo estos los responsables de la correcta aplicación del mismo. Los roles específicos que se encargan de ejecutar las acciones de implementación de servicios en el proyecto son responsables de cumplir con las normas definidas, haciéndolo de manera consciente, efectiva y en tiempo. Para que el cumplimiento de las normas sea efectivo no puede ser tarea de unos pocos dentro del equipo, sino que debe ser un proceso cooperativo de todos los involucrados en el desarrollo del producto.
- *Carácter proactivo orientado a los resultados:* el procedimiento orienta sus actividades con una perspectiva de mejorar continuamente su aplicación y así aumentar los resultados. El principal elemento que contribuye a la mejora es el aprendizaje. El procedimiento asume que un aprendizaje continuo que trascienda de proyecto a proyecto, es la clave para aumentar los resultados. Este es un factor importante para la mejora continua de la implementación de servicios, su aplicación efectiva y el éxito en general.
- *Enfocado a los clientes:* con su aplicación, el procedimiento se centra en obtener beneficios visibles para los clientes. El procedimiento mantiene un equilibrio con la calidad, tiempo y costo para lograr un efecto similar en todas las partes involucradas en el proyecto. El procedimiento vela por el ajuste y cumplimiento de estos principios y las normas asociadas a ellos.

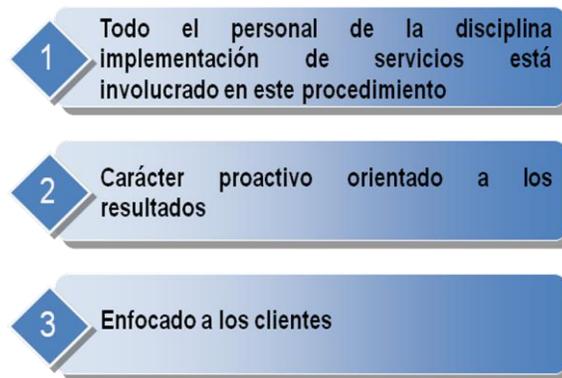


Figura 3: Principios del procedimiento

2.4 PREMISAS PARA LA APLICACIÓN DEL PROCEDIMIENTO

Para la correcta aplicación del procedimiento, son necesarias las siguientes premisas:

- *Personal capacitado en el tema:* es necesario que todo el personal del área de implementación de servicios entienda los principios del procedimiento y conozca el papel que desempeñará cada cual en la aplicación del mismo. Cada miembro de la disciplina implementación de servicios debe estar capacitado para impartir cursos de capacitación a las personas que formen parte del proyecto. Es por ello que la preparación del personal ayudará significativamente a la correcta y rápida aplicación del procedimiento.
- *Cultura organizacional:* para lograr el éxito del procedimiento es necesario que exista un buen trabajo en equipo, donde se facilite el acceso a la información y todos puedan recibir ayuda de las personas que sean capaces de brindarla en el menor tiempo posible.
- *Recursos humanos con la responsabilidad de implementación de servicios asignada:* para la correcta aplicación de este procedimiento, la dirección del proyecto debe designar personas que tengan responsabilidades específicas referentes a la disciplina y respondan por su cumplimiento.
- *Concordancia entre la dirección del proyecto y la de la empresa:* para lograr que el procedimiento se aplique con efectividad, es necesario que la dirección del proyecto y la dirección de la empresa en la que se esté laborando, estén completamente de acuerdo en todas las definiciones y

decisiones que deben tomarse, para que las actividades y las normas que se definan estén enfocadas a cumplir con las necesidades de la entidad y satisfagan sus intereses.



Figura 4: Premisas para la aplicación del procedimiento

2.5 ROLES QUE INTERACTÚAN CON EL PROCEDIMIENTO

En esta sección se resumen los roles y responsabilidades que han sido identificados para interactuar con el procedimiento. Los mismos son los encargados de realizar cada una de las actividades propuestas en el mismo.

2.5.1 Arquitecto SOA

Tiene la responsabilidad global de dirigir las principales decisiones técnicas, especialmente vinculadas con temas orientados a servicios. Esto habitualmente incluye la identificación y la documentación de los aspectos arquitectónicamente significativos de la arquitectura, que incluye las vistas de especificación, implementación y despliegue. Es una clase de arquitecto TI. Define tanto la infraestructura tecnológica como los modelos técnicos con impacto en toda la compañía, comunes a varias aplicaciones. Depende del director de sistemas de información, o el responsable de SOA dentro de la dirección de sistemas de información. Estos dos aspectos (infraestructura tecnológica y modelos) pueden también realizarse de forma especializada.

2.5.1.1 Artefactos que realiza el rol Arquitecto SOA

El arquitecto SOA es el responsable de elaborar los siguientes documentos: Reglas de Confidencialidad del Código Fuente, Plan de Transición y Especificación Detallada de Servicio.

2.5.2 Desarrollador

Este rol es el encargado de Analizar las descripciones de las Unidades de Automatización (AU), analizar las especificaciones de los servicios, analizar el Plan de Transición de los Sistemas Legados y a partir de todas estas tareas generar la especificación de las AU. Además es el encargado del diseño de las AU.

2.5.2.1 Artefactos que realiza el rol Desarrollador

El desarrollador es el responsable de elaborar el los siguientes artefactos: Clase integradora y Servicios exportados.

2.5.3 Implementador

Este rol es el encargado de codificar y probar las Unidades de Automatización (AU) a partir de la especificación de los servicios y el diseño de las AU. Además es el encargado de generar las instrucciones de despliegue de los servicios.

2.5.3.1 Artefactos que realiza el rol Implementador

El desarrollador es el responsable de elaborar el los siguientes artefactos: Clase integradora y Servicios exportados.

2.6 REPRESENTACIÓN DEL PROCEDIMIENTO

Para la aplicación del procedimiento se proponen los subprocesos: Exportación de Servicios en Java, Exportación de Servicios en C# y Exportación de Servicios en PHP.

Las actividades propuestas por el procedimiento son:

Capítulo 2. Descripción de la solución propuesta

- *Obtener el código fuente:* en esta actividad se realizarán un conjunto de negociaciones entre la dirección del proyecto y la dirección de la empresa en la que se esté laborando con el fin de lograr dicho objetivo.
- *Elaborar el Plan de Transición:* se confecciona el Plan de Transición el cual especifica como el software actual será migrado a una Arquitectura Orientada a Servicios.
- *Definir funcionalidades a exportar como servicio:* define las funcionalidades de interés para la empresa que serán exportadas como servicios.
- *Probar los servicios:* se realizarán un conjunto de pruebas con el fin de garantizar que los servicios exportados posean la calidad óptima.

La Figura 5 muestra la representación general del procedimiento de exportación de servicios que se propone, con los subprocesos y actividades, además del conjunto de los artefactos fundamentales que intervienen en el flujo.

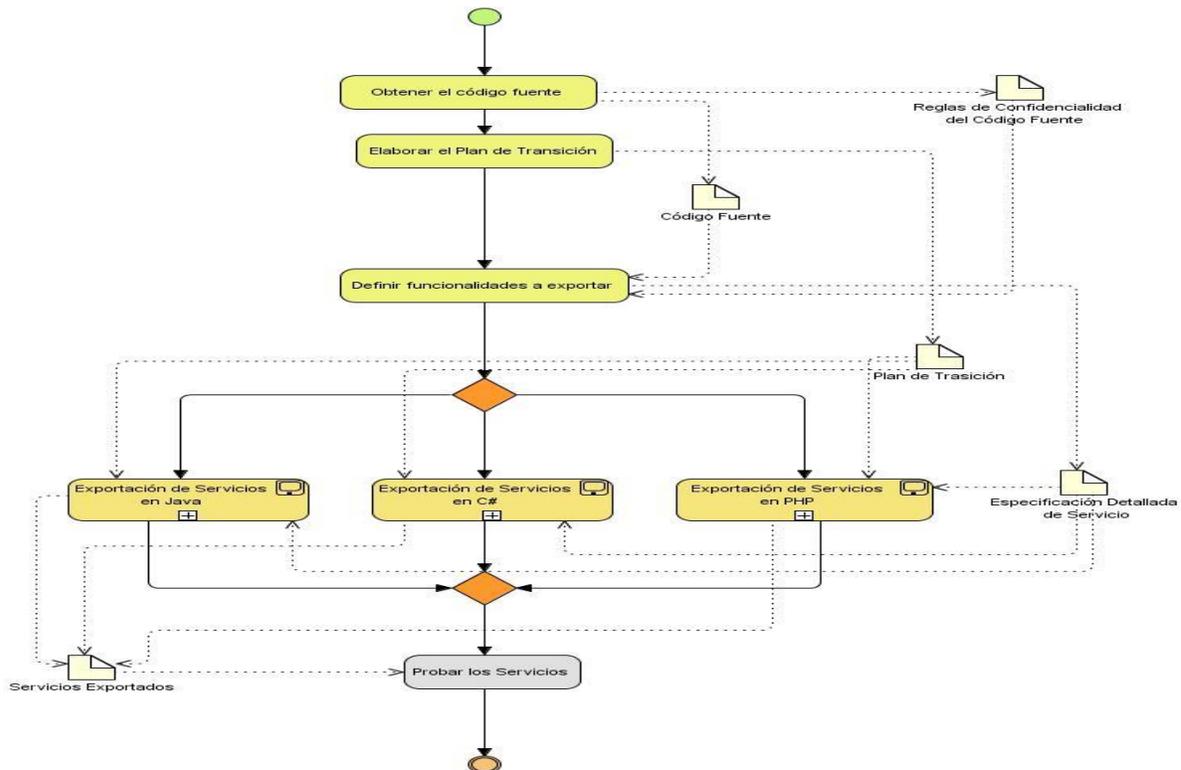


Figura 5: Representación general del procedimiento de exportación de servicios

2.7 DESCRIPCIÓN DEL PROCEDIMIENTO

A lo largo de esta sección se describen los subprocesos y actividades que conforman el procedimiento propuesto.

En cada una de las actividades se especifican los roles, herramientas, técnicas, artefactos de entrada y artefactos de salida que estén involucrados en su realización. Si es un subproceso se incluirá la descripción del flujo de actividades asociadas al mismo.

A continuación se explican en qué consisten cada uno de los elementos antes mencionados:



Roles

Aquí se muestran los roles involucrados en la realización de la actividad o subproceso que se describe, los cuales son los responsables de su cumplimiento y de la realización de los artefactos que se generen.



Técnicas y herramientas

Se especifican las técnicas y herramientas que se utilizan para la realización de la actividad o subproceso que se describe.



Artefactos de entrada

En esta sección se enuncian los artefactos que son necesarios utilizar durante la actividad o subproceso.



Artefactos de salida

Son aquellos que se generan durante la realización de la actividad que se describe. Los artefactos de salida se consideran entregables de la actividad en que fueron generados.



Flujo de actividades

Son las actividades que se desarrollan dentro de un subproceso.

2.7.1 Obtener el código fuente



Roles

- Arquitecto SOA



Técnicas y Herramientas

- Entrevista
- Procesadores de textos para la elaboración de los documentos



Artefactos de entrada

- No posee artefacto de entrada



Artefactos de salida

- Código fuente
- Reglas de Confidencialidad del Código Fuente

Para aplicar correctamente el procedimiento que se propone es necesario obtener el código fuente del sistema legado del cual se desean exportar sus funcionalidades como servicios. En esta actividad se realizarán un conjunto de negociaciones entre la dirección del proyecto y la dirección de la empresa en la que se esté laborando con el fin de lograr dicho objetivo. Para darle cumplimiento a esta actividad se utilizó la técnica de la entrevista.

Entrevista: La entrevista es una técnica que se realiza a manera de conversación. La misma se realiza de forma individual o a grupos de personas. Las personas que serán entrevistadas, deben ser seleccionadas cuidadosamente. Dichas personas deben tener un vasto conocimiento del tema que se abordará a lo largo de la entrevista para asegurar que brinden la información confiable y necesaria; con esta información, el entrevistador puede conocer y obtener datos que no estén disponibles en otras fuentes y que son de suma importancia para futuras actividades.

El encargado de la realización de esta actividad es el arquitecto SOA (Ver descripción).

Como artefactos de salida se obtienen el código fuente del sistema legado pues una vez concluida esta actividad el proyecto cuenta con lo imprescindible para iniciar la implementación de los servicios. El otro artefacto de salida que se obtiene es la planilla Reglas de Confidencialidad del Código Fuente (Ver Complementario 1) la cual constituye el documento legal entre las empresas y la dirección del proyecto, el cual establece un mayor compromiso y seriedad con el código fuente obtenido.

2.7.2 Elaborar el Plan de Transición



Roles

- Arquitecto SOA



Técnicas y Herramientas

- Procesadores de textos para la elaboración de los documentos



Artefactos de entrada

- No posee artefacto de entrada



Artefactos de salida

- Plan de Transición

En esta actividad el arquitecto SOA confecciona el Plan de Transición, este explica las principales acciones a seguir para migrar el sistema legado existente a una arquitectura orientada a servicios. Este plan también contiene un cronograma por segmento SOA, donde se especifiquen las tareas, insumos, entregables, y recursos utilizados para la ejecución; para esto preferiblemente se debe usar diagramas de Gantt. Además de esto se especifican las condiciones necesarias existentes para comenzar a ejecutar la transición y las condiciones existentes para comenzar a consumir los servicios desplegados por la transición efectuada.

El rol encargado de darle cumplimiento a esta actividad es el arquitecto SOA (Ver descripción) y como artefacto de salida se obtiene el Plan de Transición (Ver Complementario 2).

2.7.3 Definir funcionalidades a exportar



Roles

- Arquitecto SOA



Técnicas y Herramientas

- Entrevista
- Procesadores de textos para la elaboración de los documentos



Artefactos de entrada

- Código fuente
- Reglas de Confidencialidad del Código Fuente



Artefactos de salida

- Especificación Detallada de Servicio

Durante el desarrollo de esta actividad se deben definir las funcionalidades del sistema legado que serán expuestas como servicios. Con el resultado obtenido se puede realizar con más precisión las próximas actividades. El rol encargado de darle cumplimiento a esta actividad es el arquitecto SOA (Ver descripción).

La información necesaria se obtiene a través de la utilización de la técnica entrevista, la cual se realiza inicialmente a los directivos y especialistas de la empresa en la que se esté laborando, para poder obtener la información necesaria que servirá de base para actividades posteriores. Estas entrevistas también se les deben realizar a aquellas personas que interactúen directamente con el sistema legado a modificar para que estas brinden información que podría ser valiosa a la hora de exportar los servicios.

Como artefacto de entrada se utiliza el código fuente del sistema legado y la planilla Reglas de Confidencialidad del Código Fuente la cual establece la conducta a seguir sobre el código fuente de la aplicación.

Como salida se obtiene el artefacto Especificación Detallada de Servicio (Ver Complementario 3), este documento recoge las propiedades del servicio, la calidad del servicio, el cumplimiento de las normas,

entre otros aspectos que recogen de forma general las características que poseerán los servicios exportados.

2.7.4 Exportación de servicios en C#



Roles

- Desarrollador
- Implementador



Técnicas y Herramientas

- Visual Studio 2005
- Tormenta de Ideas
- Taller



Artefactos de entrada

- Código fuente
- Especificación Detallada de Servicio
- Plan de Transición



Artefactos de salida

- Servicios exportados
- Clase integradora

En este subproceso se asignan las responsabilidades a los distintos miembros del proyecto que intervendrán directamente en la exportación de los servicios en el lenguaje de programación C#, para garantizar una mayor eficiencia y calidad del resultado final. Los responsables de darle cumplimiento a las actividades de este subproceso son el desarrollador (Ver descripción) y el implementador (Ver descripción). Para que los miembros del proyecto respondan a un mismo fin se realizarán talleres y tormentas de ideas.

Taller: La técnica de taller propone un espacio y un tiempo de comunicación, reflexión y creatividad participativa. El taller debe ser protagónico y activo, movilizandando a los integrantes de un grupo determinado para compartir y colaborar. Aspira a desarrollar una comunicación espontánea entre los participantes. El taller es una técnica que proporciona la posibilidad de lograr determinados objetivos, por lo cual esta modalidad es aplicable a muy diversas temáticas. En los talleres que se realicen en este

Capítulo 2. Descripción de la solución propuesta

subproceso se deben realizar actividades tales como trabajos en grupos y secciones generales para lograr que todo el personal del proyecto trabaje con mayor facilidad y de forma coordinada.

Tormenta de ideas: La tormenta de ideas es una técnica en la que un grupo de personas, en conjunto, crean ideas. Esto es casi siempre más productivo que cada persona pensando por sí sola. Esta técnica se puede utilizar cuando se desea o se necesita obtener una conclusión grupal en relación a un problema que involucra a todo un grupo y cuando es importante motivar al grupo, tomando en cuenta las participaciones de todos, bajo reglas determinadas. El clima de participación y motivación en el que se desarrollen las reuniones asegurará mayor calidad en las decisiones tomadas por el grupo, más compromiso con la actividad y un sentimiento de responsabilidad compartido por todos es por ellos que dichas reuniones deben ser relajadas, amenas e incluso divertidas.

Durante la realización de este subproceso se utilizan los siguientes artefactos de entrada:

Código fuente: en este artefacto se encuentra descrito por completo el funcionamiento del sistema legado.

Especificación Detallada de Servicio: este documento recoge las propiedades del servicio, la calidad del servicio, el cumplimiento de las normas, entre otros aspectos.

Plan de Transición: explica las principales acciones a seguir para migrar el sistema legado existente a una arquitectura orientada a servicios.

Durante este subproceso, se genera como artefacto de salida:

Servicios Exportados: este artefacto es un WSDL el cual contiene todas las funcionalidades que fueron exportadas como servicios.

Clase Integradora: contiene todas las funcionalidades que se desean exportar como servicios que estaban dispersas en varias clases de la aplicación.



Flujo de actividades

El flujo de actividades asociado a este subproceso comienza cuando están definidas las funcionalidades que serán exportadas como servicios. Estas actividades son iniciadas por el desarrollador (Ver descripción) y el implementador (Ver descripción) y al culminar las mismas quedarán implementadas

como servicios las funcionalidades que fueron definidas en la planilla de Especificación Detallada de Servicio.

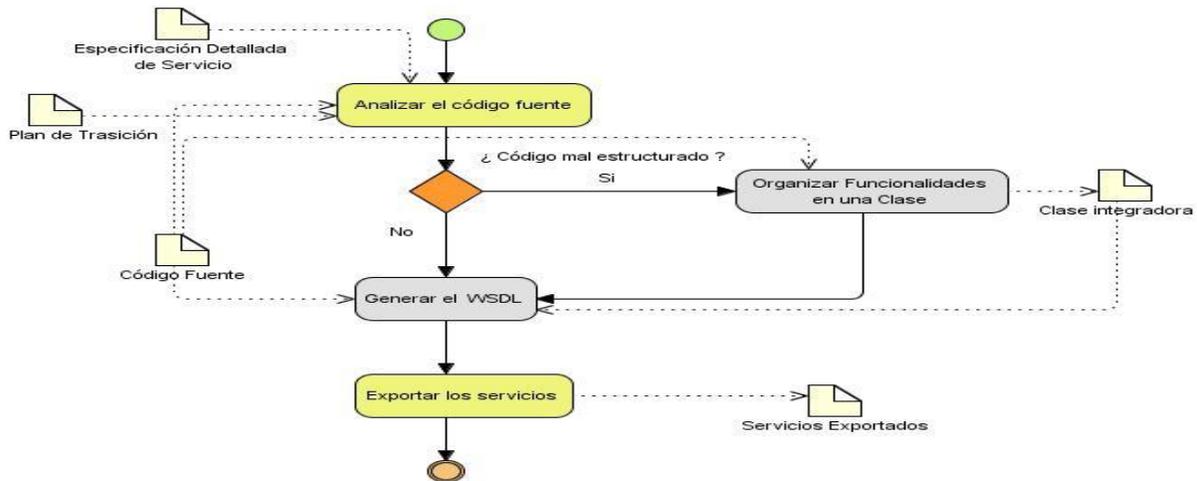


Figura 6: Flujo de actividades del subproceso Exportación de Servicios en C#

La Figura 6 muestra la representación más general del flujo de actividades del subproceso, con el conjunto de los artefactos que intervienen en el flujo.

2.7.4.1 Analizar el código fuente



Roles

- Desarrollador



Técnicas y Herramientas

- Tormenta de Ideas
- Taller



Artefactos de entrada

- Código fuente
- Especificación Detallada de Servicio
- Plan de Transición



Artefactos de salida

- No posee artefacto de salida

El primer paso que se debe dar para exportar servicios en C# es analizar detenidamente el código fuente en que está desarrollado el sistema legado, que ya fue obtenido como premisa para la aplicación de procedimiento.

Para la realización de esta actividad el grupo se debe reunir para obtener una visión del problema que se tratará. El objetivo fundamental es definir si las funcionalidades que serán exportadas como servicios están contenidas en una misma clase o dispersas en varias clases de la aplicación que se está analizando. Es por ello que se emplean las técnicas de tormenta de ideas y talleres para garantizar que todo el grupo trabaje bajo la misma línea.

El rol encargado de realizar esta actividad es el desarrollador (Ver descripción).

Como artefactos de entrada se tienen el código fuente del sistema legado, la planilla de Especificación Detallada de Servicio y el Plan de Transición.

2.7.4.2 Organizar funcionalidades en una clase



Roles

- Desarrollador
- Implementador



Técnicas y Herramientas

- Visual Studio 2005



Artefactos de entrada

- Código fuente



Artefactos de salida

- Clase integradora

Esta actividad se realiza en caso de que las funcionalidades que serán exportadas como servicios no se encuentren contenidas en una misma clase. Es por ello que surge la necesidad de crear una nueva clase que encierre dichas funcionalidades.

Los roles encargados de realizar esta actividad son el desarrollador (Ver descripción) y el implementador (Ver descripción).

Como artefacto de entrada se utiliza el código fuente del sistema legado y como artefacto de salida se obtiene la clase integradora la cual contiene todas las funcionalidades que se desean exportar como servicios que estaban dispersas en varias clases de la aplicación.

2.7.4.3 Generar el WSDL



Roles

- Desarrollador
- Implementador



Técnicas y Herramientas

- Visual Studio 2005



Artefactos de entrada

- Código fuente
- Clase integradora



Artefactos de salida

- No posee artefactos de entrada

Los roles encargados de realizar esta actividad son el desarrollador (Ver descripción) y el implementador (Ver descripción). Como artefactos de entrada se utiliza el código fuente del sistema legado en caso que las funcionalidades que se desean exportar como servicios se encuentran bien definidas en una misma clase, de no ser así el artefacto que sirve de entrada a esta actividad es la clase integradora que se fue creada en la actividad anterior.

Lo primero que se debe hacer una vez que se tengan definidas las funcionalidades en una misma clase ya sea en el código fuente original o en la clase integradora que se creó en la actividad anterior es crear un Servicio Web ASP.NET en el Visual Studio 2005, para ello se accede al menú Archivo\Nuevo\Sitio Web, en la ventana Nuevo sitio Web se selecciona la plantilla "Servicio Web ASP.NET", se especifica la dirección y el nombre del servicio web creado. (Ver anexo 1)

Una vez creado el servicio web en la ruta especificada se procede a copiar todo el código fuente del sistema legado dentro de la carpeta App_Code de ese mismo directorio. Después de haber copiado el sistema legado en la carpeta especificada anteriormente se debe actualizar el contenido del servicio web

creado, para ello se accede al Explorador de Soluciones del Visual Studio 2005 y se pulsa clic derecho encima del servicio web y se selecciona la opción Actualizar carpeta. (Ver anexo 2)

Seguidamente se accede a la clase donde se encuentran las funcionalidades que se desean exportar como servicios y en la sección “using” - en la que se especifican qué espacios de nombres se van a usar en esta clase - se incluyen los espacios de nombre: System.Web, System.Web.Services y System.Web.Services.Protocols. En caso de que la clase posea un “namespace” - el cual se utiliza para declarar un ámbito que permite organizar el código y proporciona una forma de crear tipos globalmente únicos - se elimina. Luego se asocia el atributo “WebMethod” a las funcionalidades que se desean exportar como servicios para indicar que se desea exponer dicho método como parte del servicio Web XML. Es importante señalar que el atributo “WebMethod” se escribe entre corchetes.

Además de configurar la clase donde se encuentran las funcionalidades que serán exportadas como servicios se debe configurar el archivo “Service.asmx” que no es más que un archivo de texto que sirve de punto de entrada direccionable del servicio Web XML. Este archivo cuenta con la directiva de procesamiento “WebService” situada en la parte superior del mismo que determina dónde se buscará la implementación del servicio Web XML. La directiva de procesamiento “WebService” cuenta con varios atributos. “Class” es el atributo más importante porque es el que indica qué clase del archivo de código implementará la funcionalidad del servicio Web XML, “Language” indica el lenguaje de programación utilizado para desarrollar el servicio Web XML y “Codebehind” ayuda a Visual Studio a administrar el proyecto de servicio Web XML; el archivo “Service.asmx” puede prescindir de estos dos últimos atributos porque no son necesarios en tiempo de ejecución.

2.7.4.3 Exportar servicios



Roles

- Desarrollador
- Implementador



Técnicas y Herramientas

- Visual Studio 2005



Artefactos de entrada

- No posee artefactos de entrada



Artefactos de salida

- Servicios exportados

Esta actividad comienza cuando se tengan configuradas las funcionalidades que se van a exportar como servicios. Cuando se proceda a probar el Servicio Web que se acaba de implementar el Servidor de Desarrollo de ASP.NET se ejecuta automáticamente. Al ejecutarse este servidor se mostrará en un navegador la dirección de donde está publicado el WSDL, al cual se accede mediante el vínculo “descripción de servicios”.

Los roles encargados de realizar esta actividad son el desarrollador (Ver descripción) y el implementador (Ver descripción). Como artefacto de salida de esta actividad se obtiene los servicios exportados.

2.7.5 Exportación de servicios en Java



Roles

- Desarrollador
- Implementador



Técnicas y Herramientas

- Eclipse Galileo
- Tormenta de Ideas
- Taller



Artefactos de entrada

- Código fuente
- Especificación Detallada de Servicio
- Plan de Transición



Artefactos de salida

- Servicios exportados
- Clase integradora

En este subproceso se asignan las responsabilidades a los distintos miembros del proyecto que intervendrán directamente en la exportación de los servicios en el lenguaje de programación Java, para garantizar una mayor eficiencia y calidad del resultado final. Los responsables de darle cumplimiento a las

Capítulo 2. Descripción de la solución propuesta

actividades de este subproceso son el desarrollador (Ver descripción) y el implementador (Ver descripción). Para que los miembros del proyecto respondan a un mismo fin se realizarán talleres y tormenta de ideas.

Durante la realización de este subproceso se utilizan los siguientes artefactos de entrada:

Código fuente: en este artefacto se encuentra descrito por completo el funcionamiento del sistema legado.

Especificación Detallada de Servicio: este documento recoge las propiedades del servicio, la calidad del servicio, el cumplimiento de las normas, entre otros aspectos.

Plan de Transición: explica las principales acciones a seguir para migrar el sistema legado existente a una arquitectura orientada a servicios.

Durante este subproceso, se genera como artefacto de salida:

Servicios Exportados: este artefacto es un WSDL el cual contiene todas las funcionalidades que fueron exportadas como servicios.

Clase Integradora: contiene todas las funcionalidades que se desean exportar como servicios que estaban dispersas en varias clases de la aplicación.



Flujo de actividades

El flujo de actividades asociado a este subproceso comienza cuando están definidas las funcionalidades que serán exportadas como servicios. Estas actividades son iniciadas por el desarrollador (Ver descripción) y el implementador (Ver descripción) y al culminar las mismas quedarán implementadas como servicios las funcionalidades que fueron definidas en la planilla de Especificación Detallada de Servicio.

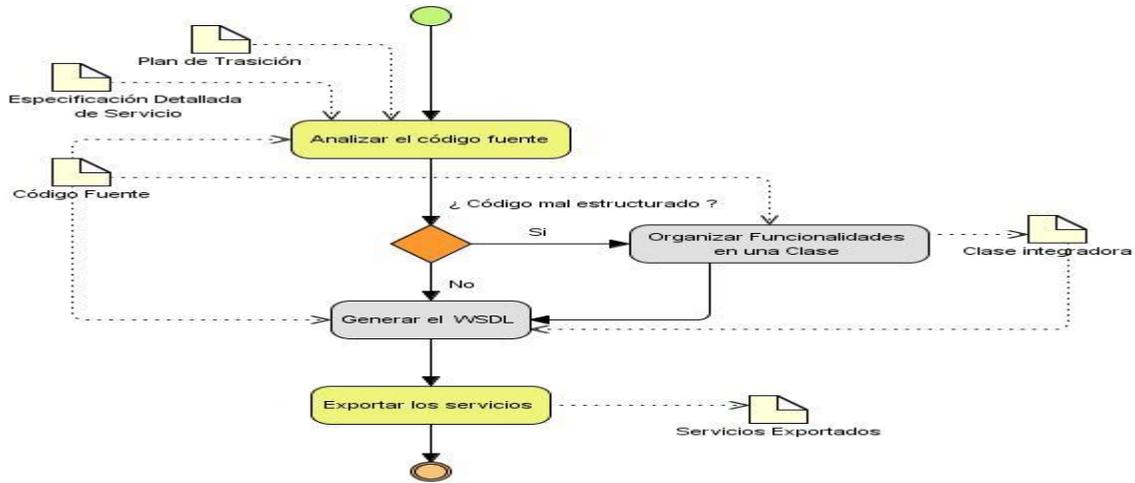


Figura 7: Flujo de actividades del subproceso Exportación de Servicios en Java

La Figura 7 muestra la representación más general del flujo de actividades del subproceso, con el conjunto de los artefactos que intervienen en el flujo.

2.7.5.1 Analizar el código fuente



Roles

- Desarrollador



Técnicas y Herramientas

- Tormenta de Ideas
- Taller



Artefactos de entrada

- Código fuente
- Especificación Detallada de Servicio
- Plan de Transición



Artefactos de salida

- No posee artefacto de salida

El paso inicial que se debe dar para exportar servicios en Java es analizar detenidamente el código fuente en que está desarrollado el sistema legado, que ya fue obtenido como premisa para la aplicación de procedimiento.

Para la realización de esta actividad el grupo se debe reunir para obtener una visión de general de cómo está implementado el sistema legado. El objetivo fundamental es definir si las funcionalidades que serán exportadas como servicios están contenidas en una misma clase o dispersas en varias clases de la aplicación que se está analizando. Es por ello que se emplean las técnicas de tormenta de ideas y talleres para garantizar que todo el grupo trabaje bajo la misma línea.

El rol encargado de realizar esta actividad es el desarrollador (Ver descripción).

Como artefactos de entrada se tienen el código fuente del sistema legado, la planilla de Especificación Detallada de Servicio y el Plan de Transición.

2.7.5.2 Organizar funcionalidades en una clase



Roles

- Desarrollador
- Implementador



Técnicas y Herramientas

- Eclipse Galileo



Artefactos de entrada

- Código fuente



Artefactos de salida

- Clase integradora

Esta actividad se realiza en caso de que las funcionalidades que serán exportadas como servicios no se encuentren contenidas en una misma clase. Es por ello que surge la necesidad de crear una nueva clase que encierre dichas funcionalidades.

Los roles encargados de realizar esta actividad son el desarrollador (Ver descripción) y el implementador (Ver descripción).

Como artefacto de entrada se utiliza el código fuente del sistema legado y como artefacto de salida se obtiene la clase integradora la cual contiene todas las funcionalidades que se desean exportar como servicios que estaban dispersas en varias clases de la aplicación.

2.7.5.3 Generar el WSDL



Roles

- Desarrollador
- Implementador



Técnicas y Herramientas

- Eclipse Galileo
- WSAS de WSO2



Artefactos de entrada

- Código fuente
- Clase integradora



Artefactos de salida

- No posee artefacto de salida

Los roles encargados de realizar esta actividad son el desarrollador (Ver descripción) y el implementador (Ver descripción). Como artefactos de entrada se utiliza el código fuente del sistema legado en caso que las funcionalidades que se desean exportar como servicios se encuentran bien definidas en una misma clase, de no ser así el artefacto que sirve de entrada a esta actividad es la clase integradora que se fue creada en la actividad anterior.

Lo primero que se debe hacer es seleccionar la clase donde están definidas las funcionalidades que serán exportadas como servicios ya sea en el código fuente original o en la clase integradora que se creó en la actividad anterior.

Luego se procede con la creación del XML del servicio web. Para ello se accede al menú Archivo/Nuevo/Otro y se selecciona Web Service en la ventana que se muestra y se pulsa el botón Siguiente. (Ver anexo 3)

Capítulo 2. Descripción de la solución propuesta

Seguidamente se configura la aplicación para que los servicios se publiquen en el servidor de aplicaciones para servicios web de WSO2⁹. Esto se hace accediendo al vínculo "Web service runtime" donde se selecciona "WSO2 Web Services Application Server" en la sección Web service runtime y "WSO2 WSAS 3.1" en la sección Server. (Ver anexo 4)

Si se hizo lo correcto ya se tiene publicado el servicio web en el servidor aplicaciones de WSO2.

WSO2 Web Services Application Server

El Servidor de Aplicaciones para Servicios Web de WSO2 es una herramienta de software libre basado en Apache Axis2. Al englobar cierto número de proyectos de servicios Web Apache, WSO2 WSAS proporciona un medio seguro, transaccional y confiable para crear, consumir, desplegar y administrar servicios Web para una SOA.

Es una plataforma simple e interoperable para recibir servicios web basados en J2EE y simplifica el desarrollo, el despliegue, y la administración de estos servicios web. Con sus características soporta el clustering y la alta disponibilidad, la integración del IDE de desarrollo Eclipse y los estándares REST, WSAS se está convirtiendo en la plataforma preferida para los desarrolladores y los arquitectos SOA. [20]

2.7.5.3 Exportar servicios



Roles

- Desarrollador
- Implementador



Técnicas y Herramientas

- Eclipse Galileo
- WSAS de WSO2

⁹ **WSO2** es una empresa de desarrollo de aplicaciones informáticas de software libre la cual se centra fundamentalmente en el abastecimiento de soluciones para desarrolladores de Arquitecturas Orientadas a Servicios.



Artefactos de entrada

- No posee artefactos de entrada



Artefactos de salida

- Servicios exportados

Cuando se ejecuta el WSAS de WSO2 se abrirá la página web de inicio del WSAS. Para ver los servicios que están disponibles en ese servidor se accede al vínculo “List” de dicha página que se encuentra en el menú izquierdo de la misma. Una vez que se muestren todos los servicios disponibles se accede al vínculo “Try this service” del servicio que se desea obtener el WSDL, inicialmente en la URL del navegador aparecerá `http://IP:Puerto/services/NombreServicio?tryit`. Para obtener el WSDL se debe sustituir la palabra “tryit” - que se encuentra detrás del signo de interrogación en la URL del servicio – por la palabra WSDL, quedando esta URL de la siguiente forma `http://IP:Puerto/services/NombreServicio?wsdl`.

Los roles encargados de realizar esta actividad son el desarrollador (Ver descripción) y el implementador (Ver descripción). Como artefacto de salida de esta actividad se obtiene los servicios exportados.

2.7.5 Exportación de servicios en PHP



Roles

- Desarrollador
- Implementador



Técnicas y Herramientas

- ZendStudio-5.5.0
- Servidor Web que soporte PHP (Server2Go)
- Tormenta de Ideas
- Taller



Artefactos de entrada

- Código fuente
- Especificación Detallada de Servicio
- Plan de Transición



Artefactos de salida

- Servicios exportados
- Clase integradora

En este subproceso se asignan las responsabilidades a los distintos miembros del proyecto que intervendrán directamente en la exportación de los servicios en el lenguaje de programación PHP, para garantizar una mayor eficiencia y calidad del resultado final. Los responsables de darle cumplimiento a las actividades de este subproceso son el desarrollador (Ver descripción) y el implementador (Ver descripción). Para que los miembros del proyecto respondan a un mismo fin se realizarán talleres y tormenta de ideas.

Durante la realización de este subproceso se utilizan los siguientes artefactos de entrada:

Código fuente: en este artefacto se encuentra descrito por completo el funcionamiento del sistema legado.

Especificación Detallada de Servicio: este documento recoge las propiedades del servicio, la calidad del servicio, el cumplimiento de las normas, entre otros aspectos.

Plan de Transición: explica las principales acciones a seguir para migrar el sistema legado existente a una arquitectura orientada a servicios.

Durante este subproceso, se genera como artefacto de salida:

Servicios Exportados: este artefacto es un WSDL el cual contiene todas las funcionalidades que fueron exportadas como servicios.

Clase Integradora: contiene todas las funcionalidades que se desean exportar como servicios que estaban dispersas en varias clases de la aplicación.



Flujo de actividades

El flujo de actividades asociado a este subproceso comienza cuando están definidas las funcionalidades que serán exportadas como servicios. Estas actividades son iniciadas por el desarrollador (Ver descripción) y el implementador (Ver descripción) y al culminar las mismas quedarán implementadas como servicios las funcionalidades que fueron definidas en la planilla de Contrato de Servicios a Exportar.

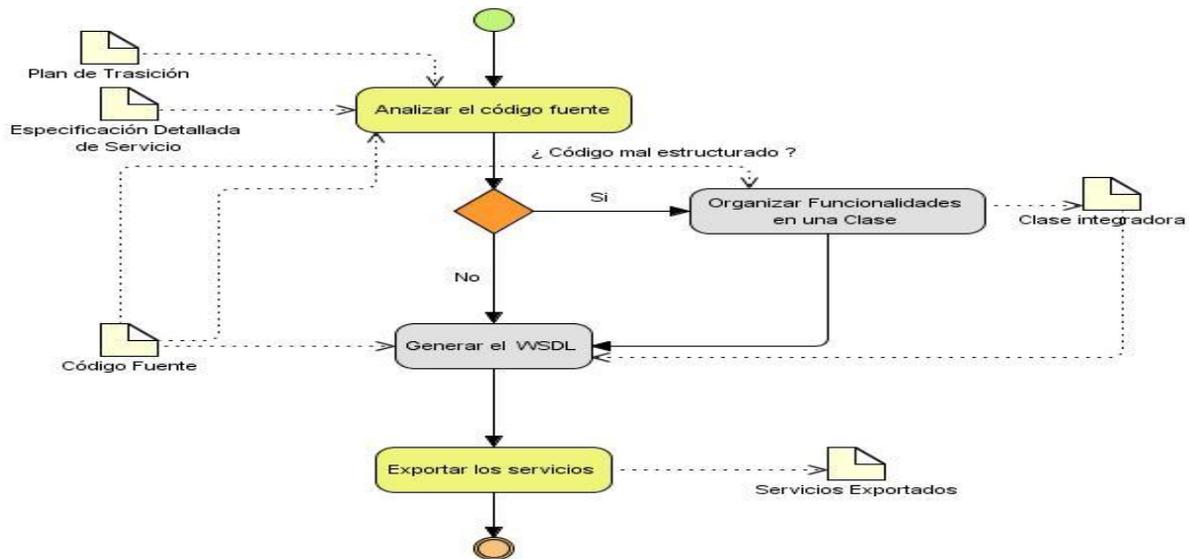


Figura 8: Flujo de actividades del subproceso Exportación de Servicios en PHP

La Figura 8 muestra la representación más general del flujo de actividades del subproceso, con el conjunto de los artefactos que intervienen en el flujo.

2.7.5.1 Analizar el código fuente



Roles

- Desarrollador



Técnicas y Herramientas

- Tormenta de Ideas
- Taller



Artefactos de entrada

- Código fuente
- Especificación Detallada de Servicio
- Plan de Transición



Artefactos de salida

- No posee artefacto de salida

El paso inicial que se debe dar para exportar servicios en PHP es analizar detenidamente el código fuente en que está desarrollado el sistema legado, que ya fue obtenido como premisa para la aplicación de procedimiento.

Para la realización de esta actividad el grupo se debe reunir para obtener una visión general de cómo está implementado el sistema legado. El objetivo fundamental es definir si las funcionalidades que serán exportadas como servicios están contenidas en una misma clase o dispersas en varias clases de la aplicación que se está analizando. Es por ello que se emplean las técnicas de tormenta de ideas y talleres para garantizar que todo el grupo trabaje bajo la misma línea.

El rol encargado de realizar esta actividad es el desarrollador (Ver descripción).

Como artefactos de entrada se tienen el código fuente del sistema legado, la planilla de Especificación Detallada de Servicio y el Plan de Transición.

2.7.5.2 Organizar funcionalidades en una clase



Roles

- Desarrollador
- Implementador



Técnicas y Herramientas

- Zend Studio-5.5.0



Artefactos de entrada

- Código fuente



Artefactos de salida

- Clase integradora

Esta actividad se realiza en caso de que las funcionalidades que serán exportadas como servicios no se encuentren contenidas en una misma clase. Es por ello que surge la necesidad de crear una nueva clase que encierre dichas funcionalidades.

Los roles encargados de realizar esta actividad es el desarrollador (Ver descripción) y el implementador (Ver descripción).

Como artefacto de entrada se utiliza el código fuente y como artefacto de salida se obtiene la clase integradora la cual contiene todas las funcionalidades que se desean exportar como servicios que estaban dispersas en varias clases del sistema legado.

2.7.5.3 Generar el WSDL



Roles

- Desarrollador
- Implementador



Técnicas y Herramientas

- Zend Studio-5.5.0
- Servidor Web que soporte PHP (Server2Go)



Artefactos de entrada

- Código fuente
- Clase integradora



Artefactos de salida

- No posee artefacto de salida

Capítulo 2. Descripción de la solución propuesta

Los roles encargados de realizar esta actividad son el desarrollador (Ver descripción) y el implementador (Ver descripción). Como artefactos de entrada se utiliza el código fuente del sistema legado en caso que las funcionalidades que se desean exportar como servicios se encuentran bien definidas en una misma clase, de no ser así el artefacto que sirve de entrada a esta actividad es la clase integradora que se fue creada en la actividad anterior.

Para generar el servicio web con sus funciones lo primero que se debe hacer es seleccionar la clase donde están definidas las funcionalidades que serán exportadas como servicios ya sea en el código fuente original o en la clase integradora que se creó en la actividad anterior.

Luego de tener la clase con todas las funcionalidades a exportar como servicios cargada en el Zend Studio, se crea el WSDL, para ello se crea un nuevo Documento de Texto en el mismo directorio donde está dicha clase. En principio este documento tendrá extensión .txt la cual será sustituida por .wsdl. Una vez creado este fichero se hace clic en la opción de menú **Herramientas** y se selecciona la opción **Generador WSDL**. (Ver anexo 5). Se mostrará posteriormente una ventana que es el asistente para generar WSDL donde se especifica el nombre del fichero, el cual debe coincidir con el nombre del fichero creado y la dirección donde se publicará el servicio web. En caso de que en el campo Nombre de la configuración no salga una por defecto, se crea haciendo clic sobre el +, se hace una configuración diferente para cada servicio. (Ver anexo 6).

Una vez especificados todos los campos de esta ventana se pulsa el botón Siguiente y se mostrará una nueva ventana donde se especifica el fichero contenedor de las clases que se quieren exportar como servicios. Dicho fichero tiene extensión .php. (Ver anexo 7).

Después de haber seleccionado las clases que se desean exportar como servicios se pulsa el botón Siguiente y a continuación se muestra una ventana resumen de la generación con un botón Terminar, que al hacer clic sobre este se crea el código del .wsdl en el fichero seleccionado anteriormente. (Ver anexo 8).

2.7.5.4 Exportar servicios



Roles

- Desarrollador
- Implementador



Técnicas y Herramientas

- Zend Studio-5.5.0
- Servidor Web que soporte PHP (Server2Go)



Artefactos de entrada

- No posee artefactos de entrada



Artefactos de salida

- Servicios exportados

Cuando se ejecuta el servidor Server2Go se abrirá la página web de inicio del mismo. Para ver a los servicios que están disponibles en ese servidor se accede a la URL, en principio saldrá de la siguiente forma `http://127.0.0.1:4001/`, luego se pondría `http://127.0.0.1:4001/www/`, y posteriormente se accedería al WSDL para ver todas las funcionalidades exportadas como servicio.

Los roles encargados de realizar esta actividad son el desarrollador (Ver descripción) y el implementador (Ver descripción). Como artefacto de salida de esta actividad se obtiene los servicios exportados.

2.7.6 Probar los servicios



Roles

- Desarrollador
- Implementador



Técnicas y Herramientas

- SoapUI 2.0.1

Capítulo 2. Descripción de la solución propuesta



Artefactos de entrada

- Servicios exportados



Artefactos de salida

- No posee artefacto de salida

Para aplicar correctamente esta actividad es necesario obtener el WSDL de las funcionalidades que se han exportado como servicios y que el mismo se encuentre en ejecución.

En esta actividad se realizarán un conjunto de pruebas con el fin de garantizar que los servicios exportados tengan la calidad óptima. Para ello se utilizó la herramienta SoapUI en su versión 2.0.1, la cual es una herramienta de software libre gráfica, está basada en Java y sirve para el testeo de Servicios Web y generación de clientes de Servicios Web. SoapUI permite testear Servicios Web de forma fácil y ver los resultados.

Al trabajar con Servicios Web sin interfaces gráficas, esta herramienta permite automatizar fácilmente las pruebas a los servicios exportados y así asegurar la calidad de los mismos.

Las pruebas a los Servicios Web que podrían usarse para más de un propósito son las siguientes:

- *Pruebas unitarias*: para validar que cada operación de los servicios funciona como se definió.
- *Prueba de aceptación*: para validar que el servicio retorna resultados aceptables según los requerimientos.
- *Pruebas de manejo de datos*: para validar el comportamiento con las entradas de datos externos al sistema (bases de datos, otros sistemas, uso de otros Servicios Web).

Una vez que se tiene la herramienta SoapUI, se crea un nuevo proyecto donde se configura la URL que contiene el WSDL que será sometido a pruebas. (Ver anexo 9).

Por cada operación del Servicios Web a probar se crea una petición. A continuación se ilustra un ejemplo de cómo se muestran las peticiones que se crean en la herramienta SoapUI.

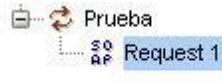


Figura 9: Petición en la herramienta SoapUI

Cada petición requiere el ingreso de valores en caso de que el servicio exportado posea parámetros de entrada. Estos valores se agregan en el XML que propone la herramienta. (Ver anexo 10).

Una vez que se hayan introducido todos los valores en el XML se procede con la ejecución de la prueba dando clic en la flecha verde que se encuentra en la parte superior izquierda del XML. En la parte derecha del XML se mostrarán los valores que esta funcionalidad debe retornar.

Los roles encargados de la realización de esta actividad son el desarrollador (Ver descripción) y el implementador (Ver descripción).

Como artefacto de entrada se emplea el WSDL de los servicios exportados.

Conclusiones parciales

El presente capítulo abordó la confección del procedimiento. Se define el alcance del mismo, las premisas y principios para su aplicación. Se define una representación del procedimiento, se describen los subprocesos y actividades que lo componen. Se detallan las técnicas y herramientas utilizadas en cada uno de los subprocesos y actividades. Se especifican los roles que interactúan con el procedimiento. Se detallan los objetivos de los artefactos propuestos en el procedimiento, así como un resumen de estos artefactos donde se especifica si son entradas o salidas de algún subproceso o actividad.

CAPÍTULO 3: VALIDACIÓN DE LA SOLUCIÓN PROPUESTA

Introducción

En este capítulo se valida el procedimiento propuesto en el capítulo anterior mediante el Método Delphi. Para ello se decidió conformar un Panel de Expertos que emitiera su criterio acerca de la adecuación del procedimiento.

3.1 Los métodos de expertos

Cuando se realiza una investigación uno de los principales inconvenientes que tiene asociado es la posibilidad de verificar y demostrar la confiabilidad de su propuesta resultante. Ante esta situación se emplean los métodos de expertos para validar los resultados arrojados por dicha investigación. Estos métodos utilizan como fuente de información un grupo de personas que poseen cierto grado de conocimiento de la materia en cuestión.

3.1.1 Ventajas de los métodos de expertos

- El conjunto de opiniones de las personas que se consulten es siempre más valioso que la opinión individual de la persona mejor preparada, aunque la de esta última contraste con las anteriores. Esta afirmación se basa en la idea de que varias cabezas son mejor que una.
- Un grupo de personas siempre tendrá en cuenta mayor número de factores para realizar la evaluación que los que considerará una sola persona. Cada experto podrá aportar a la discusión general la idea que tiene sobre el tema debatido desde su área de conocimiento. [21]

3.1.2 Desventajas de los métodos de expertos

Independientemente de las ventajas que traen los métodos de expertos es importante señalar sus principales desventajas las cuales constituyen un factor de riesgo que se debe tener en cuenta a la hora de aplicar dichos métodos. Algunos inconvenientes son:

- La presión social que el grupo ejerce sobre sus participantes puede provocar acuerdos con la mayoría, aunque la opinión de ésta sea errónea. Así, un experto puede renunciar a la defensa de su opinión ante la persistencia del grupo en rechazarla.
- En estos grupos hay ocasiones en que el argumento que triunfa es el más citado, aunque no sea el más válido.
- Estos grupos son vulnerables a la posición y personalidad de algunos de los individuos. Una persona con dotes de comunicador puede convencer al resto de individuos, aunque su opinión no sea la más acertada. Esta situación se puede dar también cuando uno de los expertos ocupe un alto cargo en la organización, ya que sus subordinados no le rebatirán sus argumentos con fuerza.

[21]

El método de expertos ideal sería aquel que extrajese los beneficios de la interacción directa entre un grupo de especialistas de una rama determinada y eliminase sus inconvenientes. Esta intenta ser la filosofía de la metodología Delphi la cual se ofrece una breve explicación en el siguiente epígrafe.

3.2 El Método Delphi

El método Delphi, cuyo nombre se inspira en el antiguo oráculo de Delphos, parece que fue ideado originalmente a comienzos de los años 50 en el seno del Centro de Investigación estadounidense RAND Corporation por Olaf Helmer y Theodore J. Gordon, como un instrumento para realizar predicciones sobre un caso de catástrofe nuclear. Desde entonces, ha sido utilizado frecuentemente como sistema para obtener información sobre el futuro. [22]

El Delphi es uno de los métodos de pronosticación más confiables, constituye un procedimiento para confeccionar un cuadro de la evolución de situaciones complejas, a través de la elaboración estadística de las opiniones de un grupo de expertos en el tema tratado. Permite rebasar el marco de las condiciones actuales más señaladas de un fenómeno y alcanzar una imagen integral y más amplia de su posible evolución, reflejando las valoraciones individuales de los expertos que pueden estar basadas en un análisis lógico, como en su experiencia intuitiva. [21]

Una Delphi consiste en la selección de un grupo de expertos a los que se les pregunta su opinión sobre cuestiones referidas a acontecimientos del futuro. Las estimaciones de los expertos se realizan en

sucesivas rondas, anónimas, al objeto de tratar de conseguir consenso, pero con la máxima autonomía por parte de los participantes. [22]

El método se basa en la organización de un diálogo anónimo entre los expertos consultados de modo individual, a partir de la aplicación de un cuestionario y con el propósito de obtener un consenso general o los motivos discrepantes entre estos. Los expertos, seleccionados previamente, se someten a una serie de interrogantes sucesivas, cuyas respuestas se procesan estadísticamente para conocer la coincidencia o discrepancia que estos tienen en cuanto a lo consultado. Este proceso iterativo, en el que en cada cuestionario se informa los resultados del precedente, posibilita al experto modificar sus respuestas anteriores, en función de los elementos aportados por otros expertos. [21]

3.2.1 Características del Método Delphi

El método Delphi pretende extraer y maximizar las ventajas que presentan los métodos basados en grupos de expertos y minimizar sus inconvenientes. Para ello se aprovecha la sinergia del debate en el grupo y se eliminan las interacciones sociales indeseables que existen dentro de todo grupo. De esta forma se espera obtener un consenso lo más fiable posible del grupo de expertos.

Este método presenta tres características fundamentales:

- *Anonimato*: Durante un Delphi, ningún experto conoce la identidad de los otros que componen el grupo de debate. Esto tiene una serie de aspectos positivos, como son:
 - ✓ Impide la posibilidad de que un miembro del grupo sea influenciado por la reputación de otro de los miembros o por el peso que supone oponerse a la mayoría. La única influencia posible es la de la congruencia de los argumentos.
 - ✓ Permite que un miembro pueda cambiar sus opiniones sin que eso suponga una pérdida de imagen.
 - ✓ El experto puede defender sus argumentos con la tranquilidad que da saber que en caso de que sean erróneos, su equivocación no va a ser conocida por los otros expertos.
- *Iteración y realimentación controlada*: La iteración se consigue al presentar varias veces el mismo cuestionario. Como, además, se van presentando los resultados obtenidos con los cuestionarios anteriores, se consigue que los expertos vayan conociendo los distintos puntos de vista y puedan ir modificando su opinión si los argumentos presentados les parecen más apropiados que los suyos.

- *Respuesta del grupo en forma estadística:* La información que se presenta a los expertos no es sólo el punto de vista de la mayoría, sino que se presentan todas las opiniones indicando el grado de acuerdo que se ha obtenido. [23]

3.3 Aplicación del método

El método consta de cuatro fases fundamentales, las cuales se exponen a continuación:

1. Elección de expertos.
2. Elaboración del cuestionario, para validación de la propuesta.
3. Establecimiento de la concordancia de los expertos mediante el uso del coeficiente de Kendall.
4. Desarrollo práctico y explotación de resultados.

3.3.1 Elección de los expertos

Se define como experto el individuo en sí, grupo de personas u organizaciones capaces de ofrecer valoraciones conclusivas de un problema en cuestión y hacer recomendaciones respecto a sus momentos fundamentales con un máximo de competencia. [24]

Los expertos fueron seleccionados teniendo en cuenta el cumplimiento de los siguientes criterios:

- Graduado de Nivel Superior.
- Un año de experiencia como mínimo.
- Vinculación al desarrollo de productos informáticos.
- Conocimientos sobre las Arquitecturas Orientadas a Servicios (SOA).
- Conocimientos acerca de implementación de servicios para una SOA.

Para aplicar este método no existe una norma generalizada para determinar el número de experto que podrán intervenir en proceso de validación, se recomienda que dicho método tenga entre siete y treinta expertos debido a que con siete el error se disminuye exponencialmente y con más de treinta aunque siga disminuyendo es de forma poco significativa y no compensa el incremento de costo y esfuerzo.

Inicialmente se seleccionaron doce posibles expertos dentro de la propia Universidad de las Ciencias Informáticas (UCI) para que formaran parte del panel de expertos, de los cuales, siete compañeros estuvieron de acuerdo en participar en la validación del procedimiento. A los mismos se le aplicó una encuesta con el objetivo de determinar sus coeficientes de competencia y recopilar información detallada sobre la labor que desempeñan, los años de experiencia en el tema, la calificación profesional y la categoría docente. Para acceder a esta encuesta. (Ver Complementario 4)

3.3.1.1 Cálculo del coeficiente de competencia

Al obtener los resultados de la encuesta aplicada al panel de experto se procede con el cálculo del coeficiente de competencia. Este coeficiente se determina mediante la fórmula: $K = \frac{1}{2} (k_c + k_a)$, donde k_c es el coeficiente de conocimientos y k_a es el coeficiente de argumentación.

El coeficiente de conocimiento (k_c) se obtiene de la primera tabla de la encuesta, la cual recoge una autoevaluación del posible experto. (Ver Complementario 4, Tabla 1)

El posible experto marcará en la casilla enumerada según su criterio acerca de la capacidad que tiene sobre el tema que se ha sometido a su consideración, en una escala del uno al diez, considerando uno como no tener ningún conocimiento y diez el de pleno conocimiento de la problemática tratada. Después para ajustarla a la teoría de las probabilidades se multiplicará por 0,1 el valor de la casilla seleccionada.

El coeficiente de argumentación (k_a) se calcula de la siguiente forma: Se utiliza la tabla correspondiente a la segunda pregunta del cuestionario. (Ver Complementario 4, Tabla 2)

El experto debe marcar, según su criterio, su grado de competencia sobre los aspectos sometidos a su consideración. Esta selección se traduce a puntos, según la siguiente escala. (Ver Complementario 7, Tabla 1)

El coeficiente de argumentación (k_a) será igual a la suma de los valores donde el posible experto haya marcado.

Al tener calculado k_c y k_a ya se cuenta con los datos suficientes para obtener el coeficiente de competencia K .

Intervalos para determinar el nivel del coeficiente de competencia (K):

Si $0,8 < k < 1,0$ el coeficiente de competencia es alto.

Si $0,5 < k < 0,8$ el coeficiente de competencia es medio.

Si $k < 0,5$ el coeficiente de competencia es bajo.

Los expertos seleccionados para formar parte del panel de validación de la propuesta tienen que ser aquellos cuyos resultados arrojaron un coeficiente de competencia alto o medio.

De los siete expertos a los que se les aplicó la encuesta de autoevaluación, todos arrojaron un coeficiente de competencia alto o medio por lo que fueron seleccionados para continuar con la ejecución del método. (Ver Complementario 7, Tabla 2)

Finalmente el panel de expertos quedó conformado con los siguientes integrantes. (Ver Complementario 5)

3.3.2 Elaboración del cuestionario, para validación de la propuesta.

Cuando se culmina con la selección del panel de expertos, se prosigue con la elaboración de la encuesta para la validación del procedimiento. Para ello se conforma un cuestionario el cual no debe tener demasiadas preguntas, pero el mismo debe abordar aspectos medulares sobre la investigación realizada.

El cuestionario fue creado de forma tal que las respuestas fueran categorizadas en Muy adecuado (C1), Bastante adecuado (C2), Adecuado (C3), Poco adecuado (C4) y No adecuado (C5).

Para acceder al cuestionario para la validación del procedimiento. (Ver Complementario 6)

3.3.3 Establecimiento de la concordancia de los expertos mediante el uso del coeficiente de Kendall

Para que la propuesta tenga una mayor validez es necesario que exista un adecuado acuerdo entre los integrantes del panel de expertos, esto se comprueba mediante el cálculo del Coeficiente de Concordancia de Kendall el cual ayuda a precisar el grado de coincidencia de las valoraciones arrojadas por los expertos.

Para determinar este coeficiente se construye una tabla de aspectos a evaluar contra expertos, estos datos fueron obtenidos al aplicar la encuesta para la validación del procedimiento. Luego de haber confeccionado dicha tabla, se siguen los siguientes pasos:

Capítulo 3. Validación de la solución propuesta

Paso 1: Determinar la suma de los valores numéricos asignados a cada aspecto a evaluar, según el criterio dado por cada experto (R_j). (Ver Complementario 7, Tabla 3)

Paso 2: Determinar el valor medio de las R_j , dado por la sumatoria de los R_j entre N , siendo N el total de aspectos a evaluar (los aspectos serán las preguntas del cuestionario, en este caso $N = 18$).

$$\overline{R_j} = \frac{\sum_{j=1}^n R_j}{N}$$

Sustituyendo los valores en la fórmula se obtiene el siguiente resultado:

$$\overline{R_j} = \frac{5(30) + 3(34) + 3(33) + 3(29) + 2(28) + 31 + 26}{18}$$

$$\overline{R_j} = 30.61$$

Paso 3: Determinación de la desviación media S , dada por la diferencia entre cada R_j y el valor de la media al cuadrado. Para calcular el valor de S se emplea la siguiente fórmula:

$$S = \sum_{j=1}^n (R_j - \overline{R_j})^2$$

Al sustituir los valores en la fórmula se obtiene como resultado que **$S = 96.28$**

Paso 4: Determinación del cuadrado del número total de expertos, K . En este caso $K = 7$, por lo que **$K^2 = 49$** .

Paso 5: Determinación del cubo del número total de aspectos a evaluar, N . En este caso $N = 18$, por lo que **$N^3 = 5832$** .

Paso 6: Determinación de la diferencia N^3 y N y su multiplicación por K^2 . Obteniéndose como resultado: **81396**.

Una vez obtenidos todos los valores necesarios para el cálculo de W se sustituyen dichos valores en la siguiente fórmula:

$$W = \frac{12 * S}{K^2(N^3 - N)}$$

Sustituyendo los valores en la fórmula se obtiene el siguiente resultado:

$$W = \frac{12 * 96.28}{284886} = 0.004$$

El coeficiente W siempre tiene que ser positivo y oscilar en el intervalo entre 0 y 1. Con este valor, se puede calcular Chi cuadrado real, con el propósito de verificar si existe o no concordancia entre los expertos. Chi cuadrado real se calcula de la siguiente forma:

$$X^2 = K(N - 1)W$$

Sustituyendo los valores en la fórmula se obtiene el siguiente resultado:

$$X^2 = 7(18 - 1)0.004 = 0.476$$

El resultado de X^2 obtenido se compara con el de las tablas estadísticas de Siegel, y si: $X^2 \text{ real} < X^2(\alpha, N - 1)$ existe una concordancia entre el panel de expertos.

Para $\alpha = 0.1$ y $(N - 1) = 17$, quedaría que $0.476 < 24.7690$, por lo que se llega a la conclusión concordancia entre los expertos.

3.3.4 Desarrollo práctico y explotación de resultados

Los expertos que integraron el panel recibieron un resumen de la propuesta de solución para responder la encuesta de validación del procedimiento. Esta encuesta fue enviada vía correo electrónico y en algunos casos fue entregada en formato duro. Se realizó una sola ronda de preguntas y al recibir los resultados se prosiguió a analizarlos.

Para la recopilación y visualización de los resultados aportados por el panel de expertos se confeccionaron tablas. Para la confección de las mismas se utilizó el Microsoft Excel 2007.

Los resultados se almacenaron en la Tabla de Frecuencias Acumuladas (Ver Complementario 7, Tabla 4), los cuales quedan representados gráficamente en la siguiente figura:

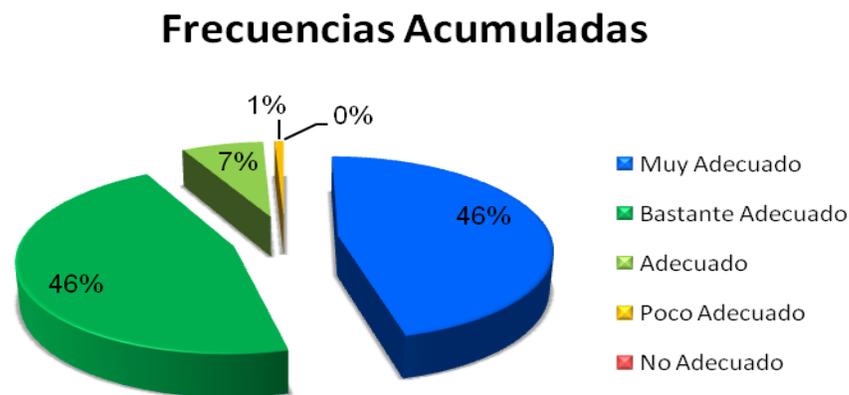


Figura 10: Frecuencias Acumuladas

Luego de tener todos los datos almacenados en la tabla de Frecuencias Acumuladas, se siguen una serie de pasos para obtener los demás resultados:

Paso 1: Se construye una tabla de Frecuencias Absolutas Acumuladas (Ver Complementario 7, Tabla 5) donde cada número en la fila (excepto el primero) se obtiene sumándole el anterior. Como se puede observar, en esta tabla desaparece la última columna.

Paso 2: Se copia la tabla anterior y se borran los resultados numéricos. En esta nueva tabla, se construye la tabla de Frecuencias Relativas Acumulativas (Ver Complementario 7, Tabla 6). Esta nueva tabla se obtiene dividiendo por 7 (cantidad de expertos) cada uno de los números de la tabla anterior. En esta tabla queda eliminada una columna pues hay 5 categorías y sólo se necesitan cuatro puntos de corte (con cuatro puntos se obtienen 5 intervalos).

Paso 3: Se buscan las imágenes de los elementos de la tabla anterior por medio de la función (DISTR.NORM.ESTAND.INV). Para esto se crea una nueva tabla a la cual se le adicionan tres columnas y una fila para colocar los resultados que se mencionan a continuación:

- Suma de las columnas.
- Suma de filas.
- Promedio de las columnas.

- Los promedios de las filas se obtienen de forma similar, en este caso también se divide por cuatro porque quedan cuatro categorías ya que la última se eliminó.
- Para hallar N, se divide la suma de las sumas entre el resultado de multiplicar el número de indicadores por el número de preguntas.
- El valor N-P da el valor promedio que otorgan los expertos para cada indicador propuesto.

Los puntos anteriores se resumen en la Tabla Puntos de Corte. (Ver Complementario 7, Tabla 7)

Las sumas obtenidas en las cuatro primeras columnas de la tabla anterior dan los puntos de corte. Al contar con cuatro puntos de corte se obtienen cinco intervalos (Ver Complementario 7, Tabla 8). Estos se utilizan para determinar la categoría o grado de adecuación de cada criterio según la opinión de los expertos consultados.

Los puntos de corte quedan representados gráficamente en la siguiente figura:

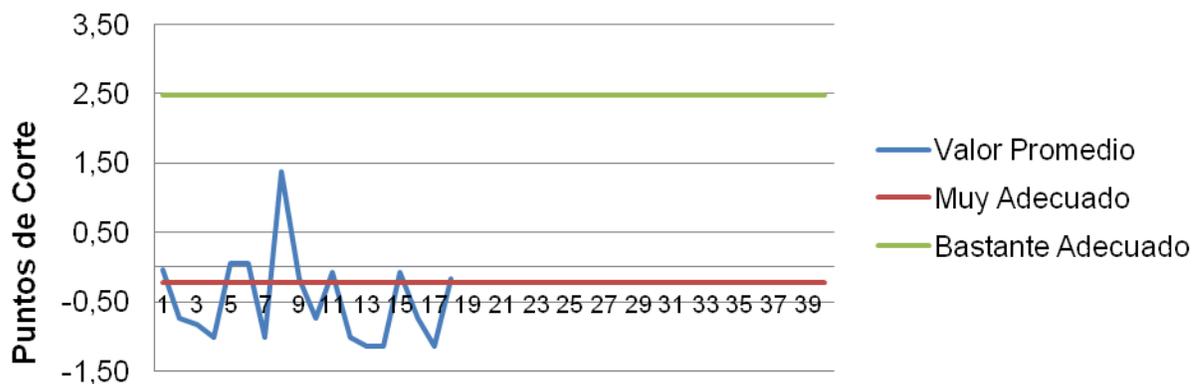


Figura 11: Puntos de Corte

Como resultado final se obtiene el grado de adecuación de los elementos sometidos a la valoración de los expertos. (Ver Complementario 7, Tabla 9)

Por los resultados arrojados por la tabla anterior se puede llegar a la conclusión que el procedimiento sometido a validación por el panel de expertos resultó Muy Adecuado y Bastante Adecuado. A continuación se representa gráficamente los resultados obtenidos:

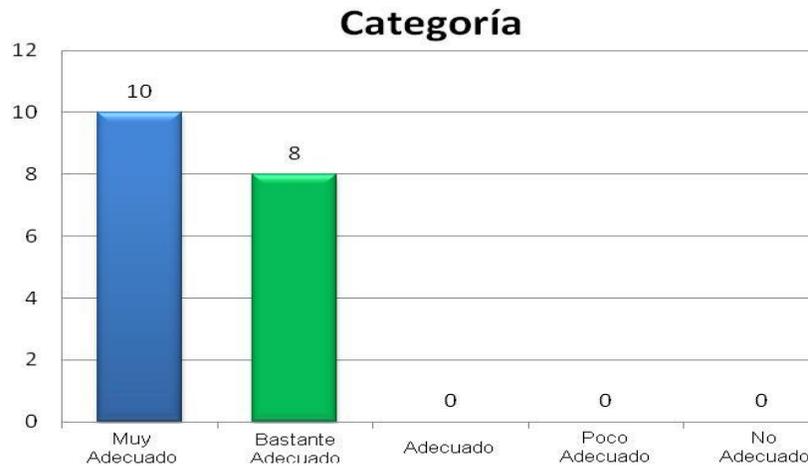


Figura 12: Resultados finales de la validación

Conclusiones parciales

En este capítulo se aplicó el Método Delphi para la validación del procedimiento para la integración de sistemas legados en organizaciones empresariales con arquitecturas orientadas a servicios. Para ello se confeccionó un panel de expertos de siete miembros, los cuales fueron seleccionados teniendo en cuenta el cumplimiento de los siguientes criterios: graduado de nivel superior, un año de experiencia como mínimo, vinculación al desarrollo de productos informáticos, conocimientos sobre las arquitecturas orientadas a servicios (SOA) y conocimiento acerca de implementación de servicios para una SOA. Por los resultados arrojados en la validación se llega a la conclusión que el procedimiento sometido al criterio del panel de expertos resultó Muy Adecuado y Bastante Adecuado.

CONCLUSIONES GENERALES

Para la disciplina de implementación de servicios del CDAE, es de gran importancia contar con un procedimiento que sirva de guía a la hora de enfrentarse a la ejecución de un proyecto sobre una arquitectura orientada a servicios, puesto que el mercado SOA está ampliamente liderado por empresas desarrolladoras de software propietario las cuales no ofrecen gratuitamente su información. Con este trabajo, se logró realizar una propuesta de un procedimiento para la integración de sistemas legados en organizaciones empresariales con arquitecturas orientadas a servicios y por tanto darle cumplimiento al objetivo principal de la investigación. El procedimiento propuesto detalla los roles que intervienen en el mismo, especifica las técnicas y herramientas utilizadas en cada uno de los subprocesos y actividades, así como un resumen de los artefactos de entrada o salida. El procedimiento fue validado por un panel de expertos mediante el Método Delphi arrojando un resultado satisfactorio puesto que todos los aspectos sometidos al criterio de los expertos fueron de Muy Adecuado y Bastante Adecuado.

RECOMENDACIONES

Partiendo de que con el desarrollo del presente trabajo se dio cumplimiento a su objetivo general se recomienda lo siguiente:

- Poner en práctica el presente procedimiento en los proyectos futuros del Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE).
- Capacitar al equipo de la disciplina de implementación de servicios del Centro de Consultoría y Desarrollo de Arquitecturas Empresariales sobre los temas tratados en la propuesta para que puedan dar cumplimiento de manera efectiva a los subprocesos y actividades que se definen en el procedimiento.

REFERENCIAS BIBLIOGRÁFICAS

1. MICROSOFT CORPORATION. Whitepaper: La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada al mundo real. Diciembre, 2006.
2. IBM. IBM, empresa líder en tecnologías de la información. [en línea] (2009). [Consultado el 25 de enero de 2010]. Disponible en: <<http://www-05.ibm.com/es/ibm/history/>>.
3. PEÑA, José Ramón. IBM SOA. [en línea] (2010). [Consultado el 25 de enero de 2010]. Disponible en: <http://computerworld.net.ve/index2.php?option=com_content&do_pdf=1&id=78>
4. MICROSOFT. Historia de Microsoft. [en línea] (2003). [Consultado el 25 de enero de 2010]. Disponible en: <<http://www.maestrosdelweb.com/editorial/microsoft/>>
5. MICROSOFT CORPORATION. Whitepaper: La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada al mundo real. Diciembre, 2006.
6. ORACLE. Oracle Suite en las Mejores y Más Conocidas Tecnologías de Oracle. [en línea]. [Consultado el 27 de enero de 2010]. Disponible en: <<http://www.oracle.com/global/lad/technologies/soa/soa-suite.html>>
7. CASANOVAS, Josep. Qué es la Arquitectura del Software y qué relación tiene con la usabilidad. Tener en cuenta la usabilidad en el momento del diseño de la arquitectura de un sistema interactivo nos puede ahorrar muchos quebraderos de cabeza. [en línea] (2004). [Consultado el 20 de enero de 2010]. Disponible en: <<http://www.desarrolloweb.com/articulos/1622.php>>
8. MICROSOFT CORPORATION. Whitepaper: La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada al mundo real. Diciembre, 2006.
9. DE SETA, Leonardo. Introducción a los servicios web RESTful. [Consultado el 24 de febrero de 2010]. [en línea] (2008). Disponible en: <<http://www.dosideas.com/noticias/java/314-introduccion-a-los-servicios-web-restful.html>>
10. GONZÁLEZ C., Benjamín. WSDL para la documentación de Servicios Web. [en línea] (2004). [Consultado el 7 de mayo de 2010]. Disponible en: <<http://www.desarrolloweb.com/articulos/1581.php>>

11. ÁLVAREZ, Miguel Angel. Descripción y características de este potente y moderno lenguaje de programación. [en línea] (2001). [Consultado el 19 de marzo de 2010]. Disponible en: <<http://www.desarrolloweb.com/articulos/497.php>>
12. ZÁRATE MEJÍA, Eduardo. La plataforma Java. [en línea]. [Consultado el 19 de marzo de 2010]. Disponible en: <http://profesores.fi-b.unam.mx/carlos/java/java_basico1_1.html>
13. GARCÍA, Angel Franco. Fundamentos del Lenguaje Java. La Máquina Virtual Java. [en línea]. [Consultado el 19 de marzo de 2010]. Disponible en: <<http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/introduccion/virtual.htm>>
14. ÁLVAREZ, Miguel Angel. Llegó el momento de hablar sobre HTML, el lenguaje con el que crean las páginas web. [en línea]. [Consultado el 19 de marzo de 2010]. Disponible en: <<http://www.desarrolloweb.com/articulos/que-es-html.html>>
15. CHÁVEZ, Carlos. Introducción a XML. [en línea] (2005). [Consultado el 19 de marzo de 2010]. Disponible en: <<http://www.maestrosdelweb.com/editorial/flashxml/>>
16. W3C. Scripts. [en línea] (2001). [Consultado el 19 de marzo de 2010]. Disponible en: <<http://html.conclase.net/w3c/html401-es/interact/scripts.html>>
17. GALLI, Ricardo. Qué es .NET? [en línea] (2004). [Consultado el 19 de marzo de 2010]. Disponible en: <<http://globaliza.blogia.com/temas/tecnologia.net.php>>
18. GONZÁLEZ SECO, José Antonio. El lenguaje de programación C#. Octubre 2006.
19. ÁLVAREZ, Miguel Angel. Es un lenguaje para programar scripts del lado del servidor, que se incrustan dentro del código HTML. Este lenguaje es gratuito y multiplataforma. [en línea]. [Consultado el 14 de mayo de 2010]. Disponible en: <<http://www.desarrolloweb.com/articulos/392.php>>
20. WSO2. WSO2 Web Services Application Server. [en línea]. [Consultado el 10 de mayo de 2010]. Disponible en: <<http://wso2.com/products/web-services-application-server/>>
21. ORTEGA HERNÁNDEZ, Giselle y CHIRINO ESQUIJAROSA, Dariel. Propuesta de modelo para la gestión de la gobernabilidad durante la fase de diseño en un proyecto BPM/SOA. Junio 2009.
22. ASTIGARRAGA, Eneko. El método Delphi.

23. El método Delphi. [en línea]. [Consultado el 19 de marzo de 2010]. Disponible en: <<http://www.gtíc.ssr.upm.es/encuestas/delphi.htm>>
24. MORÁGUEZ IGLESIAS, Ing. Arabel. El método Delphi. [en línea] (2006). [Consultado el 12 de mayo de 2010]. Disponible en: <<http://www.gestiopolis.com/canales6/eco/metodo-delphi-estadistica-de-investigacion-cientifica.htm>>

BIBLIOGRAFÍA

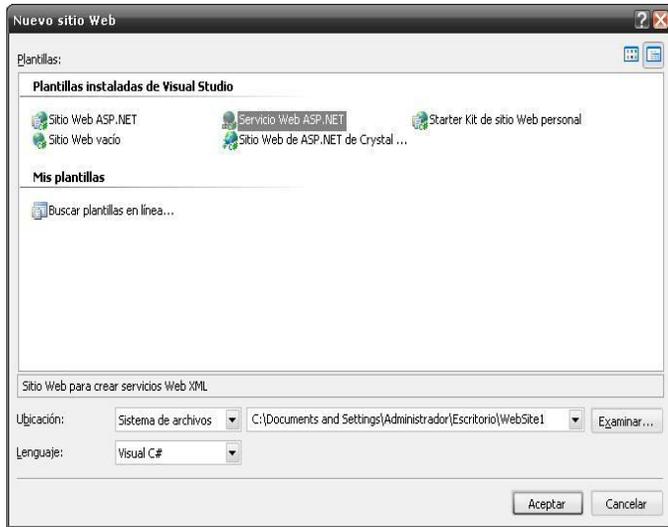
1. ÁLVAREZ, Miguel Angel. Descripción y características de este potente y moderno lenguaje de programación. [en línea] (2001). Disponible en: <<http://www.desarrolloweb.com/articulos/497.php>>
2. ÁLVAREZ, Miguel Angel. Es un lenguaje para programar scripts del lado del servidor, que se incrustan dentro del código HTML. Este lenguaje es gratuito y multiplataforma. [en línea]. Disponible en: <<http://www.desarrolloweb.com/articulos/392.php>>
3. ÁLVAREZ, Miguel Angel. Llegó el momento de hablar sobre HTML, el lenguaje con el que crean las páginas web. [en línea]. Disponible en: <<http://www.desarrolloweb.com/articulos/que-es-html.html>>
4. ASTIGARRAGA, Eneko. El método Delphi.
5. CASANOVAS, Josep. Qué es la Arquitectura del Software y qué relación tiene con la usabilidad. Tener en cuenta la usabilidad en el momento del diseño de la arquitectura de un sistema interactivo nos puede ahorrar muchos quebraderos de cabeza. [en línea] (2004). Disponible en: <<http://www.desarrolloweb.com/articulos/1622.php>>
6. CHÁVEZ, Carlos. Introducción a XML. [en línea] (2005). Disponible en: <<http://www.maestrosdelweb.com/editorial/flashxml/>>
7. DE PEDRO, José Pablo. Cómo empezar en SOA ganando en rentabilidad. [en línea] (2009). Disponible en: <<http://www.pcactual.com/Actualidad/Noticias/Como-empezar-en-SOA-ganando-en-rentabilidad>>
8. DE SETA, Leonardo. Introducción a los servicios web RESTful. [en línea] (2008). Disponible en: <<http://www.dosideas.com/noticias/java/314-introduccion-a-los-servicios-web-restful.html>>
9. El método Delphi. [en línea]. Disponible en: <<http://www.gtlic.ssr.upm.es/encuestas/delphi.htm>>
10. GALLI, Ricardo. Qué es .NET? [en línea] (2004). Disponible en: <<http://globaliza.blogia.com/temas/tecnologia.net.php>>

11. GARCÍA, Angel Franco. Fundamentos del Lenguaje Java. La Máquina Virtual Java. [en línea]. Disponible en: <http://www.sc.ehu.es/sbweb/fisica/cursoJava/fundamentos/introduccion/virtual.htm>
12. GONZÁLEZ C., Benjamín. WSDL para la documentación de Servicios Web. [en línea] (2004). Disponible en: <http://www.desarrolloweb.com/articulos/1581.php>
13. GONZÁLEZ SECO, José Antonio. El lenguaje de programación C#. Octubre 2006.
14. IBM. IBM, empresa líder en tecnologías de la información. [en línea] (2009). Disponible en: <http://www-05.ibm.com/es/ibm/history/>.
15. MARTÍNEZ TORRES, Yeilin. Propuesta para el Levantamiento de Requisitos Funcionales en función del Aseguramiento de la Calidad de Software. Junio, 2009.
16. MICROSOFT CORPORATION. Whitepaper: La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada al mundo real. Diciembre, 2006.
17. MICROSOFT CORPORATION. Whitepaper: La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada al mundo real. Diciembre, 2006.
18. MICROSOFT CORPORATION. Whitepaper: La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada al mundo real. Diciembre, 2006.
19. MICROSOFT. Historia de Microsoft. [en línea] (2003). Disponible en: <http://www.maestrosdelweb.com/editorial/microsoft/>
20. MORÁGUEZ IGLESIAS, Ing. Arabel. El método Delphi. [en línea] (2006). Disponible en: <http://www.gestiopolis.com/canales6/eco/metodo-delphi-estadistica-de-investigacion-cientifica.htm>
21. ORACLE. Oracle Suite en las Mejores y Más Conocidas Tecnologías de Oracle. [en línea]. Disponible en: <http://www.oracle.com/global/lad/technologies/soa/soa-suite.html>
22. ORTEGA HERNÁNDEZ, Giselle y CHIRINO ESQUIJAROSA, Dariel. Propuesta de modelo para la gestión de la gobernabilidad durante la fase de diseño en un proyecto BPM/SOA. Junio 2009.
23. PEÑA, José Ramón. IBM SOA. [en línea] (2010). Disponible en: http://computerworld.net.ve/index2.php?option=com_content&do_pdf=1&id=78

24. POSAVAC, Vladimir. Arquitectura Orientada a Servicios (SOA). [en línea] (2006). Disponible en: <<http://www.emb.cl/gerencia/articulo.mv?num=120&sec=12>>
25. W3C. Scripts. [en línea] (2001). Disponible en: <<http://html.conclase.net/w3c/html401-es/interact/scripts.html>>
26. WSO2. WSO2 Web Services Application Server. [en línea]. Disponible en: <<http://wso2.com/products/web-services-application-server/>>
27. ZÁRATE MEJÍA, Eduardo. La plataforma Java. [en línea]. Disponible en: <http://profesores.fi-b.unam.mx/carlos/java/java_basico1_1.html>

ANEXOS

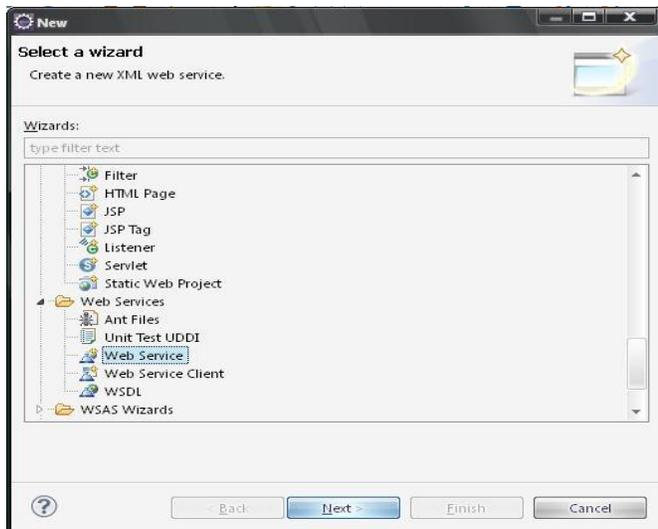
Anexo 1. Creación Servicio Web ASP.NET



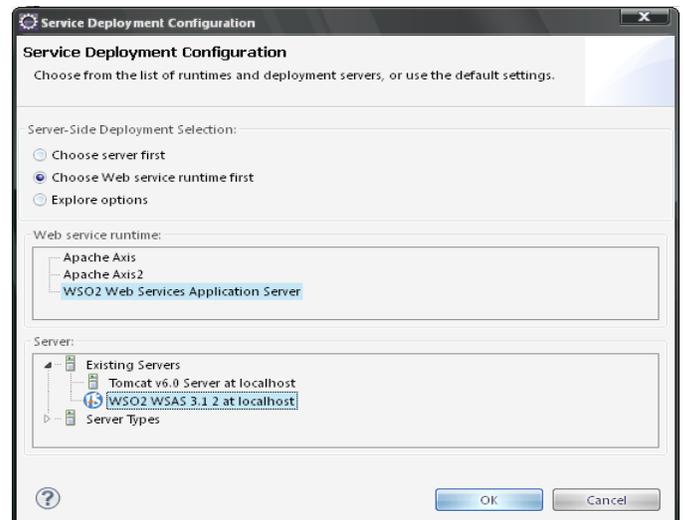
Anexo 2. Actualizar carpetas en el Explorador de soluciones de soluciones



Anexo 3. Creación del XML del servicio web



Anexo 4. Configuración del Servidor de Despliegue



Anexo 5. Generador WSDL



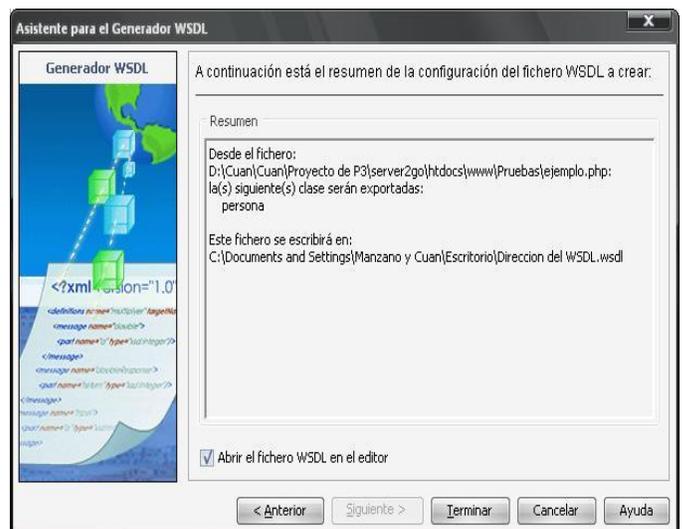
Anexo 6. Asistente para el Generador WSDL

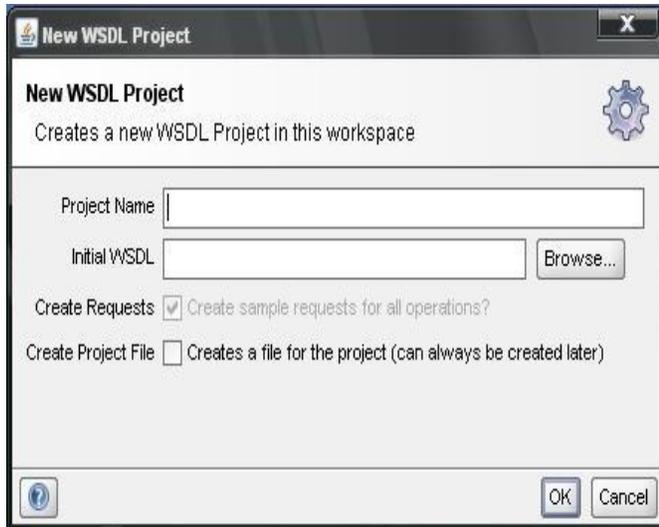
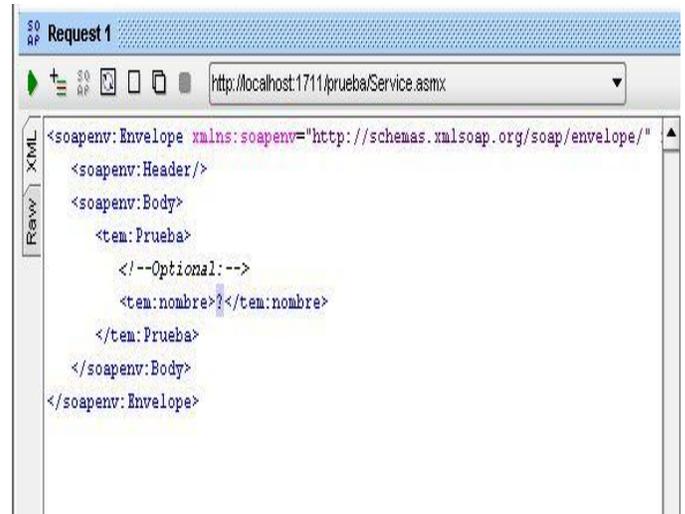


Anexo 7. Asistente para el Generador WSDL



Anexo 8. Asistente para el Generador WSDL



Anexo 9. Creación de un nuevo proyecto de prueba**Anexo 10. Entrada de parámetros al XML**

GLOSARIO DE TÉRMINOS

1. **SOAP** (del inglés *Simple Object Access Protocol*) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.
2. **WSDL** (del inglés *Web Services Description Language*) describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligán después al protocolo concreto de red y al formato del mensaje.
3. El protocolo de transferencia de hipertexto **HTTP** (del inglés *HyperText Transfer Protocol*) es el protocolo usado en cada transacción de la Web. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.
4. **Middleware** es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red). El middleware nos abstrae de la complejidad y heterogeneidad de las redes de comunicaciones subyacentes, así como de los sistemas operativos y lenguajes de programación, proporcionando una API para la fácil programación y manejo de aplicaciones distribuidas.
5. **URI** (Uniform Resource Identifier) es una cadena corta de caracteres que identifica inequívocamente un recurso (servicio, página, documento, dirección de correo electrónico, enciclopedia, etc.). Normalmente estos recursos son accesibles en una red o sistema.
6. El **bytecode** es un código intermedio más abstracto que el código máquina. Habitualmente es tratado como un fichero binario que contiene un programa ejecutable similar a un módulo objeto, que es un fichero binario producido por el compilador cuyo contenido es el código objeto o código máquina.
7. **Metalinguaje:** es un lenguaje que se usa para hablar acerca de otro lenguaje

8. **COM** (del inglés **Component Object Model**) es una plataforma de Microsoft para componentes de software introducida por dicha empresa en 1993. Esta plataforma es utilizada para permitir la comunicación entre procesos y la creación dinámica de objetos, en cualquier lenguaje de programación que soporte dicha tecnología.
9. **WSO2** es una empresa de desarrollo de aplicaciones informáticas de software libre la cual se centra fundamentalmente en el abastecimiento de soluciones para desarrolladores de Arquitecturas Orientadas a Servicios.
10. **Sistema Legado**: Aplicación de software que posee un importante valor de negocio debilitada por el constante cambio tecnológico.
11. **Procedimiento**: Básicamente, un procedimiento consiste de una serie de pasos bien definidos que permitirán y facilitarán la realización de un trabajo de la manera más correcta y exitosa posible.