



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
CDAE, CENTRO DE CONSULTORIA Y DESARROLLO DE ARQUITECTURAS EMPRESARIALES, FACULTAD 5

CAPA DE SERVICIOS PARA ALMACÉN DE DATOS DE INFOTUR

**Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas**

Autor: Yerson González Álvarez

Tutores: DrC. Orestes Febles Díaz

Ing. Lianet Cabrera Rodríguez

La Habana, 2015

*Quiero dedicar la presente investigación a:
Mis padres quienes han sido siempre un ejemplo y un apoyo.
Mi novia Jany por su amor y cariño durante todos estos años.
La Revolución por darme educación y salud gratuita.
A todos mis compañeros de grupo desde la primaria hasta la universidad y
en especial a los que fueron como una familia en estos 5 años.*

Agradecimientos

Le agradezco a ...

Mi familia por su amor, confianza y paciencia, y en especial a mis padres que siempre han querido lo mejor para mí y que son un ejemplo para mí

Mi novia por su amor y su confianza en todo momento.

Mis amigos que a lo largo de estos cinco años han sido los hermanos que no tengo.

Mis tutores por tener la paciencia necesaria y por brindarme un poco de sus propias experiencias.

En fin a todas las personas que han contribuido en mi formación durante toda mi vida.

Declaración de autoría

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales sobre esta, con carácter exclusivo.

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

Yerson González Álvarez

Autor

DrC. Orestes Febles Díaz

Tutor

Ing. Lianet Cabrera Rodríguez

Tutora

Cuba trabaja en el mejoramiento de la infraestructura tecnológica y en el proceso de informatización de los distintos sectores. El sector del turismo tiene como objetivo lograr la comunicación entre un conjunto de aplicaciones que le posibiliten un aumento en la calidad de sus servicios y una mejor gestión de la información. Esta investigación tiene como objetivo lograr exponer de forma segura e interoperable información del almacén de datos de la empresa Información Turística, INFOTUR. Para lo cual se propone la utilización de una capa de servicios web. Para la implementación y despliegue de los servicios se utiliza *WSO2 Developer Studio* y *WSO2 Application Server*, siendo en este último donde se implementan un grupo de políticas de seguridad. También la capa de servicios es probada para verificar que cumple con los estándares y requerimientos de rendimiento, diseño y funcionamiento requeridos para una solución de su tipo. Como conclusión se puede decir que se obtiene una capa de servicios que expone de forma segura la información contenida en el almacén de datos para que sea consumida por los diferentes portales de INFOTUR.

Palabras clave: Almacén de datos, capa de servicios web, interoperabilidad, seguridad, acceso a datos.

Introducción	1
1 Fundamentación Teórica	4
1.1 Conceptos Fundamentales	4
1.2 Arquitectura de Software	5
1.3 Acceso a Datos	7
1.4 Ciclo de vida de los servicios	10
1.5 Taxonomía de los Servicios (JOSUTTIS, 2007)	11
1.5.1 Sevicios Básicos	12
1.6 Servicios Web	14
1.7 Tecnologías y Herramientas	16
1.7.1 WSO2	16
1.7.2 WSO2 <i>Application Server</i>	17
1.7.3 PostgreSQL Server	17
1.7.4 SoapUI	17
1.7.5 <i>Java Database Connectivity (JDBC)</i>	18
1.7.6 <i>WSO2 Developer Studio (WSO2, 2014)</i>	18
1.8 Comparación entre diferentes soluciones	18
1.9 Metodología de desarrollo	19
1.9.1 CBDI - Servicio de Arquitectura e Ingeniería	19
1.9.2 Propuesta de Metodología SOA para el CDAE	21
1.10 Conclusiones parciales	22
2 Propuesta de solución	23
2.1 Situación de la organización	23
2.2 Identificación de servicios	24
2.3 Diseño	32
2.4 Implementación de Servicios	32
2.4.1 Implementación de la seguridad	37
2.5 Arquitectura de Despliegue	40

3 Resultados	42
3.1 Pruebas de Diseño	42
3.2 Pruebas Funcionales	43
3.3 Pruebas de Rendimiento	43
Conclusiones	46
Recomendaciones	47
Acrónimos	48
Referencias bibliográficas	50
Apéndices	53
A Web Services	54
B Implementación	58
C Pruebas	61

Índice de figuras

1.1	Vista Lógica	8
1.2	Las capas de servicio fundamentales y las etapas de expansión Arquitectura Orientada a Servicios (por sus siglas en inglés, <i>Services Oriented Architecture</i>) (SOA)	12
1.3	Suite de herramientas de WSO2	16
2.1	Situación Actual	23
2.2	Diseño del Almacén de Datos	24
2.3	Dashboard del WSO2 Developer Studio	33
2.4	Desplegando el Proyecto de Aplicaciones Compuestas	36
2.5	Configurando Response Caching	37
2.6	Escenario <i>UsernameToken</i>	38
2.7	Escenario <i>Non-repudiation</i>	39
2.8	Añadiendo la opción <i>Access Throttling</i>	39
2.9	Propuesta de diagrama de despliegue simple	40
2.10	Propuesta de diagrama de despliegue	41
3.1	Comparación de los tiempos promedios obtenidos con el requerido	44
3.2	Comparación de los tiempos máximos obtenidos con el requerido	45
A.1	<i>WS-Addressing</i>	55
A.2	<i>WS-MetadataExchange</i>	55
A.3	<i>WS-ReliableMessaging</i>	56
A.4	<i>WS-Policy</i>	56
A.5	<i>WS-Security</i>	57
B.1	Creando un Proyecto de Servicio de Datos	58
B.2	Adicionando los datos necesarios	59
B.3	Adicionando los elementos de la consulta	59
B.4	Exportando	60

Índice de tablas

1.1	Comparación entre soluciones	19
2.1	Especificación de requisitos funcionales	25
2.1	continuación de la página anterior	26
2.1	continuación de la página anterior	27
2.1	continuación de la página anterior	28
2.2	Especificación de los servicios	29
2.2	continuación de la página anterior	30
2.2	continuación de la página anterior	31
3.1	Pruebas de Diseño	42
3.1	continuación de la página anterior	43
3.2	Servicio Comida	43
3.3	Servicio Centro_Nocturno	43
C.1	Servicio Canje_Monedas	61
C.2	Servicio Comunicacion	61
C.2	continuación de la página anterior	62
C.3	Servicio Cultura	62
C.4	Servicio Evento	62
C.5	Servicio Excursion	63
C.6	Servicio Hospedaje	63
C.7	Servicio Informacion	63
C.8	Servicio Museo_Monumento	63
C.9	Servicio Naturaleza	64
C.10	Servicio Nautica	64
C.11	Servicio Patrimonio	64
C.12	Servicio Playa	64
C.13	Servicio Religion	65
C.14	Servicio Salud	65
C.15	Servicio Tienda	65

C.16 Servicio Transportacion 65
C.16 continuación de la página anterior 66
C.16 continuación de la página anterior 67

Lista de códigos fuentes

2.1	Implementación de un servicio	33
2.2	Estructura del Lenguaje de Recursos y Servicios de Datos (por sus siglas en inglés, <i>Data Services and Resource Language</i>) (DSRL)	35

La evolución de las [Tecnologías de la Información y las Comunicaciones \(TIC\)](#) en los últimos años han provocado un gran impacto en todos los ámbitos. Internet, desde su surgimiento, ha propiciado el acceso a información desde lugares muy distantes geográficamente. Además brinda oportunidades sin precedentes para alcanzar niveles más elevados de desarrollo y pone a disposición de todos sus usuarios una serie de funcionalidades básicas que abren infinitas posibilidades de superación personal, profesional, y de entretenimiento. Constituye un canal de comunicación a escala mundial, cómodo y versátil, que permite compartir y debatir ideas, que facilita el trabajo cooperativo y la difusión de las creaciones personales (CASTELLS, 2004).

Los portales Web han evolucionado con el desarrollo de Internet, pues han pasado de ser desarrollados de forma estática, enfocados principalmente en mostrar información permanente y contruidos básicamente solo por hipervínculos o enlaces(*links*), a ser desarrollados dinámicamente, propiciando servicios informativos e interactivos de forma eficiente como pueden ser foros, noticias, encuestas, buscadores, transacciones electrónicas o de manera general información en distintos formatos. Hoy día los portales Web pueden llegar a ser muy complejos, gestionando gran cantidad de información y brindando variadas funcionalidades al usuario final (POLANCO, 2013).

Las organizaciones turísticas suelen aprovechar de manera especial las ventajas que ofrecen los portales en Internet para presentar su oferta al público. El poder evocador de un portal y su estructura multidisciplinar permiten la coexistencia en un mismo portal turístico de información de bases de datos centralizadas y gestionadas por los organismos encargados de la promoción de los destinos, así como la información propia de cada empresa proveedora de servicios turísticos o de las oficinas de turismo locales. Además permiten que el usuario pueda buscar, por diferentes medios y a diferentes niveles, la información. En el turismo los flujos de información son intensos y constantes, y el éxito en la producción y en las ventas depende en buena parte de la calidad de los procesos de información en los cuales los portales juegan un papel vital. Dichos procesos no sólo se refieren a la elaboración, sino también a la distribución de la información de la oferta turística de un destino. Un proceso informativo eficaz y rápido como proveen los portales, es esencial para cualquier organización turística no sólo a la hora de ofrecer los bienes intangibles(servicios) sino también en otros momentos de la relación con el turista como:

- Después de la reservación.
- A la hora de preparar el viaje desde el lugar de origen.

- Durante y después de las vacaciones.
- Cuando se quiere que el turista vuelva a preferir a la organización.

En todos estos momentos los portales de la organización deben ser capaces de satisfacer la necesidad de información del usuario, que exigirá diferentes formatos y presentaciones (MIRALBELL, 2010).

Cuba no está aislada del acontecer tecnológico que sucede en el mundo por lo que, a pesar de la situación económica que atraviesa, nuestro país trabaja en el mejoramiento de la infraestructura tecnológica y en el proceso de informatización de los distintos sectores. El sector del turismo ha dirigido sus esfuerzos a lograr la comunicación entre un conjunto de aplicaciones y la ubicuidad de su información de tal manera que le posibiliten un aumento en la calidad de sus servicios y una mejor gestión de la información. Además es de vital importancia para este sector que la información publicada sea lo más actual posible.

La empresa **Información Turística (INFOTUR)**, perteneciente al **Ministerio del Turismo en Cuba (MINTUR)**, cuenta con una red de centros de información que tienen como objetivo brindarle toda la información de interés que necesita el turista, tanto en su preparación de viaje como durante su estancia en la Isla. Actualmente esta empresa cuenta con un grupo de 14 portales, encargados de la divulgación de gran parte de la información turística del país. Cada portal está asociado a una zona turística relevante y contiene información referente a destinos, sitios de interés y servicios ofertados, tanto por el sector estatal como por el privado, en sus alrededores. Además los portales cuentan con una base de datos propia, pero existe un grupo de datos que están siendo duplicados en las distintas instancias de estos portales. Para mitigar este problema se diseñó e implementó un almacén de datos que permite a la organización mantener una ubicuidad de su información primaria. Este almacén de datos no se encuentra en explotación ya que la empresa no cuenta con un software capaz de exponer de manera segura e interoperable la información que necesitan los portales. Para la creación de este software **INFOTUR** realizó una solicitud a la **Universidad de las Ciencias Informáticas (UCI)**. El desarrollo de este software fue asignado al **Centro de Consultoría y Desarrollo de Arquitecturas Empresariales (CDAE)** ya que es el centro que estudia y basa sus soluciones en la interoperabilidad.

A partir de lo expuesto anteriormente se identifica como **problema de la investigación** el siguiente: ¿Cómo exponer de manera segura e interoperable la información que necesitan los portales del almacén de datos de **INFOTUR**?

A partir del problema planteado se escoge como **objeto de estudio**: el acceso seguro e interoperable a datos. Enmarcado en el **campo de acción**: el acceso seguro e interoperable a datos usando servicios web.

La **idea a defender** es: con el desarrollo de una capa de servicios web se logra exponer de manera segura e interoperable información del almacén de datos de **INFOTUR**.

El **objetivo general** es: desarrollar una capa de servicios web que exponga de manera segura e interoperable información del almacén de datos de **INFOTUR**; y como **objetivos específicos** los siguientes:

1. Elaborar el marco teórico de la investigación sobre el acceso a datos seguro e interoperable.
2. Diagnosticar el escenario actual en **INFOTUR** y los beneficios de la utilización de servicios web para exponer la información del almacén de datos.

3. Diseñar e implementar una capa de servicios que permita exponer la información del almacén de datos con la calidad y seguridad requeridas.
4. Realizar pruebas de rendimiento, diseño y funcionamiento a la capa de servicios.

Métodos Teóricos

Entre los métodos teóricos utilizados se encuentran: Histórico-Lógico: Se utilizó con el objetivo de comprender los antecedentes y las tendencias tomadas en cuanto al tema de la interoperabilidad y seguridad informática. También se usa para la búsqueda de conceptos y términos propios de la [SOA](#).

Analítico-Sintético: Se utilizó para seleccionar la bibliografía y para resumir los criterios de los diferentes autores que tratan el tema de la interoperabilidad y la seguridad informática en sus escritos.

Métodos Empíricos

Experimental: Este método se utilizó para comprobar y fundamentar de qué manera una solución [SOA](#) resuelve el problema de seguridad e interoperabilidad del almacén de datos de [INFOTUR](#).

El documento está estructurado en 3 capítulos. En el Capítulo 1 se presentan los elementos teóricos que sirven de base a la investigación del problema planteado. Además, se hace un estudio de las diferentes soluciones de interoperabilidad haciendo énfasis en la relacionadas con [SOA](#). En el Capítulo 2 se propone una solución basada en una [SOA](#) y se exponen elementos de la identificación, implementación y despliegue de los servicios. Por último, en el Capítulo 3 se muestran y discuten los resultados obtenidos. Además, se exponen los resultados de las pruebas funcionales y de rendimiento realizadas a la capa de servicios.

1.1. Conceptos Fundamentales

Interoperabilidad

El término interoperabilidad tiene muchas connotaciones, incluyendo los objetivos de la comunicación, el intercambio de información, la cooperación y el uso compartido de recursos entre distintos tipos de sistemas. De hecho, la esencia de la interoperabilidad es garantizar las relaciones entre los sistemas, donde cada relación es una forma de compartir, comunicar, intercambiar y cooperar (GRADMANN, 2009). Existen múltiples definiciones de la interoperabilidad defendidas por varios autores reconocidos a nivel internacional; entre ellas se encuentran las siguientes:

- Es la capacidad de compartir datos y posibilitar el intercambio de información de los sistemas y los procedimientos a los que dan soporte (GONZÁLEZ MOREIRO; CUADRADO SÁNCHEZ y LARA MORATO, 2012).
- Se define como la capacidad de los sistemas de información, de las tecnologías de la comunicación y de los procesos de negocio para intercambiar datos, compartir la información y el conocimiento (GRADMANN, 2009).
- La capacidad de diferentes productos y servicios de las TIC para intercambiar y usar datos e información con el objetivo de funcionar juntos en un entorno conectado en red. En su acepción más simple, la interoperabilidad trata de asegurar que los sistemas trabajen juntos (CHAMORRO BORGES, 2007).

El autor de esta investigación coincide con la propuesta de definición de interoperabilidad dada por *González, Cuadrado y Lara* ya que integra los elementos esenciales del término.

Seguridad Informática

El término seguridad informática tiene innumerables definiciones, algunas de estas son:

- Es el conjunto de procedimientos, estrategias y herramientas que permitan garantizar la integridad, la disponibilidad y la confidencialidad de la información (LUGO JEREZ, 2004).
- Consiste en garantizar que el material y los recursos de software de una organización se usen únicamente para los propósitos para los que fueron creados y dentro del marco previsto (KIOSKEA, 2015).

La seguridad informática tiene cinco objetivos principales:

- **Integridad:** Los componentes del sistema permanecen inalterados a menos que sean modificados por los usuarios autorizados.
- **Confidencialidad:** Los componentes del sistema son accesibles sólo por los usuarios autorizados.
- **Disponibilidad:** Los usuarios deben tener disponibles todos los componentes del sistema cuando así lo deseen.
- **Evitar el rechazo:** Evita que cualquier entidad que envió o recibió información alegue, que no lo hizo.
- **Autenticación:** Asegura que sólo los individuos autorizados tengan acceso a los componentes del sistema.

Almacén de Datos

La definición del término almacén de datos para la presente investigación responde a la definición de un **Almacén de Datos Operacionales**, (por sus siglas en inglés, *Operational Data Store*) (ODS). El mismo se define como: un sistema que recopila la información existente en los sistemas operacionales (Bases de Datos) y la ofrece a los sistemas informacionales o directamente a los usuarios finales de negocio (CASTILLO; MORATA y ARBOL, 2005).

Servicios

Un servicio es una unidad autónoma de software que realiza una tarea específica. Tiene tres componentes: interfaz, contrato e implementación. La interfaz define como un consumidor de servicios realizará peticiones a un proveedor, el contrato define la forma en que el proveedor y el consumidor de servicios deben interactuar, y la implementación es el código real del servicio (THAI, 2011).

1.2. Arquitectura de Software

La arquitectura de software tuvo sus inicios hacia 1968 por Edsger Dijkstra, de la Universidad Tecnológica de Eindhoven en Holanda y Premio de Turing 1972, el cual propuso que se estableciera una estructuración correcta de los sistemas de software antes de lanzarse a programar, escribiendo código de cualquier manera (DIJKSTRA, 1983). Aunque Dijkstra no utiliza el término arquitectura para describir el diseño conceptual

del software, sus ideas estarían presentes en la conferencia de la NATO de 1969 en los comentarios de P. I. Sharp: *“Pienso que tenemos algo, aparte de la ingeniería de software, algo de lo que hemos hablado muy poco pero que deberíamos poner sobre el tapete y concentrar la atención en ello. Es la cuestión de la arquitectura de software. La arquitectura es diferente de la ingeniería de software... con el tiempo probablemente llegue a hablarse de la arquitectura de software de Dijkstra”* (NAUR; RANDELL y BUXTON, 1976).

En 1975, Brooks, diseñador del sistema operativo OS/360 y Premio de Turing 2000, utiliza el concepto de arquitectura del sistema para designar *“la especificación completa y detallada de la interfaz de usuario”* (BROOKS JR, 1975). En el mismo texto él distinguía entre arquitectura e implementación; mientras aquella decía *qué hacer*, la implementación se ocupa de *cómo*.

Clements y Northrop escriben que en todo el desenvolvimiento ulterior de la disciplina permanecería en primer plano la idea de que: la estructura es primordial y su elección correcta ha de ser crítica para el éxito del desarrollo de una solución. Ni duda cabe que la elección de la estructura correcta sintetiza, como ninguna otra expresión, el programa y la razón de ser de la arquitectura de software (CLEMENTS y NORTHROP, 1996).

En la década de 1980, los métodos de desarrollo estructurado demostraron no escalar suficientemente y fueron dejando el lugar a un nuevo paradigma, el de la programación orientada a objetos. En teoría, parecía posible modelar el dominio del problema y el de la solución en un lenguaje de implementación. Paralelamente, hacia fines de los 80' y comienzo de los 90' la expresión *arquitectura de software* comienza a aparecer en la literatura para hacer referencia a la configuración morfológica de una aplicación (REYNOSO, 2004).

El primer estudio en que aparece el término *“arquitectura de software”* en el sentido en que hoy se conoce es sin dudas el de Perry y Wolf en 1992, que comienza diciendo: *El propósito de este papel es construir el fundamento para la arquitectura de software. Primero desarrollaremos una intuición para la arquitectura de software recurriendo a diversas disciplinas arquitectónicas bien definidas. Sobre la base de esa intuición, presentamos un modelo para la arquitectura de software que consiste en tres componentes: elementos, forma y razón(rationale). Los elementos son elementos ya sea de procesamiento, datos o conexión. La forma se define en términos de las propiedades de, y las relaciones entre, elementos, es decir, restricciones operadas sobre ellos. La razón proporciona una base subyacente para la arquitectura en términos de las restricciones del sistema, que lo más frecuente es que se deriven de los requerimientos del sistema... La década de 1990, creemos, será la década de la arquitectura de software. Usamos el término “arquitectura” en contraste con “diseño”, para evocar nociones de codificación, de abstracción, de estándares, de entrenamiento formal(de los arquitectos de software) y de estilo... Es tiempo de re-examinar el papel de la arquitectura de software en el contexto más amplio del proceso de software y de su administración, así como señalar las nuevas técnicas que han sido adoptadas* (PERRY y WOLF, 1992).

Cumpliendo con lo que avizoraran Perry y Wolf, la década de 1990 fue sin duda la de un período de consolidación y diseminación de la arquitectura de software. En ella surge la programación basada en componentes, que en su momento de mayor impacto impulsó a algunos arquitectos mayores, como Paul Clements, a afirmar que la arquitectura de software promovía un modelo que debería ser más de integración

de componentes pre-programados que de programación. Clements definió la arquitectura de software como: una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones (CLEMENTS, 1996).

Arquitectura Orientada a Servicios (SOA)

SOA es un estilo arquitectónico que propone la creación de servicios de negocios con bajo acoplamiento, que son interoperables; aumentando así la flexibilidad del negocio. Una solución SOA consiste en la orquestación de un proceso de negocio de principio a fin mediante la invocación de diversos servicios. Cada uno de ellos provee de una descripción basada en una interfaz estándar que da soporte a la flexibilidad y a la reconfiguración dinámica del proceso de negocio (REYNOLDS, 2005).

Según OASIS¹, SOA es un paradigma para organizar y utilizar capacidades distribuidas que pueden estar bajo el control de varios propietarios (dominios). Provee medios uniformes para ofrecer, descubrir, interactuar y utilizar capacidades para producir los efectos deseados que son consistentes con precondiciones y expectativas medibles (OASIS, 2006).

También se puede definir como *“una arquitectura que básicamente está orientada a un conjunto de servicios tanto de negocio como tecnológicos que interactuando entre si nos proporcionan la lógica necesaria para construir aplicaciones de una manera rápida y cumpliendo con los principios de la Orientación a Servicios. Además proporciona una serie de guías y recomendaciones para conseguir los objetivos que se impone una organización a la hora de desarrollar aplicaciones. Surge de la necesidad de mejorar la integración e interoperabilidad entre aplicaciones; cuyo fin o propósito es la convergencia entre el desarrollo e integración que minimice las imitaciones entre aplicaciones”* (JIMENEZ, 2010).

La idea principal de SOA es la de construir aplicaciones que soporten los procesos de negocios mediante la combinación de elementos de software poco acoplados entre si, llamados servicios (GRAHAM, 2008).

1.3. Acceso a Datos

A nivel arquitectural la mayoría de las aplicaciones Web se componen de tres capas: la capa de presentación, la de lógica de negocio y la capa de persistencia; aunque estas varían según la complejidad de la aplicación. En la Figura 1.1 se muestra la descomposición de las capas de una aplicación web y cuál es la relación que existe entre estas. El acceso a datos como se puede apreciar, en dicha figura, forma parte de una de las capas: la capa de persistencia (TAHUITON MORA, 2011).

¹Consortio internacional sin ánimo de lucro que gestiona el desarrollo, convergencia y adopción de estándares sobre comercio electrónico

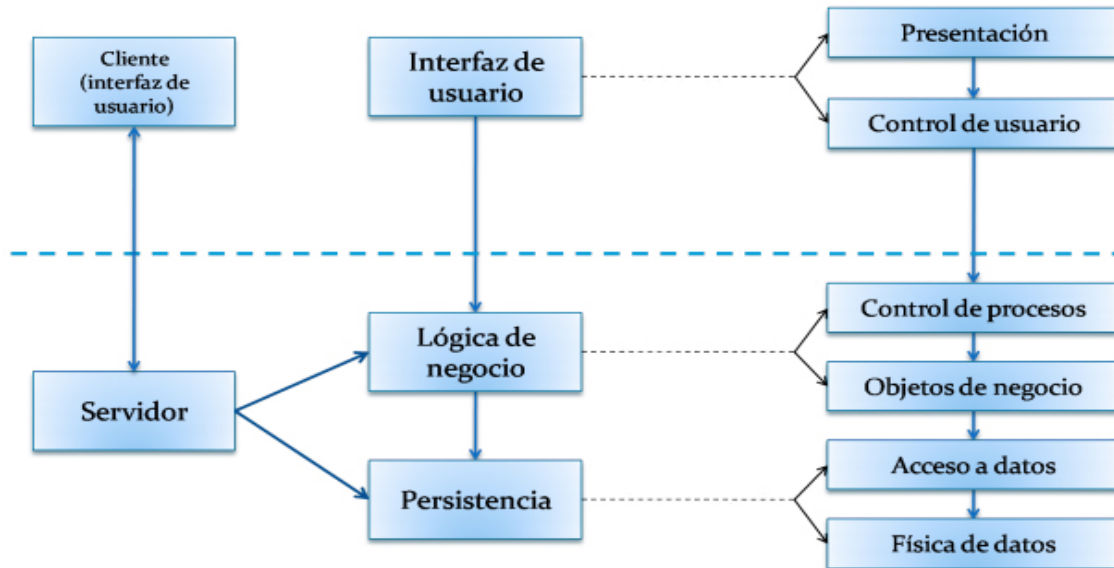


Figura 1.1. Vista Lógica

Existen varias herramientas que permiten exponer datos de manera segura e interoperable tanto de una base de datos como de un almacén, entre ellas se encuentran las siguientes:

Open Database Connectivity ODBC (CASAS GUIJARRO, 1997)

Open Database Connectivity (ODBC) es una *Interfaz de Programación de Aplicaciones* (por sus siglas en inglés, *Applications Programming Interface*) (API) estándar que permite acceder a datos contenidos y manejados por *Sistema de Gestión de Bases de Datos (SGBD)*. Haciendo uso de ella las aplicaciones pueden acceder a datos almacenados en una gran variedad de ordenadores personales, miniordenadores y grandes ordenadores, incluso aunque cada *SGBD* utilice un formato diferente para guardar la información.

Entre sus características, destacan:

- Utiliza *Structured Query Language (SQL)*.
- Oculta al programador la complejidad a la hora de conectarse a un origen de datos: por ejemplo, el acceso a los datos a través de redes de comunicación es transparente.
- Permite a múltiples aplicaciones acceder a múltiples orígenes de datos.
- Proporciona un modelo de programación homogéneo, es decir, bases de datos muy diferentes se manejan como si fueran idénticas, siendo *ODBC* el encargado de realizar las adaptaciones necesarias.
- Se basa en el modelo cliente/servidor

Algunas desventajas son (MEJÍAS, 2001):

- Se limita a datos relacionales basados en la sintaxis de [SQL](#).
- Las consultas tienen que pasar varias capas lo que hace que el tiempo de respuesta se incremente.

ADO.NET (APONCIO BORGES, 2009)

Es un conjunto de clases que exponen servicios de acceso a datos para el programador de .NET. *ActiveX Data Objects .NET Framework (ADO.NET)* ofrece abundancia de componentes para la creación de aplicaciones de uso compartido de datos distribuido. Constituye una parte integral de *.NET Framework* y proporciona acceso a datos relacionales, XML y de aplicaciones. Satisface diversas necesidades de desarrollo, como la creación de clientes de base de datos de aplicaciones para usuarios y objetos empresariales de nivel medio que utilizan aplicaciones, herramientas, lenguajes o exploradores de Internet. Además proporciona acceso coherente a orígenes de datos como *Microsoft SQL Server* y XML, así como a orígenes de datos expuestos mediante *Object Linking and Embedding for Databases (OLE DB)* y *ODBC*. Las aplicaciones para usuarios que comparten datos pueden utilizar [ADO.NET](#) para conectarse a estos orígenes de datos y recuperar, manipular y actualizar los datos contenidos.

[ADO.NET](#) separa el acceso a datos de la manipulación de datos y crea componentes discretos que se pueden utilizar por separado o conjuntamente. También incluye proveedores de datos de *.NET Framework* para conectarse a una base de datos, ejecutar comandos y recuperar resultados. Los resultados se procesan directamente o se colocan en un objeto *DataSet* de [ADO.NET](#) con el fin de exponerlos al usuario para un propósito específico, combinados con datos de varios orígenes, o de utilizarlos de forma remota entre niveles. Existen dos componentes de [ADO.NET](#) que se pueden utilizar para obtener acceso a datos y manipularlos:

- Proveedores de datos de *.NET Framework* son componentes diseñados explícitamente para la manipulación de datos y el acceso rápido a datos de solo lectura y avance.
- El *DataSet* está expresamente diseñado para el acceso a datos independientemente del origen de datos. Como resultado, se puede utilizar con múltiples y distintos orígenes de datos, con datos XML o para administrar datos locales de la aplicación.

OLE DB

[OLE DB](#) es una [API](#) de acceso a datos de *Microsoft* diseñada para permitir el acceso a todo tipo de fuentes de datos, independientemente del formato o ubicación. [OLE DB](#) se basa en el *Component Object Model (COM)* de *Microsoft*, que es la base de *OLE* y *ActiveX*. Además es el núcleo de la estrategia de Acceso Universal de *Microsoft*. *OLE DB* tiene como objetivo proporcionar un entorno para que aplicaciones de negocios puedan acceder a todo tipo de fuentes de datos de una manera integrada, incluyendo los datos de escritorio como hojas de cálculo, documentos de procesamiento de texto y correo electrónico; datos del servidor almacenados en el sistema de archivos, archivos secuenciales indexados, relacionales, bases de datos jerárquicas y de red; y los datos calculados por los objetos de negocio de nivel medio (BLAKELEY, 1997).

1.4. Ciclo de vida de los servicios

El ciclo de vida de un servicio es bastante parecido al de un software, el mismo cuenta con las siguientes fases (JOSUTTIS, 2007) y (HEVIA VEGA, 2013):

1. Identificación: En esta etapa se identifica la necesidad de un nuevo servicio para responder a un requerimiento de negocio. Se registra el nombre del servicio y una descripción de su funcionalidad.
2. Diseño: Se define el contrato del servicio, con su interfaz de entrada/salida.
3. Implementación: Es donde se programa como tal el servicio.
4. Pruebas: En esta fase se valida que el servicio cumple con los objetivos propuestos.
5. Despliegue: En esta etapa, el servicio recibe las invocaciones de sus clientes.

Técnicas de identificación de Servicios (ERL, 2005)

Para la realización de la identificación de los servicios se puede aplicar una de las siguientes técnicas propuestas por Thomas Erl:

Top-Down

Esta estrategia se basa en realizar un análisis primario que requiere orientar los procesos a servicios y crear o reajustar el modelo de negocio de toda la organización. A partir de ello se descomponen todos los procesos de negocio y se analizan junto con el modelo de entidades general, después se identifican operaciones candidatas que se agrupan en servicios candidatos, estos son sometidos a un análisis buscando reusabilidad y autonomía, además se identifican las posibles composiciones y agrupaciones de las operaciones. El mismo tiene como entradas los procesos de negocio y modelos de entidades.

Este enfoque es eficaz en la creación de servicios altamente reusables tanto de negocio como de aplicación, produce arquitecturas de alta calidad ya que permite realizar un análisis cuidadoso de las potencialidades de reutilización y composición. Afirma las bases para una empresa estandarizada donde los servicios mantienen un estado de adaptabilidad, mientras se continúa unificando la heterogeneidad existente. Sin embargo, implica una cantidad de tiempo y un alto costo debido a que las organizaciones requieren de grandes inversiones para llevar a cabo el análisis del negocio que puede requerir gran cantidad de tiempo sin mostrar un resultado inmediato, esta situación casi nunca es admitida por los directivos de la organización provocando la cancelación de los proyectos.

Bottom-Up

Este enfoque da valor a la creación de servicios como medio para satisfacer los requisitos de las aplicaciones. Se construyen servicios según se necesiten de forma que encapsulen lógica de aplicación para servir

requisitos inmediatos de una solución. Es un enfoque muy rápido y útil para la integración de aplicaciones, por lo que permite la creación rápida de servicios que den respuestas a requisitos inmediatos de aplicación consumiendo poco esfuerzo del análisis. Sin embargo, no es un enfoque conveniente para lograr todos los beneficios que deben reportar una SOA. Además, no potencia la alineación al negocio, la reusabilidad, ni la flexibilidad. Permite la creación eficiente de los servicios requeridos por una solución.

Meet-in-the-Middle

En paralelo con el análisis Top-Down se descomponen un determinado grupo de procesos de negocio y se analizan junto con su modelo de entidades. A partir de esto se identifican operaciones candidatas que se agrupan en servicios candidatos, estos son sometidos a un análisis buscando reusabilidad y autonomía en los servicios, además se identifican las posibles composiciones y agrupaciones de las operaciones. Luego si es necesario se realiza un análisis de los requisitos de aplicación para derivar servicios que falten, después de que se termine otro grupo de procesos estos servicios son revisitados y alineados con el nuevo modelo de negocio. Se realiza a partir de los procesos, modelos de entidad y requisitos de aplicaciones si son necesarios.

Esta estrategia toma lo mejor de las dos anteriores y combina la consecución de resultados a corto plazo permitiendo que se satisfagan los requisitos inmediatos sin disminuir la integridad y calidad de la SOA y el modelo de negocio de la organización. Dichos resultados en menor tiempo constituyen una motivación directa para continuar con el desarrollo de la iniciativa SOA. Pero esta estrategia es más compleja que las dos anteriores porque necesita responder a dos grupos de requisitos opuestos. Debe resolver necesidades a corto y largo plazo por lo que requiere de un mayor esfuerzo ya que la arquitectura y los servicios deben ser revisados y rediseñados acorde a lo que se va modelando en el negocio por lo que se necesita gran atención y control para lograr que los servicios se mantengan alineados con los modelos de negocio que cambian constantemente.

1.5. Taxonomía de los Servicios (JOSUTTIS, 2007)

La Real Academia de la Lengua Española define taxonomía como la ciencia que trata de los principios, métodos y fines de la clasificación. Se aplica en particular, dentro de la biología, para la ordenación jerarquizada y sistemática, con sus nombres, de los grupos de animales y de vegetales. Los servicios dentro de una empresa pueden clasificarse según su taxonomía en tres categorías fundamentales:

- Básicos.
- Compuestos.
- De Procesos

Basado en estas tres categorías de servicio, pueden definirse tres capas diferentes de servicio y etapas de expansión (ver 1.2):

1. Fundamental SOA: tiene solamente una capa de servicios básicos.
2. Federal SOA: tiene, además de los servicios básicos, una capa de servicios compuestos.
3. Proceso-Habilitado SOA: tiene una tercera capa para los servicios de proceso.

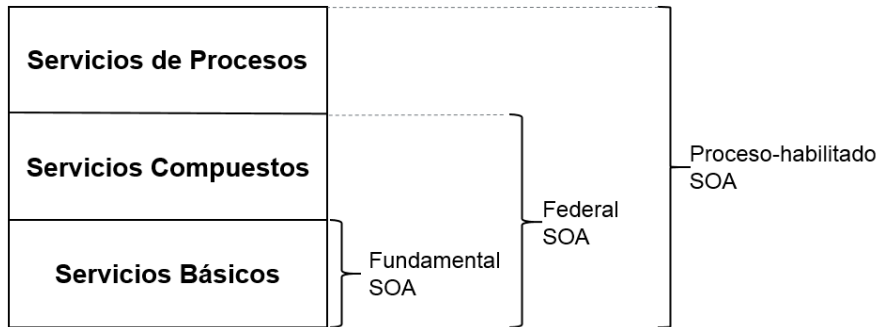


Figura 1.2. Las capas de servicio fundamentales y las etapas de expansión SOA

1.5.1. Servicios Básicos

Servicios básicos son servicios que cada uno provee una funcionalidad básica de negocio y que no se pueden dividir en múltiples servicios. Usualmente, estos servicios proveen la primera capa de negocio fundamental para un *backend*² específico o problema de dominio. Típicamente, estos servicios son de corto plazo de ejecución. El rol de estos servicios es envolver un *backend* o problema de dominio de manera que los consumidores (y los servicios del nivel superior) pueden acceder al *backend* usando la infraestructura común SOA. Existen dos tipos de servicios básicos:

- Servicios Básicos de Datos.
- Servicios Básicos de Lógica.

Servicios Básicos de Datos o Servicios de Datos

Los servicios básicos encapsulan aspectos específicos de la plataforma y detalles de implementación del mundo exterior, de manera que el consumidor puede pedir un servicio sin saber cómo éste está implementado.

Estos servicios deberían proveer alguna funcionalidad mínima de negocio. Esto quiere decir, que no debería haber interfaz técnica, como en un servicio que devuelve una tabla de la base de datos. En lugar de eso, los servicios básicos de datos típicamente o leen y retornan o escriben conjuntos de datos que usualmente permanecen juntos, según las necesidades de negocio. Es importante que estos servicios se encapsulen o se abstraigan de tal manera que se puedan solucionar algunos problemas como la creación de inconsistencias

²Puede ser una base de datos, una conexión remota con otra empresa, un motor de reglas de negocios, etc.

en el *backend*. Además los servicios básicos deberían tener las llamadas propiedades ACID, lo cual quiere decir que deberían ser:

- Atómico: Esto implica que la llamada del servicio, o sucede o no tiene efecto. Asegurar esta propiedad es la tarea del *backend* que provee un servicio.
- Consistente: Esto quiere decir que después de la llamada de un servicio, el *backend* es dejado en un estado legal, consistente. Así, ninguna llamada de servicio debería poder traer al *backend* a un estado inconsistente.
- Incomunicado: Un servicio que está siendo procesado por un *backend* no está influenciado por otras llamadas de servicios que corren en el mismo *backend* y al mismo tiempo. Es decir, un servicio que ya leyó, debería devolver los datos leídos consistentes en sí.
- Durable: Después de que la llamada de servicio sucede, está garantizado que su efecto es persistente. Es decir, ninguna falla de sistema puede deshacer el resultado de la llamada del servicio por accidente.

Servicios Básicos de Lógica

Los servicios básicos de lógica representan reglas fundamentales de negocio. Estos servicios usualmente procesan algunos datos de entrada y devuelven los resultados correspondientes. Comparados con servicios básicos de datos, los servicios lógicos básicos son una minoría. Ejemplos típicos de servicios básicos de lógica podrían ser servicios que:

- Definan catálogos del producto y listas de precios.
- Definan reglas para contratos cambiantes del cliente.
- Devuelvan si un año es bisiesto.

Servicios Compuestos

La segunda etapa de expansión añade los servicios compuestos. Estos representan la primera categoría de servicios que están compuestos de otros servicios (básico y / u otros servicios compuestos). Estos funcionan en un nivel más alto que los servicios básicos, pero son todavía de breve período de ejecución y conceptualmente sin estados. Para usar un término de flujo de trabajo, un servicio compuesto representa un flujo del micro, lo cual es un flujo corto de ejecución de actividades (servicios, en este caso) dentro de un proceso de negocio.

Servicios de Procesos

La tercera etapa de expansión suma los servicios de proceso, los cuales representan flujos de trabajo de largo plazo o procesos de negocio. Desde un punto de vista del negocio, un servicio de proceso representa

un flujo de macro, lo cual es un flujo largo de ejecución de actividades (servicios) que es interruptible por la intervención humana.

Un ejemplo típico de un servicio de proceso es un servicio de carrito de compras. Su estado contendría el contenido del carrito de compras, quizá combinado con algunos datos del cliente de manera que la orden del cliente podría ser mantenida y podría ser manipulado sobre sesiones múltiples. Dependiendo del diseño, el proceso puede terminar cuando la orden final del cliente se coloca, o incluso con la entrega de todos los artículos pedidos.

1.6. Servicios Web

Servicios web se refieren a una colección de estándares que cubren la interoperabilidad. De hecho, estos estándares definen los protocolos que se usan para comunicarse y el formato de las interfaces que se usan para especificar los servicios y los contratos de servicio (JOSUTTIS, 2007). Es por esto que son una de las tecnologías más usadas para alcanzar la interoperabilidad entre aplicaciones informáticas. Pueden ser usados en diferentes plataformas, sistemas operativos y mediante varios lenguajes de programación (JURIC, 2010).

Esta tecnología está basada fundamentalmente en especificaciones y estándares como:

- **Protocolo de Acceso Simple a Objetos** (por sus siglas en inglés, *Simple Object Access Protocol*) (**SOAP**): provee un formato de mensaje y mecanismo de mensaje que soporta el intercambio estandarizado de XML entre aplicaciones informáticas. Es utilizado para expresar la estructura de datos basada en XML muy relacionado a la creación de los contratos de servicios (SATTARI y KUNJUMOHAMED, 2012).
- **Lenguaje de Descripción de Servicios Web** (por sus siglas en inglés, *Web Services Description Language*) (**WSDL**): es el estándar definido por el *World Wide Web Consortium* (**W3C**), es una notación XML para describir un servicio web. Un **WSDL** incluye la definición del contrato y describe la interfaz de un servicio web (CENTER, 2014).
- **XML Schema Definition Language** (**XSD**): esquema que provee un lenguaje formal para la definición de la estructura y validación de documentos XML. Sirve para describir la estructura detallada de elementos que forman parte del contrato del servicio (TONG, 2010).
- **Universal Description Discovery and Integration** (**UDDI**): gestiona la información de proveedores, las implementaciones y metadatos de los servicios tanto web como electrónicos y no electrónicos. Puede ser utilizado para anunciar, descubrir y consumir los servicios (GEYER, 2006).

WS-* (ERL, 2005)

- **WS-Addressing:** Provee una colección de elementos de encabezado de mensajes que pueden suplementar un mensaje con meta información diversa (principalmente relacionada con el ruteo).
- **WS-ReliableMessaging:** Introduce el concepto de *acknowledgements*(acuses de recibo), que proporcionan un mecanismo inherente para una notificación garantizada de entregas exitosas y fallidas.
- **WS-MetadataExchange:** Proporciona una forma para consultar y descubrir metadatos asociados a un servicio Web, incluyendo la posibilidad de buscar un archivo [WSDL](#) y las definiciones de *WS-Policy* asociadas.
- **WS-Policy:** Proporciona un medio para adjuntar propiedades a recursos Web(tales como: reglas, comportamientos, necesidades y preferencias). Las Políticas añaden una capa importante de metadatos para [SOA](#) que aumenta la interoperabilidad y el potencial de descubrimiento de servicios, mientras que también eleva la calidad general de la mensajería dentro de [SOA](#).
- **WS-Security:** Proporciona extensiones que se pueden utilizar para implementar las medidas de seguridad a nivel de mensaje. Estas extensiones protegen el contenido del mensaje durante el transporte y durante el procesamiento de los intermediarios de servicios; y además protegen al proveedor de servicios de solicitantes maliciosos.

Ver en Anexos [A.1](#) la relación entre cada uno de los WS-* con los demás.

Beneficios (VEGA LEBRÚN, 2005):

- **Promueven la interoperabilidad:** La interacción entre un proveedor y un solicitante de servicio está diseñada para que sea completamente independiente de la plataforma y el lenguaje. Esta interacción requiere un documento WSDL para definir la interfaz y describir el servicio, junto con un protocolo de red (generalmente HTTP).
- **Permiten la integración “justo-a-tiempo”(JIT):** El proceso de descubrimiento se ejecuta dinámicamente, a medida que los solicitantes de servicio utilizan a los agentes para encontrar proveedores de servicio. Una vez el solicitante y el proveedor de servicio se han ubicado, se utiliza el documento [WSDL](#) del proveedor para enlazar al solicitante con el servicio. Esto significa que los solicitantes, los proveedores y los agentes actúan en conjunto para crear sistemas que son auto-configurables y adaptativos.
- **Reducen la complejidad por medio del encapsulamiento:** Los solicitantes y los proveedores del servicio se preocupan por las interfaces necesarias para interactuar. Como resultado, un solicitante de servicio no sabe cómo fue implementado el servicio por parte del proveedor, y éste a su vez, no sabe cómo utiliza el cliente el servicio. Estos detalles se encapsulan en los solicitantes y proveedores. El encapsulamiento es crucial para reducir la complejidad.

- **Dan una “nueva vida” a las aplicaciones de legado:** Es relativamente correcto tomar una aplicación, generar una envoltura(*wrapper*) SOAP, luego generar un documento WSDL para moldear la aplicación como un servicio web.
- **Abren la puerta a nuevas oportunidades de negocio:** Los servicios web facilitan la interacción con socios de negocios, al poder compartir servicios internos con un alto grado de integración.
- **Disminuyen el tiempo de desarrollo de las aplicaciones:** Pues gracias a la filosofía de orientación a objetos utilizada, el desarrollo se convierte más bien en una labor de composición.

1.7. Tecnologías y Herramientas

1.7.1. WSO2

La empresa WSO2 fue fundada en el 2005 por el Dr. Sanjiva Weerawarana. Tiene su sede en Sri Lanka y cuenta con una suite de herramientas 1.3 de código abierto y libre bajo la licencia de Apache 2 para el desarrollo de SOA.

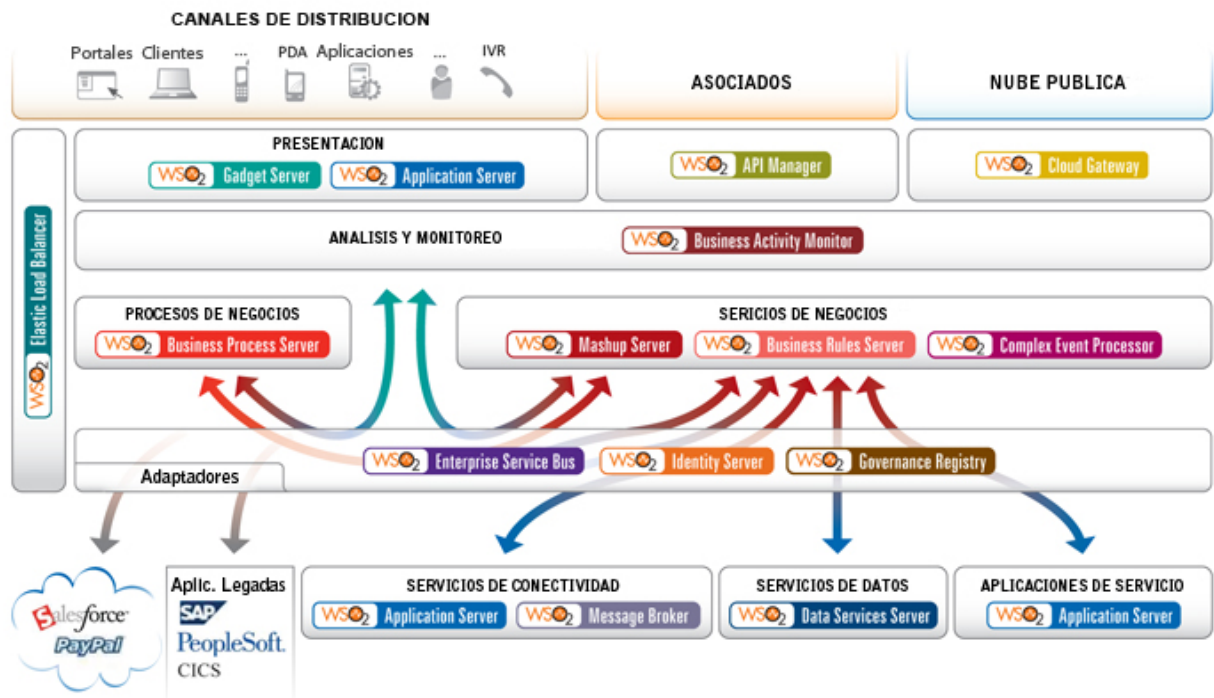


Figura 1.3. Suite de herramientas de WSO2

1.7.2. WSO2 Application Server

La empresa WSO2 ha posicionado en 3 fases diferentes el producto [Servidor de Aplicaciones \(por sus siglas en inglés, WSO2 Application Server\) \(AS\)](#). Este es el sucesor de *WSO2 Web Services Application Server* (WSAS) puesto en marcha en 2008, siendo su predecesor WSO2 Tungsteno.

El [AS](#) es un software de código abierto y desarrollado en base a la plataforma basada en componentes, WSO2 Carbon. Es capaz de albergar cualquier tipo de aplicaciones web desplegadas en el Apache Tomcat. Los usuarios pueden crear, consumir y gestionar sus aplicaciones y servicios de una manera unificada a través de la interfaz de usuario del mismo, *Management Console*. Además soporta la comunicación segura. Las aplicaciones web heredan características tales como la autenticación/autorización y la gestión de usuarios de la plataforma de WSO2 Carbon. También soporta el despliegue de los diferentes tipos de servicios, como los servicios de datos, spring, axis2 y JAX-WS. Posee un conjunto de escenarios de seguridad aplicables a los servicios desplegados, los cuales permiten confidencialidad, integridad, autenticación y/o no repudio (WSO2, 2014).

1.7.3. PostgreSQL Server

El origen de PostgreSQL se sitúa en el gestor de bases de datos POSTGRES desarrollado en la Universidad de Berkeley y que se abandonó en favor de PostgreSQL a partir de 1994. Ya para ese entonces, contaba con prestaciones que lo hacían único en el mercado.

PostgreSQL Server es un gestor de base de datos orientadas a objetos, muy conocido y usado en entornos de software libre porque cumple con los estándares SQL92 y SQL99, y también por el conjunto de herramientas avanzadas que soporta. Puede funcionar en múltiples plataformas. Cuenta con un rico conjunto de tipos de datos, permitiendo además su extensión mediante tipos y operadores definidos y programados por el usuario. Es altamente confiable en cuanto a estabilidad se refiere. Puede extenderse con librerías externas para soportar encriptación (GINESTÀ y PÉREZ MORA, 2006). Esta herramienta se utiliza porque el almacén de datos con que cuenta la empresa [INFOTUR](#) está desplegado en PostgreSQL Server.

1.7.4. SoapUI

SoapUI es una herramienta para invocar, inspeccionar, simular y testear servicios web sobre HTTP³. Es una completa herramienta de test que agiliza las labores de construcción y favorece la creación de servicios web. Es capaz de generar, automáticamente, a partir de un [WSDL](#), *tests* con todas las peticiones [SOAP](#) de las operaciones definidas en ese descriptor. Adicionalmente pueden crearse servicios web que simulen el comportamiento de un objeto verdadero. También es muy versátil en otros aspectos, pudiendo seleccionar el método HTTP, aceptando varios métodos de autenticación, ficheros adjuntos y permitiendo la incorporación de *scripting* que dinamice el comportamiento de peticiones respuestas según nuestras necesidades.

³Protocolo de Transporte de Hipertextos

1.7.5. *Java Database Connectivity (JDBC)*

Java Database Connectivity (JDBC) es un [API](#) que describe o define una librería estándar para acceso a fuentes de datos, principalmente orientado a bases de datos relacionales que usan [SQL](#). [JDBC](#) no sólo provee una interfaz para acceso a motores de bases de datos, sino que también define una arquitectura estándar, para que los fabricantes puedan crear los *drivers* que permitan a las aplicaciones *Java* el acceso a los datos. Puede definirse como un conjunto de clases, métodos e interfaces escritos en lenguaje *Java*, que permiten el acceso a sistemas de bases de datos relacionales utilizando instrucciones [SQL](#).

1.7.6. *WSO2 Developer Studio (WSO2, 2014)*

WSO2 Developer Studio es una plataforma de herramientas donde se puede desarrollar, implementar, probar y depurar aplicaciones [SOA](#). Trabaja en el popular y de código abierto [Entorno de Desarrollo Integrado](#) (por sus siglas en inglés, *Integrated Development Environment*) ([IDE](#)) Eclipse. Mediante la integración con la plataforma WSO2 Carbon permite crear, implementar y administrar una variedad de artefactos.

Developer Studio tiene soporte para todas las fases del ciclo de vida de la aplicación. Durante la fase de desarrollo, se crean diferentes tipos de artefactos de acuerdo a los requerimientos, incluyendo aplicaciones web, servicios de datos y proxy. Debido a su integración con Eclipse permite explotar todas las capacidades de un [IDE](#) Java como ayuda de contenido, ricos editores y vistas, y así sucesivamente. También le permite definir un proyecto que representa una aplicación compuesta completa que abarca múltiples productos y características. Simplifica enormemente la creación de artefactos con los editores y la gestión de las relaciones y dependencias entre estos servicios gráficos.

Después de desarrollar la aplicación [SOA](#), Developer Studio hace que se pueda desplegar y probar con sólo un par de clics. Puede agregar varios servidores al entorno Eclipse, incluyendo servidores locales y remotos. Al probar la aplicación, este hace que sea fácil de eliminar y modificar del servidor. Puede incluso permitir el despliegue en caliente en un servidor para que la aplicación se vuelve a re-desplegar de forma automática cada vez que se haga un cambio en el nivel de artefacto. También puede iniciar los servidores en modo de depuración y depurar e implementar sus aplicaciones como ellos, lo cual es muy útil durante el desarrollo. Además puede iniciar servidores con la consola [Open Services Gateway Initiative \(OSGi\)](#) habilitada, lo que permite ver el estado de los paquetes de la plataforma Carbon durante el tiempo de ejecución.

1.8. Comparación entre diferentes soluciones

La Tabla [1.1](#) muestra una comparación entre diferentes tecnologías que permiten el acceso a datos, en cuanto a:

- Interoperabilidad.
- Seguridad.

- La licencia.
- Plataformas donde se ejecutan.

Tabla 1.1. Comparación entre soluciones

Nombre	Interoperable	Seguro	Licencia	Plataforma
ODBC	Si	Si	Privativa	Windows
ADO.NET	Si	Si	Privativa	Windows
OLE DB	Si	Si	Privativa	Windows
AS	Si	Si	Libre	Windows, Linux

1.9. Metodología de desarrollo

1.9.1. CBDI - Servicio de Arquitectura e Ingeniería

Component Based Development and Integration - Service Architecture and Engineering (CBDI-SAE) es una estrategia para la entrega y manejo de las actividades del ciclo de vida de una SOA y cuenta con un fórum que se encuentra actualmente desarrollando una metodología como parte de su *framework* de referencia para dicha arquitectura; la cual consta de las siguientes vistas (BUTLER, 2007):

- **Negocio:** Analizar y comprender las necesidades del negocio, y cómo opera en términos de metas y objetivos, organización estructural, procesos e información.
- **Especificación:** Planear y especificar los servicios software desde una perspectiva independiente de la plataforma. Provee un medio para pensar en profundidad sobre la lógica de los servicios y su interrelación.
- **Implementación:** Empaquetado de los servicios en unidades de automatización, identificando interdependencias y determinando las restricciones de implementación que gobernarán el diseño interno y el despliegue de las mismas.
- **Despliegue:** Exploración y definición de la plataforma para la ejecución de los servicios. Permite el mapeo de la vista de implementación a unidades de automatización, construyendo una configuración óptima de la plataforma.
- **Tecnología:** Asegurar que las tecnologías estén en su lugar para habilitar el ciclo de vida de los servicios a todos los niveles. Desde planificación hacia especificación, diseño y ejecución hasta su retiro.

El proceso de *CBDI-SAE* tiene como objetivo abarcar todo el ciclo de vida de SOA, incluido el despliegue, la supervisión y las actividades de gobierno. Además se propone soportar estándares de desarrollo y sugerir que las empresas vendedoras de soluciones aplicables en SOA implementen sus herramientas de acuerdo a estándares basados en meta-modelos para que el trabajo a realizar sea más fácil y efectivo.

Modelo de referencia CBDI-SAE (SPROTT, 2010)

Es un *framework* que extiende el modelo de referencia de *Organization for the Advancement of Structured Information Standards (OASIS)* con el objetivo de proveer una vista conceptual más detallada, a un nivel profesional, que pueda simplemente ser mapeado al modelo *OASIS*, más una arquitectura de referencia y modelos de procesos que informen las tareas de arquitectura. Las partes esenciales en este modelo de referencia proponen ser guía para la arquitectura, aprovisionamiento, ensamblado de la solución y tareas de tipo operacionales y de gobierno. También deja en claro su foco en estándares, protocolos, herramientas, patrones y la identificación de políticas y gobierno.

Principios sobre los que se basa CBDI-SAE son:

- Nivel de empresa: atiende a las necesidades de la empresa en su totalidad.
- Dirigido por políticas: separación de la designación de políticas de las responsabilidades del proyecto.
- Basado en gobierno: asegurando que se cumplan las políticas.
- Estandarización: reducción de la complejidad a través de servicios compartidos para el negocio y la infraestructura.
- Abstracción: permite una mayor agilidad entregando servicios abstractos que puedan soportar varios contextos.
- Composición: soporta líneas de requerimientos del negocio específicas que reusan e incorporan estándares del negocio y servicios tecnológicos.
- Modularidad: reduce la dependencia y por esto mismo el horizonte de cambio.
- Virtualización: crea transparencia sobre los recursos subyacentes y luego la independencia de proveer y demandar decisiones del ciclo de vida y adquisiciones.
- Dirigido a futuro: comprensión de las necesidades reales del negocio para la agilidad y estabilidad de ingeniería donde sea apropiado y donde sea realmente necesario.
- Arquitectura inherentemente ágil: implementa una arquitectura de capas que facilita el cambio.
- Basado en modelo: usa modelos de forma proactiva para manejar el alcance a los principios acordados.

Meta Modelo SOA

Contiene una definición detallada de todos los artefactos que se necesitan para gestionar el ciclo de vida completo de un servicio, de tal manera que se pueda establecer normas, técnicas, herramientas y políticas adecuadas que aseguren la existencia de una trazabilidad completa desde la perspectiva del negocio a través del servicio ejecutado.

Define una arquitectura de referencia partiendo de un *framework* que unifique de forma cohesiva las áreas y componentes de una arquitectura empresarial convencional con los cambios que introduce SOA. Comprende una serie de áreas que proveen una estructura de clasificación para documentar librerías o base de datos que contengan o hagan alusión a estándares, herramientas, políticas, plantillas, patrones y componentes reusables que formen la arquitectura de referencia SOA.

Estructura de Proceso: Se conoce que cuenta con cuatro capas y que dentro de las mismas aparecen diferentes disciplinas en las que se desarrollan las actividades que dan solución a determinado proceso. No se cuenta con una descripción ni de las capas existentes ni de las disciplinas contenidas en dichas capas, por lo que se puede tomar como punto de partida este esqueleto y desarrollarlo para confeccionar un *framework* propio.

Un punto a favor es que especifica un plan de adopción SOA. Además, se basa en estándares, protocolos, herramientas, patrones, identificación de políticas y gobierno. Sin embargo, hasta el momento no se encuentran implementaciones llevadas a cabo mediante este *framework* que permitan determinar mejor sus beneficios y limitaciones en la práctica.

1.9.2. Propuesta de Metodología SOA para el CDAE

Esta propuesta está basada en la metodología de CBDI-SAE y agrupa el desarrollo de una SOA en las siguientes disciplinas:

- **Disciplina de Arquitectura y Diseño Orientada a Servicios:** Permite realizar un mapeo de la arquitectura de referencia a las especificaciones concretas de la empresa obtenidas a partir de la captura de los requerimientos del negocio. Prepara y evoluciona el Plan del Portafolio de servicios para el proyecto.
- **Disciplina de Arquitectura y Diseño de las Soluciones:** Se encarga de desarrollar la arquitectura de referencia de las soluciones de forma que cumpla con las especificaciones del cliente y del equipo de desarrollo.
- **Disciplina de Diseño de Servicios:** Es la encargada de tomar los servicios provenientes de la disciplina Arquitectura y Diseño Orientados a Servicios y formalizar su especificación y diseño, certificándolos como mecanismo de liberación de los mismos para su implementación y uso en las aplicaciones a desarrollar en el negocio.
- **Disciplina de Implementación de Servicios:** Se encarga de describir, implementar y probar las unidades de automatización requeridas para realizar los servicios.
- **Disciplina de Diseño e Instalación de la plataforma de soluciones y servicios:** Se encarga del despliegue de las soluciones y servicios en sus ambientes de despliegue de acuerdo con sus especificaciones y requerimientos de diseño.

1.10. Conclusiones parciales

En el estudio realizado se analizaron diferentes conceptos referentes a seguridad informática, interoperabilidad, almacén de datos y servicios. Además se realizó una comparación entre diferentes tecnologías que permiten exponer datos como son las [API](#) y [AS](#), donde este último fue escogido para la realización de la propuesta de solución por poder ejecutarse en diferentes plataformas y lograr una mayor abstracción entre las aplicaciones y la fuente de datos en sus servicios. También se relacionaron un conjunto de herramientas y tecnologías que permiten dar solución al problema planteado mediante el uso de servicios; y por último se decidió utilizar la propuesta de metodología [SOA](#) para el [CDAE](#) como metodología de desarrollo ya que cuenta con una documentación clara y detallada en cuanto a los artefactos a generar en cada fase(disciplina) y cumple con las necesidades de las soluciones desarrolladas en el [CDAE](#).

2.1. Situación de la organización

La empresa **INFOTUR** cuenta con 14 portales web que tienen actualmente una base de datos cada uno.

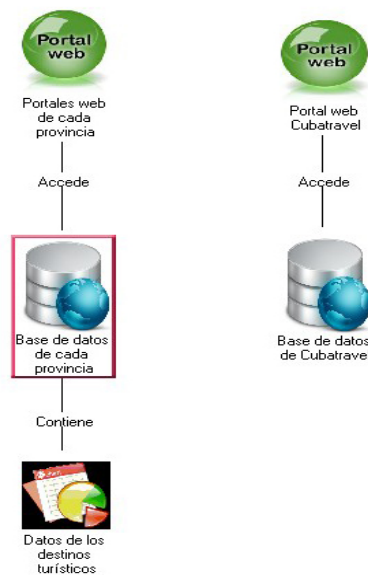


Figura 2.1. Situación Actual

Además dicha empresa tiene un almacén de datos que se diseñó e implementó para guardar los datos principales de los diferentes portales. Dicho almacén cuenta con 91 tablas, la mayoría estrechamente relacionadas entre sí como muestra la figura 2.2. **INFOTUR** ha presentado problemas con la puesta en funcionamiento del almacén de datos ya que no cuenta con un software que permita exponer de manera segura e interoperable la información que necesitan los portales.

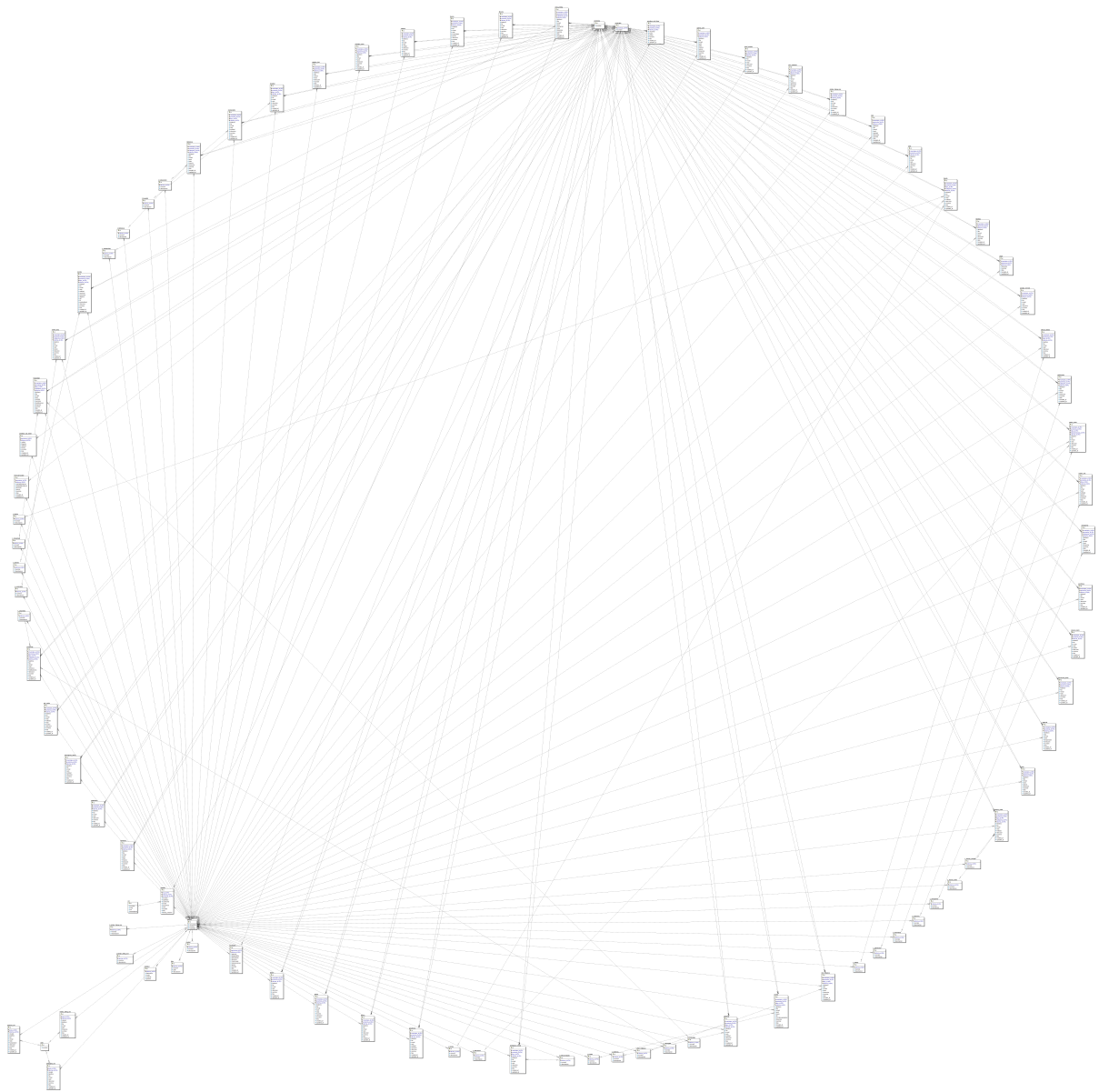


Figura 2.2. Diseño del Almacén de Datos

2.2. Identificación de servicios

Con el objetivo de identificar los servicios candidatos se entrevistó al cliente, el cual expuso que los datos se deberían exponer según la estructura expuesta en el portal web: www.infotur.cu. En el mismo se identificaron los requisitos funcionales que debería cumplir la capa de servicios web y se especificaron en la siguiente tabla 2.1 donde se utilizó la nomenclatura RF #, nombre descriptivo del requisito (**Nombre**), breve descripción del requisito (**Descripción**), prioridad del requisito para el cliente (**Prioridad**), **Costo de**

desarrollo.

Tabla 2.1. Especificación de requisitos funcionales

No	Nombre	Descripción	Prioridad	Costo de desarrollo
RF 1	Exponer información de Bancos	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email y foto de los Bancos	Media	Baja
RF 2	Exponer información de Cadecas	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto y código de las Cadecas	Media	Baja
RF 3	Exponer información de Centros Nocturnos	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto, tipo, entidad y precio de los Centros Nocturnos	Media	Baja
RF 4	Exponer información de Cafeterías	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto, tipo y entidad de las Cafeterías	Media	Baja
RF 5	Exponer información de Restaurantes	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto, tipo y entidad de los Restaurantes	Media	Baja
RF 6	Exponer información de Mensajería internacional	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto y entidad de Centros de Mensajería internacional	Media	Baja
RF 7	Exponer información de Servicios de Internet	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto y servicios de Centros que prestan el Servicio de Internet	Media	Baja
RF 8	Exponer información de Telefonía celular	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email y foto de Centros de Telefonía Celular	Media	Baja
RF 9	Exponer información de Bibliotecas	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto, categoría, horario y días de las Bibliotecas	Media	Baja
RF 10	Exponer información de Cines	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto y precio de los Cines	Media	Baja
RF 11	Exponer información de Galerías de Arte	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto, precio, horario y días de las Galerías de Arte	Media	Baja

Continúa en la página siguiente

2.2. IDENTIFICACIÓN DE SERVICIOS

Tabla 2.1. continuación de la página anterior

No	Nombre	Descripción	Prioridad	Costo de desarrollo
RF 12	Exponer información de Librerías	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email y foto de las Librerías	Media	Baja
RF 13	Exponer información de Teatros	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email y foto de los Teatros	Media	Baja
RF 14	Exponer información de Eventos	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto, tipo, fecha inicio, fecha fin y comité organizador de los Eventos	Media	Baja
RF 15	Exponer información de Locaciones para eventos	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email y foto de Locales para eventos	Media	Baja
RF 16	Exponer información de Excursiones	Dado un idioma y una provincia, devolver nombre, agencia, lugar de partida, descripción, duración, mínimo pax, observaciones, precio y foto de Agencias de Excursiones	Media	Baja
RF 17	Exponer información de Hospedaje	Dado un idioma y una provincia, devolver nombre, dirección, foto, tipo, categoría, teléfono, fax, email, web, entidad, servicios y habitaciones de Agencias de Hospedaje	Media	Baja
RF 18	Exponer información de Dónde informarse	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto y tipo Centros de información turística	Media	Baja
RF 19	Exponer información de Museos y monumentos	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto, entidad, tipo y categoría de los Museos y Monumentos	Media	Baja
RF 20	Exponer información de Áreas protegidas	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto, categoría, comercial y área de las Áreas Protegidas	Media	Baja
RF 21	Exponer información de Caza	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email y foto de Agencias de Caza	Media	Baja
RF 22	Exponer información de Buceo	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto, hospedaje y precio de Agencias de Buceo	Media	Baja

Continúa en la página siguiente

Tabla 2.1. continuación de la página anterior

No	Nombre	Descripción	Prioridad	Costo de desarrollo
RF 23	Exponer información de Marinas	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto, tipo, entidad, servicios, ubicación, atraque, vhf, hf y profundidad de las Marinas	Media	Baja
RF 24	Exponer información de Patrimonio	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto y categoría de los Patrimonios	Media	Baja
RF 25	Exponer información de Playas	Dado un idioma y una provincia, devolver nombre, dirección, municipio y foto de las Playas	Media	Baja
RF 26	Exponer información de Religión	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto y tipo de las Instituciones Religiosas	Media	Baja
RF 27	Exponer información de Hospitales	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto y tipo de los Hospitales	Media	Baja
RF 28	Exponer información de Farmacias	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto, días y horario de las Farmacias	Media	Baja
RF 29	Exponer información de Ópticas	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto, días y horario de las Ópticas	Media	Baja
RF 30	Exponer información de Tiendas	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto, entidad, categoría y tipo de las Tiendas	Media	Baja
RF 31	Exponer información de Líneas Aéreas	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email y foto de las Agencias de Aerolíneas	Media	Baja
RF 32	Exponer información de Aeropuertos	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto y categoría de los Aeropuertos	Media	Baja
RF 33	Exponer información de Bicicletas	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email y foto de donde se Alquilan Bicicletas	Media	Baja
RF 34	Exponer información de Cruceros	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto y puerto de las Terminales de Cruceros	Media	Baja
Continúa en la página siguiente				

Tabla 2.1. continuación de la página anterior

No	Nombre	Descripción	Prioridad	Costo de desarrollo
RF 35	Exponer información de los Itinerarios de ferrocarriles	Dado un idioma y una provincia, devolver nombre, foto, fecha salida, fecha llegada, terminal y precio de los Itinerarios de los Trenes	Media	Baja
RF 36	Exponer información de Terminales de ferrocarriles	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email y foto de Terminales de Ferrocarriles	Media	Baja
RF 37	Exponer información de Gasolineras	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto, entidad y servicios de Gasolineras	Media	Baja
RF 38	Exponer información de los Itinerarios Ómnibus Nacionales	Dado un idioma y una provincia, devolver nombre, foto, fecha salida, fecha llegada, terminal y precio de los Itinerarios de los Ómnibus Nacionales	Media	Baja
RF 39	Exponer información de los Itinerarios Ómnibus Urbanos	Dado un idioma y una provincia, devolver nombre, foto, municipio, recorrido, lugar y precio de los Itinerarios de los Ómnibus Urbanos	Media	Baja
RF 40	Exponer información de Terminales de Ómnibus	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email y foto de Terminales de Ómnibus	Media	Baja
RF 41	Exponer información de Taxis	Dado un idioma y una provincia, devolver nombre, dirección, teléfono, municipio, fax, web, email, foto y entidad de Agencias de Taxis	Media	Baja

También se identificaron un conjunto de Requisitos no Funcionales(RnF), los cuales se agruparon en categorías:

Eficiencia:

RnF 1: El tiempo promedio de respuesta de los servicios debe ser menor que 45 milisegundos.

RnF 2: El tiempo máximo de respuesta de los servicios no debe ser mayor que 4 segundos.

Restricciones de diseño e implementación:

RnF 3: Utilizar el [WSDL](#) en su versión 1.1.

RnF 3.2: Usar del patrón *Request/Response*.

RnF 3.4: Usar el estilo diseño *document-literal*.

RnF 4: La información expuesta solo puede ser accedida desde ip y/o dominios específicos.

De las técnicas de identificación de servicios expuestas en el capítulo anterior se decidió utilizar Buttom-Up ya que se conocen las diferentes funcionalidades expuestas por los portales y no los procesos que realiza

la empresa.

Con el objetivo de dar cumplimiento a lo planteado por dicha técnica se analizaron los requerimientos 2.1 que tienen los portales y se identificaron un grupo de servicios. En la siguiente tabla 2.2 se muestran los nombres de los servicios identificados, las operaciones de los mismos, los datos de salidas con sus tipos de datos correspondientes a cada operación, así como el requisito funcional al cual se va a dar cumplimiento. Como los datos de entradas son los mismos para todas las operaciones se decidió no incluirlos en la tabla, los mismos son: provincia_id e idioma_id ambos de tipo *integer*.

Tabla 2.2. Especificación de los servicios

Servicio	Operación	Salida	RF #
Canje_Monedas	GetCadecasXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto	1
	GetBancosXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, código	2
Centro_Nocturno	GetCentrosNocturnosXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, tipo, entidad, precio	3
Comida	GetRestaurantesXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, tipo, entidad	4
	GetCafeteriasXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, tipo, entidad	5
Comunicacion	GetMensajeInternacionalXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, entidad	6
	GetServiciosInternetXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, servicios	7
	GetTelefoniaCelularXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto	8
Cultura	GetBibliotecasXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, categoria, dias, horario	9
	GetCinesXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, precio	10
	GetGaleriasArteXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, dias, horario, precio	11
	GetLibreriasXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto	12
Continúa en la página siguiente			

2.2. IDENTIFICACIÓN DE SERVICIOS

Tabla 2.2. continuación de la página anterior

Servicio	Operación	Salida	RF #
	GetTeatrosXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto	13
Evento	GetEventosXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, tipo, comiteorganizador; <i>date</i> : inicio, fin	14
	GetLocacionesEventosXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto	15
Excursion	GetExcursionesXprovincia	<i>string</i> : nombre, foto, precio, agencia, lugarpartida, descripcion, minimopax, observaciones; <i>integer</i> : duracion	16
Hospedaje	GetHospedajeXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, tipo, categoria, entidad, servicios; <i>integer</i> : habitaciones	17
Informacion	GetInformacionXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, tipo	18
Museo_Monumento	GetMuseosMonumentosXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, tipo, categoria, entidad	19
Naturaleza	GetAreasProtegidasXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, categoria, comercial, area	20
	GetCazaXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto	21
Nautica	GetBuceoXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, precio; <i>boolean</i> : hospedaje	22
	GetMarinasXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, tipo, entidad, servicios, ubicacion, vhf, hf; <i>integer</i> : atraque, profundidad	23
Patrimonio	GetPatrimoniosXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, categoria	24
Playa	GetPlayasXprovincia	<i>string</i> : nombre, direccion, municipio, foto	25
Religion	GetInstitucionesReligiosasXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, tipo	26
Continúa en la página siguiente			

Tabla 2.2. continuación de la página anterior

Servicio	Operación	Salida	RF #
Salud	GetHospitalesXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, tipo	27
	GetFarmaciasXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, dias, horario	28
	GetOpticasXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, dias, horario	29
Tienda	GetTiendaXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, tipo, categoria, entidad	30
Transportacion	GetAerolineasXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto	31
	GetAeropuertosXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, categoria	32
	GetBicicletasXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto	33
	GetCruceirosXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, puerto	34
	GetFerrocarrilesItinerariosXprovincia	<i>string</i> : nombre, foto, precio, terminal; <i>dateTime</i> : salidafechahora, llegadafechahora	35
	GetFerrocarrilesTerminalesXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto	36
	GetGasolinerasXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, entidad, servicios	37
	GetOmnibusNaItinerariosXprovincia	<i>string</i> : nombre, foto, precio, entidad; <i>dateTime</i> : salida, llegada	38
	GetOmnibusUrbItinerariosXprovincia	<i>string</i> : nombre, foto, precio, lugar, municipio, recorrido	39
	GetOmnibusTerminalesXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto	40
	GetTaxisXprovincia	<i>string</i> : nombre, direccion, telefono, municipio, fax, web, email, foto, entidad	41

2.3. Diseño

En esta fase del ciclo de vida de un servicio, que coincide con la disciplina de diseño de servicio de la metodología de desarrollo escogida, no se realizó ya que una de las ventajas del [AS](#) es que genera de manera automática los [WSDL](#) de los servicios. No obstante el autor de la presente investigación decidió abordar algunos elementos sobre el [WSDL](#).

Un [WSDL](#) usa los siguientes elementos (CHRISTENSEN; CURBERA; MEREDITH y WEERAWARANA, 2001):

- **Types:** Contenedor para definiciones de tipo de datos usando algún sistema de tipo.
- **Message:** definición abstracta, escrita de los datos siendo comunicados.
- **Operación:** descripción abstracta de una acción sustentada por el servicio.
- **Port Type:** conjunto abstracto de operaciones sustentadas por uno o más *endpoint*.
- **Binding:** protocolo concreto y una especificación del formato de datos para un tipo particular de puerto.
- **Port:** *endpoint* solo definido como una combinación de un *binding* y una dirección de la red.
- **Service:** colección de *endpoint* relacionados.

Es imposible garantizar completamente la interoperabilidad de un servicio en particular. Sin embargo, existen varios elementos que permiten hacerle frente a los problemas más comunes y lograr una mayor interoperabilidad, entre ellos se encuentran (BALLINGER; EHNEBUSKE; FERRIS et al., 2007) :

- Utilizar la versión 1.0 de la Recomendación de *eXtensible Markup Language(XML)* de W3C.
- Utilizar la codificación UTF-8 ó UTF-16.
- Utilizar *Document-Literal*.
- Utilizar [XSD](#).

2.4. Implementación de Servicios

Una vez identificados los servicios se pasa a la implementación de los mismos, para ello existen varias formas entre ellas se encuentran:

1. Utilizando las funcionalidades que provee la herramienta [AS](#).
2. Utilizando el *plugin WSO2 Developer Studio*.

El *plugin WSO2 Developer Studio* permite implementar los servicios de manera offline, además es una recomendación de la propia empresa WSO2 por las comodidades que la misma brinda; es por esto que fue seleccionada por el autor para la implementación.

Para la implementación se creó un Proyecto de Aplicación Compuesta desde el *Dashboard 2.3*, el cual sirvió de soporte para los 18 Proyectos de Servicios de Datos *B.1* que se crearon después. Para la creación de los proyectos se especificaron los datos del Recurso de Datos (*Data Source*) a utilizar. Una vez creados los 18 proyectos de servicios de datos se pasó a la implementación de los mismos, que comenzó con la creación de las consultas *B.2*. A las mismas se le agregaron las sentencias *SQL* correspondientes, así como los elementos de entrada y salida con sus respectivos nombres y tipos de datos *B.3*. A partir de las consultas se crearon las operaciones que son como tal las funcionalidades que van a ser consumidas por el cliente del servicio. Una vez implementados los servicios se les cambió el Servidor donde se iban a desplegar ya que por defecto traen el *DataServicesServer* y después se seleccionaron todos y se exportó el Proyecto de Aplicación Compuesta como *.car B.4* para su despliegue posterior en el *AS*.

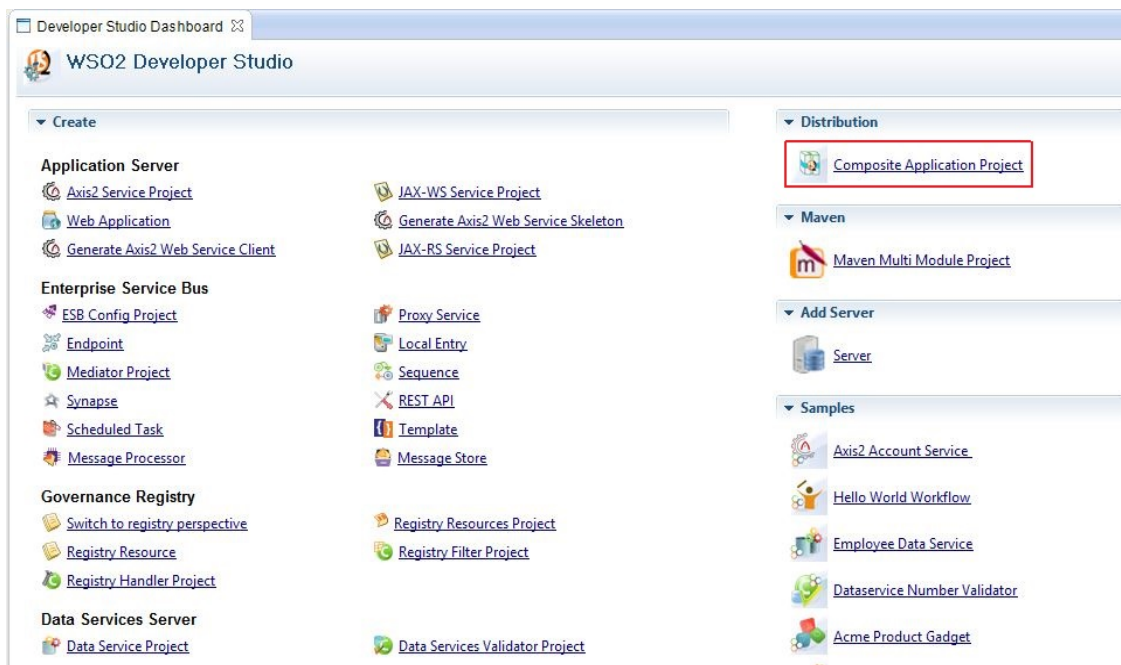


Figura 2.3. Dashboard del WSO2 Developer Studio

La implementación de los servicios *2.1* cumple con la estructura *2.2* propuesta por la empresa WSO2 para el desarrollo de los servicios de datos.

Código fuente 2.1. Implementación de un servicio

```
<data name="Canje_Monedas">
  <query id="select_banco" useConfig="default">
```

```
<sql>SELECT b.id, m.nombre municipio, telefono, fax, email, web
, direccion, b.nombre, foto FROM banco b, municipio m WHERE
b.provincia_id=? AND b.idioma_id=? AND b.municipio_id=m.id</
sql>
<result element="GetBancosXprovincia" rowName="banco">
  <element column="id" name="id" xsdType="xs:integer"/>
  <element column="municipio" name="municipio" xsdType="
  xs:string"/>
  <element column="telefono" name="telefono" xsdType="
  xs:string"/>
  <element column="fax" name="fax" xsdType="xs:string"/>
  <element column="email" name="email" xsdType="xs:string"/>
  <element column="web" name="web" xsdType="xs:string"/>
  <element column="direccion" name="direccion" xsdType="
  xs:string"/>
  <element column="nombre" name="nombre" xsdType="xs:string"/
  >
  <element column="foto" name="foto" xsdType="xs:string"/>
</result>
<param name="provincia_id" sqlType="INTEGER"/>
<param name="idioma_id" sqlType="INTEGER"/>
</query>
<query id="select_cadeca" useConfig="default">
  <sql>SELECT c.id, m.nombre municipio, telefono, fax, email, web
, codigo, direccion, c.nombre, foto FROM cadeca c, municipio
m WHERE c.provincia_id=? AND c.idioma_id=? AND c.
municipio_id=m.id</sql>
  <result element="GetCadecasXprovincia" rowName="cadeca">
    <element column="id" name="id" xsdType="xs:integer"/>
    <element column="municipio" name="municipio" xsdType="
    xs:string"/>
    <element column="telefono" name="telefono" xsdType="
    xs:string"/>
    <element column="fax" name="fax" xsdType="xs:string"/>
    <element column="email" name="email" xsdType="xs:string"/>
    <element column="web" name="web" xsdType="xs:string"/>
```

```

    <element column="codigo" name="codigo" xsdType="xs:string"/
    >
    <element column="direccion" name="direccion" xsdType="
        xs:string"/>
    <element column="nombre" name="nombre" xsdType="xs:string"/
    >
    <element column="foto" name="foto" xsdType="xs:string"/>
</result>
<param name="provincia_id" sqlType="INTEGER"/>
<param name="idioma_id" sqlType="INTEGER"/>
</query>
<operation name="GetBancosXprovincia">
    <call-query href="select_banco">
        <with-param name="provincia_id" query-param="provincia_id"/
        >
        <with-param name="idioma_id" query-param="idioma_id"/>
    </call-query>
</operation>
<operation name="GetCadecasXprovincia">
    <call-query href="select_cadeca">
        <with-param name="provincia_id" query-param="provincia_id"/
        >
        <with-param name="idioma_id" query-param="idioma_id"/>
    </call-query>
</operation>
<config id="default">
    <property name="carbon_datasource_name">Conexion</property>
</config>
</data>

```

Código fuente 2.2. Estructura del DSRL

```

<data name="xs:NMTOKEN">
    <config id="xs:ID">
        <property name="xs:NMTOKEN">xs:urType</property>
    </config>
    (<query id="xs:ID"? useConfig="xs:ConfigID" >
        <sql dialect="xs:NMTOKEN">xs:string</sql>

```

```

    <result element="xs:QName" rowName="xs:QName" />
      (<element name="xs:QName" column="xs:NMTOKEN" xsdType="
        xs:QName"/> )+
  </result>
  (<param name="xs:NMTOKEN" sqlType="xs:string" />)*
</query>)+
(<operation name="xs:NCName">
  <description>"xs:string"</description>
  <call-query href="xs:IDREF" />
    (<with-param name="xs:NMTOKEN" query-param="xs:NMTOKEN"/>)*
  </call-query>
</operation>)+
</data>

```

Despliegue

Una vez implementados y exportados los servicios se realizó el despliegue de los mismos en el [AS 2.4](#). Pero para ello fue necesario adicionarle el *driver JDBC* y configurar un nuevo *data source*.

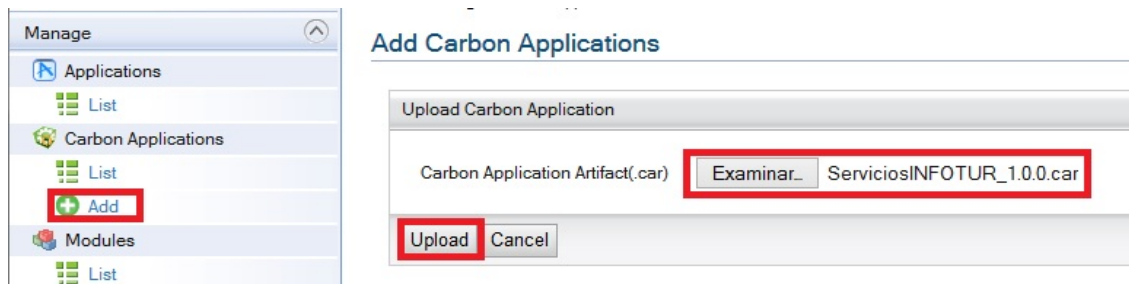


Figura 2.4. Despliegando el Proyecto de Aplicaciones Compuestas

Response Caching

Esta opción permite mejorar los tiempos de respuesta de los servicios según la configuración. Para ello se configuran los siguientes parámetros:

- **Hash Generador:** Especifica el nombre completo de clase del generador de valor *hash*. Esta clase implementa la interfaz `org.wso2.caching.digest.DigestGenerator`, la cual es responsable de generar un valor *hash* para cada mensaje de solicitud, de manera que el mensaje de respuesta correspondiente puede ser localizado en la caché. El valor predeterminado es `org.wso2.caching.digest.DOMHASHGenerator`.

- **El intervalo de espera:** Muestra el período de tiempo, en milisegundos, durante el cual una respuesta en caché se mantiene en la memoria. El conteo se inicia desde el momento en que se recibió la primera solicitud. Cualquier solicitud recibida después de que el contador ha comenzado y este dentro del período de tiempo de espera se sirve con la respuesta almacenada en caché después de recibir la primera solicitud. Una vez que el tiempo de espera expira, la caché se actualizará con la respuesta generada por la primera petición que se recibió después de la expiración.
- **Tamaño máximo de la caché:** Almacena el número máximo de mensajes de respuesta que se almacenan en caché. Si el número máximo de respuestas ya están en caché y ninguno de los tiempos de espera de las respuestas en caché han expirado, entonces las respuestas futuras no se almacenan en caché hasta que esta memoria no este lo suficientemente libre como para dar cabida a una nueva respuesta.
- **Tamaño máximo de mensaje:** muestra el tamaño máximo de un mensaje de respuesta que se almacena en caché. Los mensajes de respuesta con mayor tamaño que el valor anterior, no se almacenan en caché.

Para la solución propuesta quedó configurada así:

Caching Configuration for Canje_Monedas Service

Enable Caching? ▾

Hash Generator	<input type="text" value="org.wso2.carbon.caching.core.digest.DOMHASHGenerator"/>
Timeout (ms)	<input type="text" value="20000"/>
Maximum Cache Size (in number of messages)	<input type="text" value="1000"/>
Maximum Message Size (bytes)	<input type="text" value="1000"/>

Figura 2.5. Configurando Response Caching

2.4.1. Implementación de la seguridad

La herramienta [AS](#) cuenta con una serie de escenarios de seguridad para los servicios desplegados como son:

UsernameToken

El escenario 2.6 es usado para la autenticación. **Funcionamiento:** El cliente envía una petición a un servicio del servidor junto con su usuario y contraseña. El servidor valida que el usuario y la contraseña son correctas, son iguales que las que el tiene almacenada, y si son validos, la petición es ejecutada por el servicio. De lo contrario si el usuario o la contraseña enviada por el cliente no son válidos, no se accede al servicio.

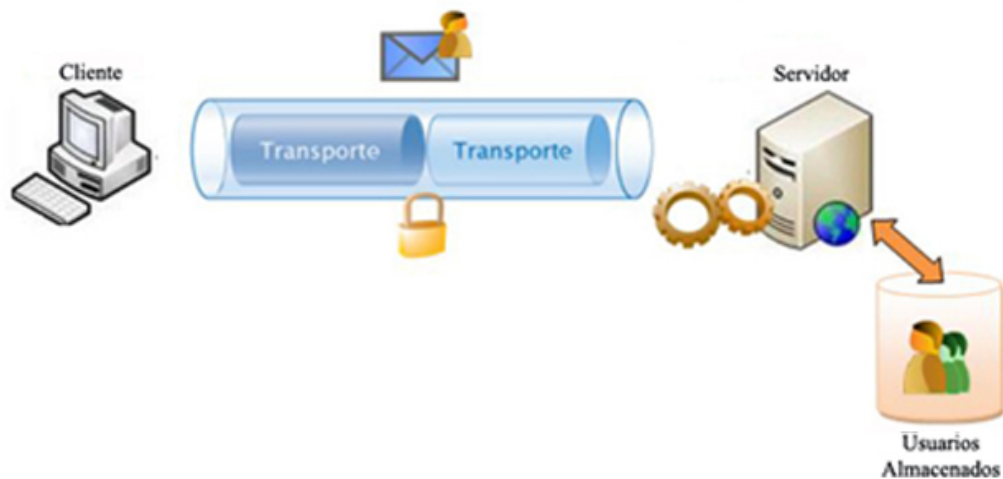


Figura 2.6. Escenario *UsernameToken*

Non-repudiation

Este escenario 2.7 es usado para evitar que la persona(usuario) que envía una petición no pueda decir que no lo hizo. **Funcionamiento:** El cliente envía una petición a un servicio del servidor firmada con su clave privada. Cuando el servidor recibe dicha petición verifica si la firma es correcta, que el certificado del cliente se encuentra entre los almacenados, y si es válida, la petición es ejecutada por el servicio. De lo contrario no se accede al servicio.

Se decide implementar el escenario *UsernameToken* 2.6 ya que influye en menor medida en el tiempo de respuesta de los servicios.

Access Throttling

También se habilitó la opción *Access Throttling* 2.8 para permitir/denegar el acceso a cada servicio desde direcciones de ip y/o dominio específicos. Esto tiene la ventaja de que se prohíbe el acceso solamente al servicio y no al servidor como tal, por lo que se podrá seguir accediendo normalmente desde los ip a las demás aplicaciones hospedadas en él. Además permite controlar el acceso configurando los siguiente parámetros:



Figura 2.7. Escenario *Non-repudiation*

- Cantidad máxima de peticiones: Contendrá el número máximo de solicitudes atendidas en el intervalo de tiempo especificado por Unidad de tiempo.
- Unidad de tiempo (UT): Es el período de tiempo en milisegundos durante el cual es atendida la cantidad máxima de peticiones. El acelerador empieza a contar el número de unidades desde el momento en que es activado y el número de solicitudes atendidas dentro de ese período.
- Prohibir por períodos de tiempo(PPT): Si se llega a la cantidad máxima de peticiones antes que a la unidad de tiempo, entonces este es el período durante el cual no se permite atender las solicitudes entrantes. Al establecer este valor, la ranura de la unidad de tiempo se altera de la siguiente manera: $UT = \text{el tiempo en el que llegó la última petición atendida} + PPT$.

Throttling Configuration for Canje_Monedas Service

Enable Throttling? Yes No

Maximum Concurrent Accesses

Range	Type	Max Request Count	Unit Time (ms)	Prohibit Time Period (ms)	Access	Actions
<input type="text" value="other"/>	<input type="text" value="IP"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="Deny"/>	<input type="text" value="Delete"/>
<input type="text" value="other"/>	<input type="text" value="DOMAIN"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="Deny"/>	<input type="text" value="Delete"/>
<input type="text" value="10.88.140"/>	<input type="text" value="IP"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="Allow"/>	<input type="text" value="Delete"/>
<input type="text" value="*.uci.cu"/>	<input type="text" value="DOMAIN"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="Allow"/>	<input type="text" value="Delete"/>

Figura 2.8. Añadiendo la opción *Access Throttling*

2.5. Arquitectura de Despliegue

La figura 2.9 muestra una de las formas de como puede quedar el despliegue de la capa de servicios. En la misma se puede observar un grupo de portales web que se relacionan por https con el AS. En él cual se encuentran desplegados todos los servicios que le van a proveer la información que necesitan los portales. También se puede observar la relación del AS por TCP con el almacén de datos.

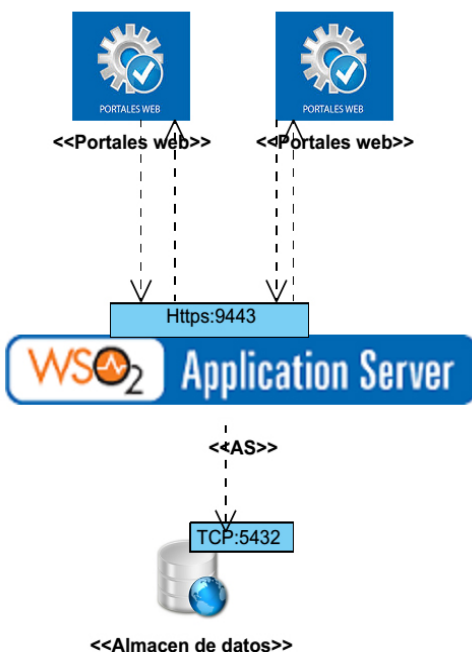


Figura 2.9. Propuesta de diagrama de despliegue simple

Para una mayor disponibilidad¹ de los servicios se propone un despliegue de la siguiente manera 2.10 donde la herramienta *WSO2 Elastic Load Balancer* permite, como su nombre lo indica, balancear la carga emitida por los usuarios entre los AS.

¹es el grado en que una aplicación o servicio está funcionando cuando y como esperan los usuarios a los que está dirigido

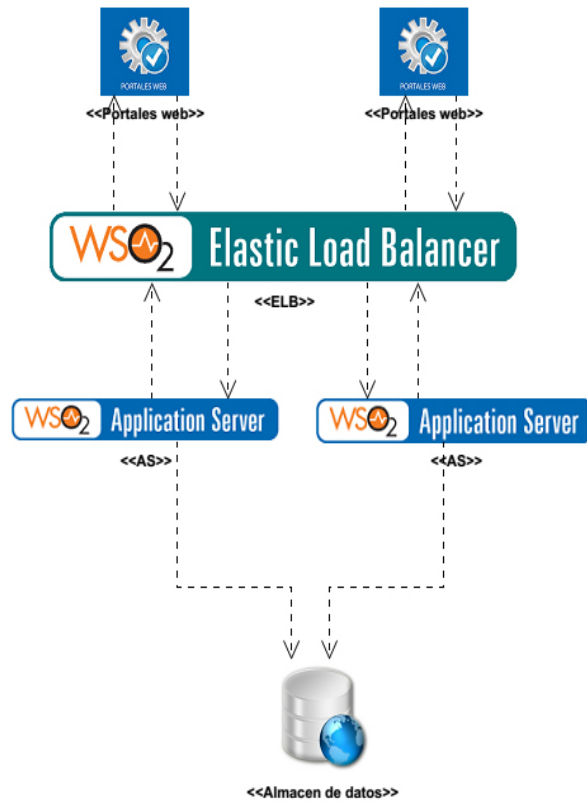


Figura 2.10. Propuesta de diagrama de despliegue

3.1. Pruebas de Diseño

Estas pruebas se realizaron con el objetivo de validar un grupo de parámetros sobre los [WSDL](#), en su versión 1.1, generados por el [AS](#). Parámetros:

- Correcto uso del patrón *Request/Response*.
- Uso de *Document-literal*
- Resultado de la validación del perfil básico de WS-I(para este parámetro se utilizó la *Test Suite SOA-PUI* específicamente la opción *Check WSI Compliance*).

Tabla 3.1. Pruebas de Diseño

Servicio	Request/Response	Document-literal	WSI Compliance Resultado
Canje_Monedas	Correcto	Si	Passed
Centro_Nocturno	Correcto	Si	Passed
Comida	Correcto	Si	Passed
Comunicacion	Correcto	Si	Passed
Cultura	Correcto	Si	Passed
Evento	Correcto	Si	Passed
Excursion	Correcto	Si	Passed
Informacion	Correcto	Si	Passed
Museo_Monumento	Correcto	Si	Passed
Naturaleza	Correcto	Si	Passed
Nautica	Correcto	Si	Passed
Patrimonio	Correcto	Si	Passed
Playa	Correcto	Si	Passed
Continúa en la página siguiente			

Tabla 3.1. continuación de la página anterior

Servicio	Request/Response	Document-literal	WSI Compliance Resultado
Religion	Correcto	Si	Passed
Salud	Correcto	Si	Passed
Tienda	Correcto	Si	Passed
Transportacion	Correcto	Si	Passed
Hospedaje	Correcto	Si	Passed

3.2. Pruebas Funcionales

Para las pruebas funcionales se creó un proyecto en la herramienta SOAPUI y se le añadieron todos los [WSDL](#) de los servicios, seleccionando para cada uno las opciones: Crear peticiones (*Create Requests*) y Crear Suite de Prueba (*Create TestSuite*). Después se creó una configuración de seguridad para poder consumir los servicios. A continuación se exponen los resultados de los servicios. El campo Entrada puede tomar los valores: Válida, que quiere decir que es un número entero y No Válida para todos los demás casos.

Tabla 3.2. Servicio Comida

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetRestaurantesXprovincia	Válida	Lista de restaurantes de esa provincia	Lista de restaurantes de esa provincia	Satisfactorio
GetRestaurantesXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetCafeteriasXprovincia	Válida	Una lista de cafeterías de esa provincia	Lista de cafeterías de esa provincia	Satisfactorio
GetCafeteriasXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Tabla 3.3. Servicio Centro_Nocturno

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetCentrosNocturnosXprovincia	Válida	Lista de centros nocturnos de esa provincia	Lista de centros nocturnos de esa provincia	Satisfactorio
GetCentrosNocturnosXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Las demás pruebas aparecen en el Anexo [C.1](#).

3.3. Pruebas de Rendimiento

Para la realización de este tipo de prueba se utilizó la *Test Suite* SOAPUI en su versión 4.5.1 la cual permitió la generación de 6000 peticiones por cada operación de servicio, las mismas realizadas por 100

3.3. PRUEBAS DE RENDIMIENTO

usuarios virtuales simultáneamente. A continuación se muestra en gráficas los tiempos promedio 3.1 y máximos 3.2 obtenidos para cada servicio y se comparan con los requisitos no funcionales correspondientes(Línea resaltada).

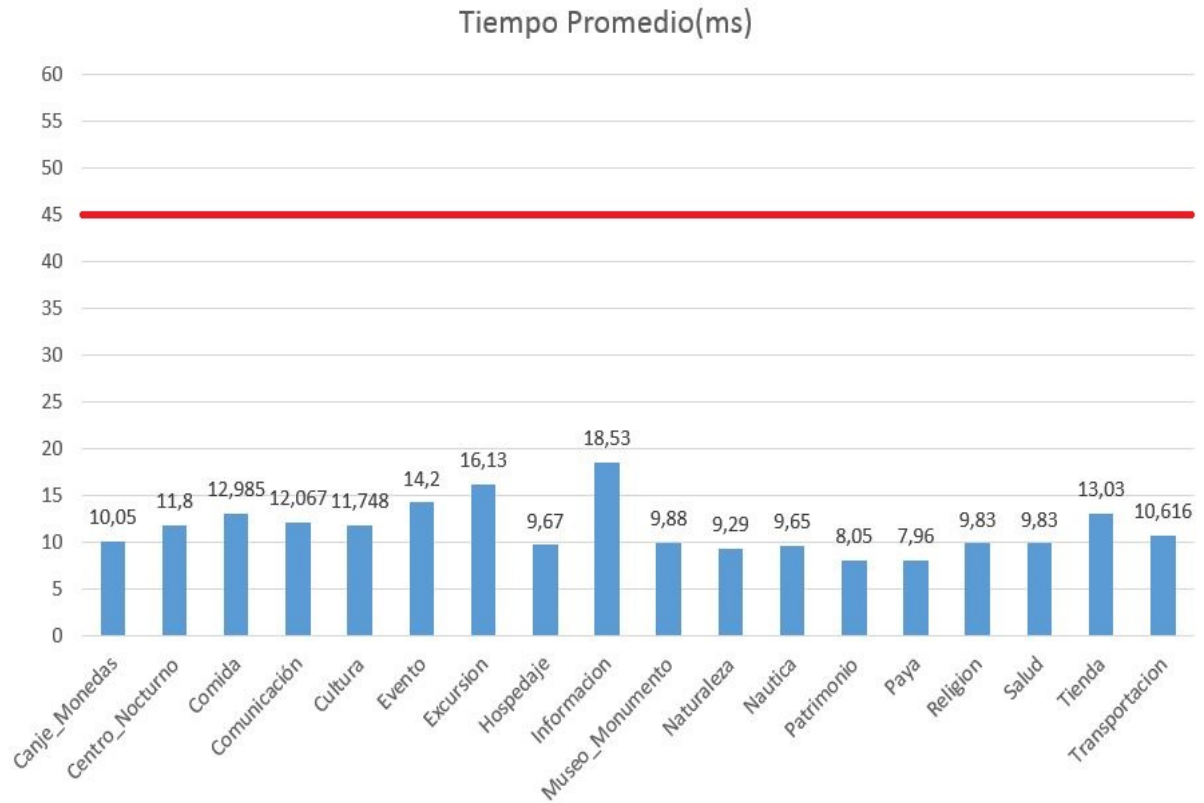


Figura 3.1. Comparación de los tiempos promedios obtenidos con el requerido

Propiedades de la PC donde se realizaron las pruebas

- Modelo: VIT P3400
- Procesador: Intel Core i5
- Memoria RAM: 4GB(DDR3)

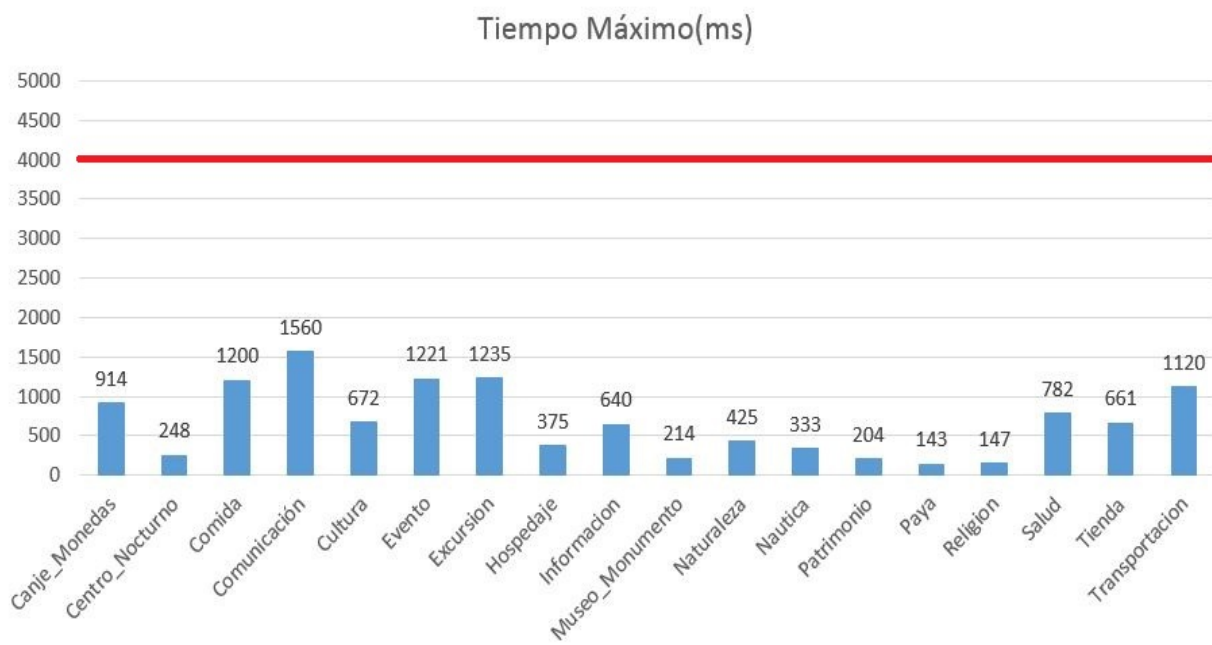


Figura 3.2. Comparación de los tiempos máximos obtenidos con el requerido

El desarrollo de este trabajo arroja las siguientes conclusiones:

- El estudio realizado en relación al acceso a datos de forma segura e interoperable a través de los servicios web, sentó las bases teóricas para seleccionar las herramientas, tecnologías y conceptos fundamentales para el desarrollo de la capa de servicios web.
- El diagnóstico realizado en [INFOTUR](#) demostró que la entidad no contaba con una infraestructura de software que permitiera exponer la información del almacén de datos operacional, por lo cual se propuso la utilización de los servicios web.
- El diseño e implementación de la capa de servicios permite que se exponga de manera segura e interoperable la información contenida en el almacén de datos operacional de [INFOTUR](#).
- La validación realizada a la capa de servicios web, a través de pruebas funcionales, de diseño y rendimiento tuvo resultados satisfactorios y demuestra la calidad de la solución desarrollada.

Recomendaciones

Se recomienda:

- Continuar evolucionando y creando nuevos servicios que le permitan a la empresa [INFOTUR](#) gestionar una mayor cantidad de información.
- Analizar que otras herramientas de la *suite* WSO2 pueden enriquecer aún más la solución propuesta.

ADO.NET *ActiveX Data Objects .NET Framework*. 9, 19

API Interfaz de Programación de Aplicaciones (por sus siglas en inglés, *Applications Programing Interface*). 8, 9, 18, 22

AS Servidor de Aplicaciones (por sus siglas en inglés, *WSO2 Application Server*). 17, 19, 22, 32, 33, 36, 37, 40, 42

CBDI-SAE *Component Based Development and Integration - Service Architecture and Engineering*. 19–21

CDAE Centro de Consultoría y Desarrollo de Arquitecturas Empresariales. 2, 22

DSRL Lenguaje de Recursos y Servicios de Datos (por sus siglas en inglés, *Data Services and Resource Language*). 35

IDE Entorno de Desarrollo Integrado (por sus siglas en inglés, *Integrated Development Environment*). 18

INFOTUR Información Turística. 2, 3, 17, 23, 46, 47

JDBC *Java Database Connectivity*. 18, 36

MINTUR Ministerio del Turismo en Cuba. 2

OASIS *Organization for the Advancement of Structured Information Standards*. 20

ODBC *Open Database Connectivity*. 8, 9, 19

ODS Almacén de Datos Operacionales, (por sus siglas en inglés, *Operational Data Store*). 5

OLE DB *Object Linking and Embedding for Databases*. 9, 19

OSGi *Open Services Gateway Initiative*. 18

SGBD Sistema de Gestión de Bases de Datos. 8

SOA Arquitectura Orientada a Servicios (por sus siglas en inglés, *Services Oriented Architecture*). [3](#), [7](#), [11](#), [12](#), [15](#), [16](#), [18](#), [19](#), [21](#), [22](#)

SOAP Protocolo de Acceso Simple a Objetos (por sus siglas en inglés, *Simple Object Access Protocol*). [14](#), [16](#), [17](#)

SQL *Structured Query Language*. [8](#), [9](#), [18](#), [33](#)

TIC Tecnologías de la Información y las Comunicaciones. [1](#), [4](#)

UCI Universidad de las Ciencias Informáticas. [2](#)

UDDI *Universal Description Discovery and Integration*. [14](#)

W3C *World Wide Web Consortium*. [14](#)

WSDL Lenguaje de Descripción de Servicios Web (por sus siglas en inglés, *Web Services Description Language*). [14–17](#), [28](#), [32](#), [42](#), [43](#)

XSD *XML Schema Definition Language*. [14](#), [32](#)

Referencias bibliográficas

- APONCIO BORGES, Rasiel. 2009. Migración de la Capa de Acceso a Datos del Sistema de Gestión de Emergencias de Seguridad Ciudadana (171). 2009.
- BALLINGER, Keith; EHNEBUSKE, David; FERRIS, Christopher et al., 2007. Basic profile version 1.2. *WS-I Specification*. 2007, págs. 1-1. Consultado 2015-06-10. Url: <http://www.ws-i.org/Profiles/BasicProfile-1.2-2007-03-28.html>).
- BLAKELEY, Jose A. 1997. Universal data access with OLE DB. En. *Compcon'97. Proceedings, IEEE*. 1997, págs. 2-7.
- BROOKS JR, Frederick P. 1975. The Mythical Man-Month. En. *ACM SIGPLAN Notices*. 1975.
- BUTLER, John. 2007. SOA Best Practice Report: The Architecture Component of the SAE Reference Framework for SOA. *CBDI-SAE*. 2007. Consultado 2014-12-07.
- CASAS GUIJARRO, Alejandro. 1997. Sistema Telefónico Multilínea con Reconocimiento de voz y acceso a Base de Datos Remota. 1997. Consultado 2015-06-03. Url: <http://lorien.die.upm.es/juancho/pfcs/ACG>).
- CASTELLS, Manuel. 2004. *La era de la información: economía, sociedad y cultura*. 2004.
- CASTILLO, Rosa María; MORATA, Jesús y ARBOL, Luis del. 2005. Operational Data Store (ODS). 2005. Consultado 2015-06-07.
- CENTER, IBM Knowledge. 2014. ¿Qué es WSDL? 2014. Consultado 2014-12-08. Url: http://www-01.ibm.com/support/knowledgecenter/SSMKHH_9.0.0/com.ibm.etools.mft.doc/ac34640_.htm?lang=es).
- CHAMORRO BORGES, Alejandro E. 2007. La interoperabilidad y los estándares abiertos, base del desarrollo de la Sociedad de la Información. *Bit*. 2007, n.º 161, págs. 36-39. Consultado 2015-06-04.
- CHRISTENSEN, Erik; CURBERA, Francisco; MEREDITH, Greg y WEERAWARANA, Sanjiva. 2001. Web Services Description Language (WSDL) 1.1. 2001. Consultado 2015-06-10. Url: <http://www.w3.org/TR/wsdl>).
- CLEMENTS, Paul C. 1996. A survey of architecture description languages. En. *Proceedings of the 8th international workshop on software specification and design*. 1996, págs. 16.

- CLEMENTS, Paul C y NORTHROP, Linda M. 1996. *Software Architecture: An Executive Overview*. 1996. Consultado 2015-01-27.
- DIJKSTRA, Edsger W. 1983. The Structure of THE -multiprogramming System. *Commun. ACM*. 1983, vol. 26, n.º 1, págs. 49-52. Consultado 2015-02-05. ISSN 0001-0782.
- ERL, Thomas. 2005. Service-oriented architecture: concepts, technology, and design. *Prentice Hall*. 2005, vol. 31, págs. 792. Consultado 2015-06-10.
- GEYER, Carol. 2006. What is UDDI? 2006. Actualizado 2006-08-28 23:06 y consultado 2014-12-07. Url: <http://uddi.xml.org/node/96>.
- GINESTÀ, M. G. y PÉREZ MORA, Oscar. 2006. *Bases de datos en PostgreSQL*. 2006. Url: http://ocw.uoc.edu/computer-science-technology-and-multimedia/bases-de-datos/bases-de-datos/P06_M2109_02152.pdf.
- GONZÁLEZ MOREIRO, José Antonio; CUADRADO SÁNCHEZ, Sonia y LARA MORATO, Jorge. 2012. Mejora de la interoperabilidad semántica para la reutilización de contenidos mediante sistemas de organización del conocimiento. *Encontros Bibli*. 2012. Consultado 2014-12-09.
- GRADMANN, Stefan. 2009. *Interoperability. A key concept for large scale, persistent digital libraries*. 2009.
- GRAHAM, I. 2008. *Requirements Modelling and Specification for Service Oriented Architecture*. 2008.
- HEVIA VEGA, Andrés. 2013. Ciclo de vida de un servicio. 2013. Consultado 2015-06-10. Url: <http://pensandoensoa.com/2013/08/07/ciclo-de-vida-de-un-servicio/>.
- JIMENEZ, G.M.C. 2010. Tecnologías de web semántica orientada al desarrollo de servicios web. 2010. Consultado 2014-12-09. Url: <http://cepra.utpl.edu.ec/handle/123456789/1706>.
- JOSUTTIS, Nicolai M. 2007. *SOA in practice: the art of distributed system design*. 2007.
- JURIC, M.B. 2010. *WS-BPEL 2.0 for SOA Composite Applications*. 2010.
- KIOSKEA. 2015. Introducción a la seguridad. 2015. Consultado 2015-06-04. Url: <http://es.kioskea.net/contents/seguridad-informatica-2020435058#622>.
- LUGO JEREZ, Carlos Augusto. 2004. Seguridad para lograr Confiabilidad y Calidad de los Servicios Digitales en Internet. 2004. Consultado 2015-06-07.
- MEJÍAS, Milagros. 2001. ODBC. 2001. Consultado 2015-06-11. Url: <http://www.oocities.org/milagrosmejias/odbc.htm>.
- MIRALBELL, Oriol. 2010. Portales de destinos turísticos en Internet: Una reflexión estratégica. *Métodos de información*. 2010, vol. 8, n.º 42, págs. 74-81.
- NAUR, Peter; RANDELL, Brian y BUXTON, John N. 1976. *Software engineering: concepts and techniques: proceedings of the NATO conferences*. 1976.

- OASIS. 2006. Reference Model for Service Oriented Architecture. *OASIS*. 2006. Consultado 2015-12-15. Url: <https://www.oasis-open.org/committees/download.php/16587/wd-soa-rm-cd1ED.pdf>.
- PERRY, Dewayne E y WOLF, Alexander L. 1992. Foundations for the study of software architecture. *ACM SIGSOFT Software Engineering Notes*. 1992, vol. 17, n.º 4, págs. 40-52. Consultado 2015-02-05.
- POLANCO, IGNACIO. 2013. LAS TECNOLOGIAS DE LA INFORMACION Y LA COMUNICACION (TIC) EN LA EDUCACION: RETOS Y POSIBILIDADES. 2013.
- REYNOLDS, John. 2005. The SOA Elevator Speech. *John Reynolds's Blog*. 2005. Consultado 2014-12-09. Url: http://weblogs.java.net/blog/johnreynolds/archive/2005/01/the_soa_elevato.html.
- REYNOSO, Carlos Billy. 2004. Introducción a la Arquitectura de Software. 2004. Consultado 2014-12-07. Url: <http://carlosreynoso.com.ar/archivos/arquitectura/Arquitectura-software.pdf>.
- SATTARI, H. y KUNJUMOHAMED, S. 2012. *Spring Web Services 2 Cookbook*. 2012.
- SPROTT, David. 2010. The CBDI-SAE Reference Framework. *CBDI-SAE*. 2010. Consultado 2015-02-03.
- TAHUITON MORA, Juan. 2011. *Arquitectura de software para aplicaciones Web*. 2011.
- THAI, Jenny. 2011. Services in SOA. *MuleSoft*. 2011. Consultado 2015-02-05. Url: <https://www.mulesoft.com/resources/esb/services-in-soa>.
- TONG, K.K.I. 2010. *Developing Web Services with Apache CXF and Axis2*. 2010.
- VEGA LEBRÚN, Carlos Arturo. 2005. *Integración de Herramientas de Tecnologías de Información: Portales Colaborativos de Trabajo Como Soporte en La Administración Del Conocimiento*. 2005. Consultado 2015-06-21. Url: www.eumed.net/tesis-doctorales/2007/cavl/cavl.zip.
- WSO2. 2014. 2014. Consultado 2014-12-09. Dirección: www.wso2.org.

Apéndices

APÉNDICE **A**

Web Services

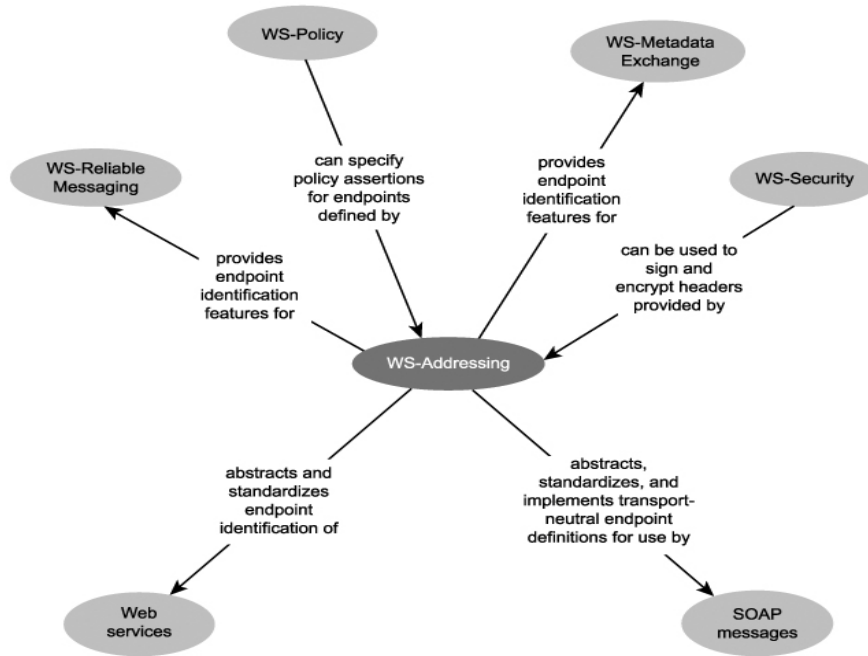


Figura A.1. *WS-Addressing*

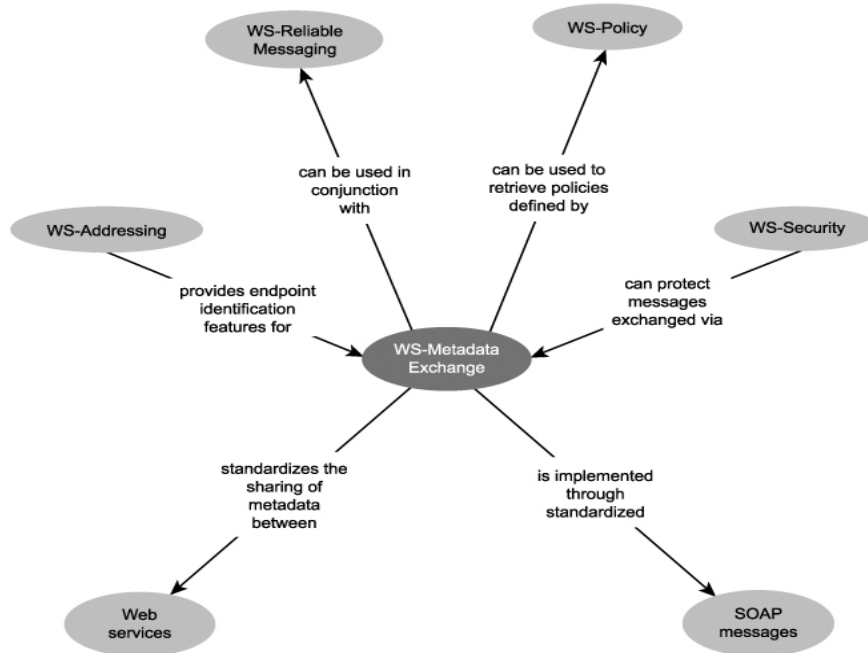


Figura A.2. *WS-MetadataExchange*

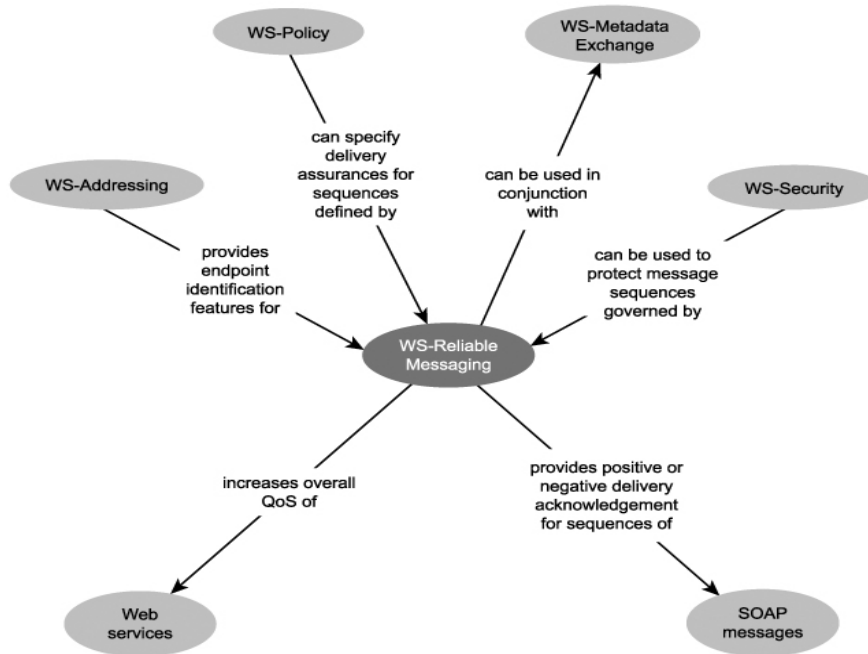


Figura A.3. *WS-ReliableMessaging*



* A separate WS-SecurityPolicy specification provides a set of predefined policy assertions for WS-Security.

Figura A.4. *WS-Policy*

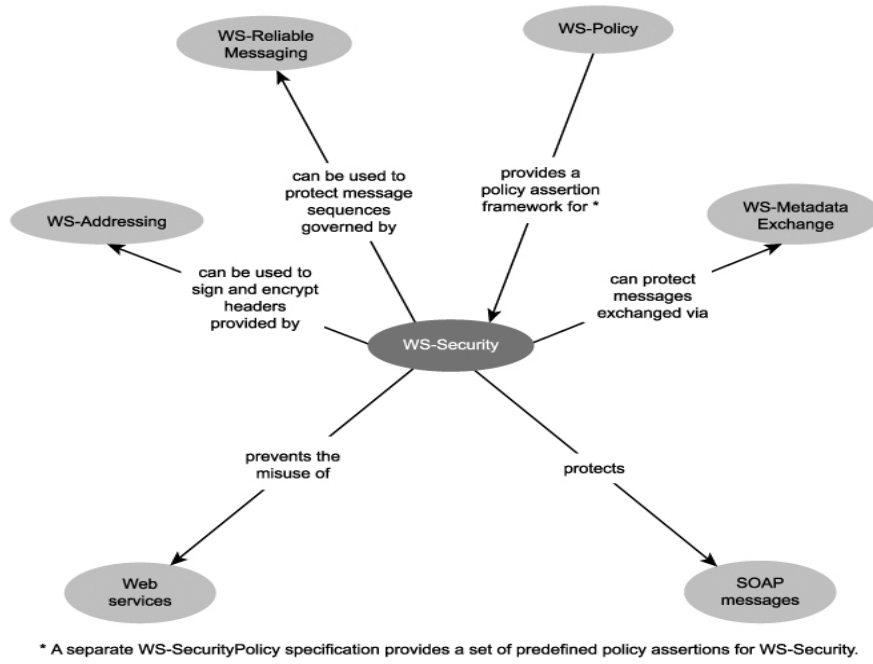


Figura A.5. *WS-Security*

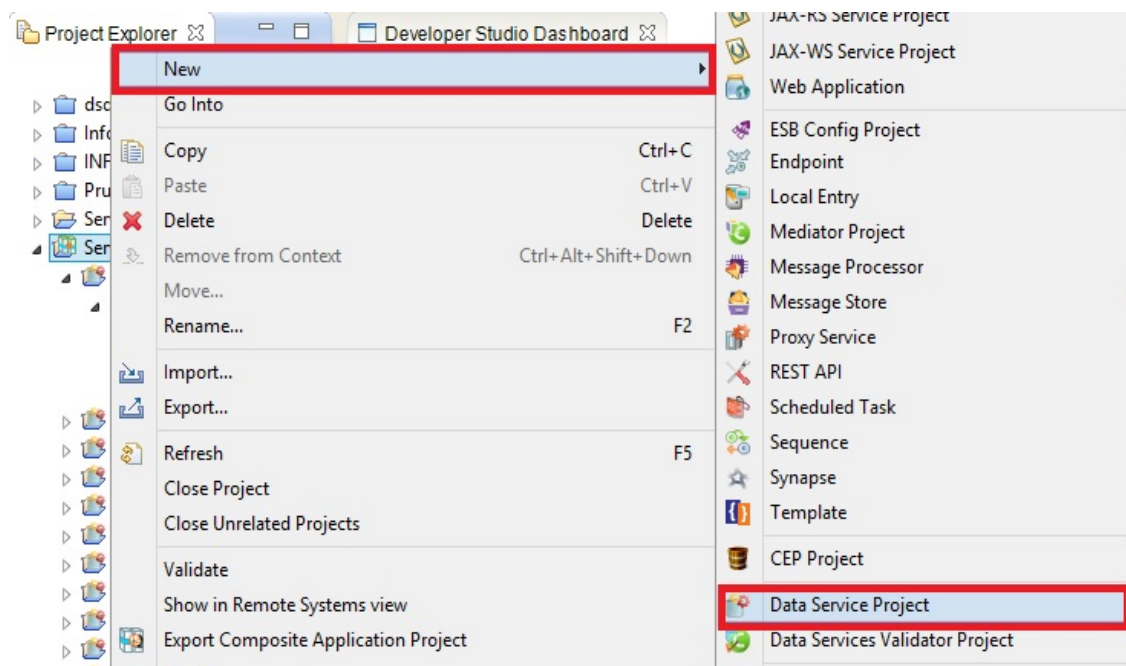


Figura B.1. Creando un Proyecto de Servicio de Datos

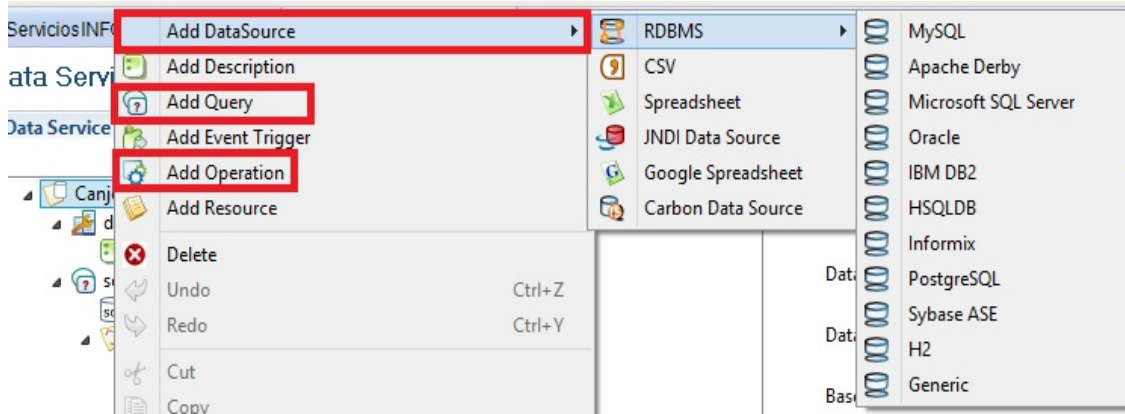


Figura B.2. Adicionando los datos necesarios

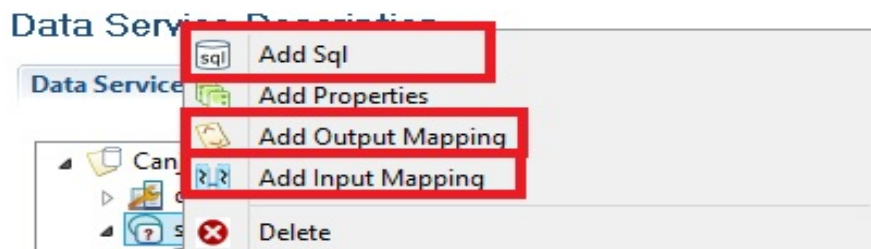


Figura B.3. Adicionando los elementos de la consulta

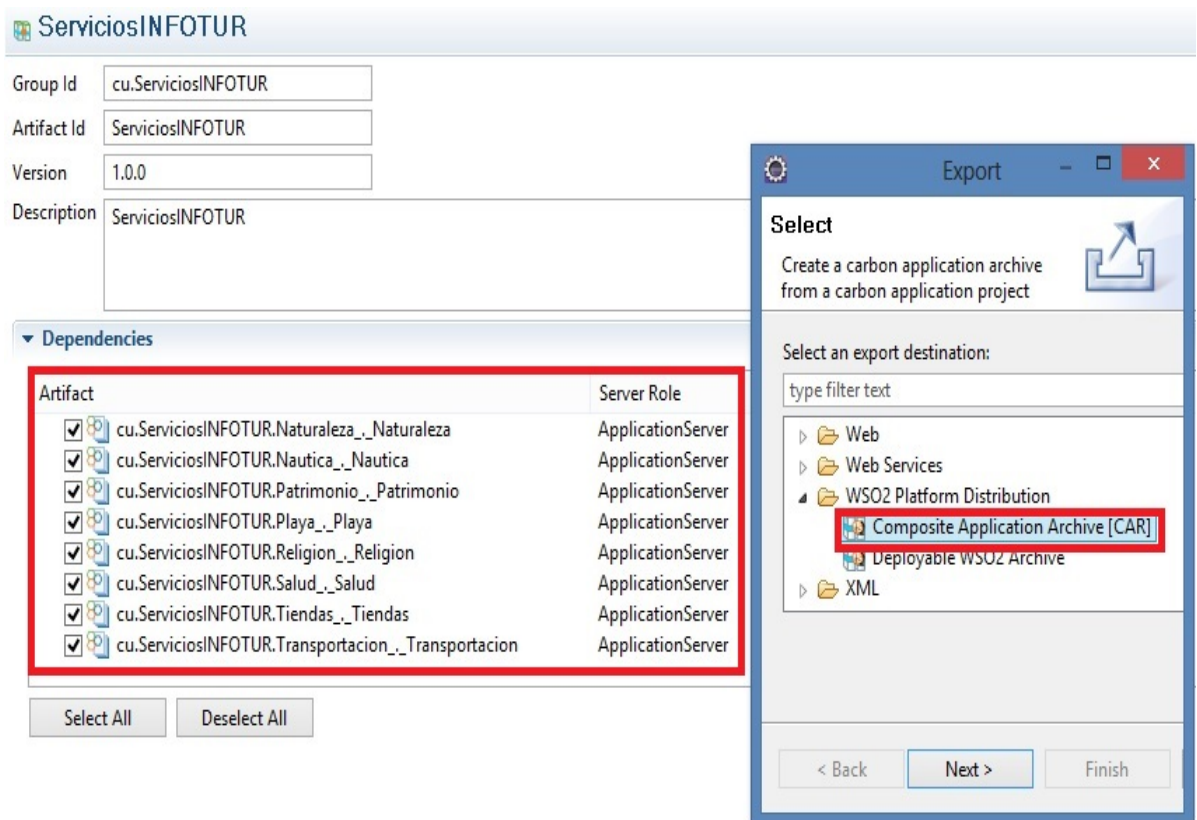


Figura B.4. Exportando

Tabla C.1. Servicio Canje_Monedas

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetBancosXprovincia	Válida	Lista con los bancos de esa provincia	Lista con los bancos de esa provincia	Satisfactorio
GetBancosXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetcadecasXprovincia	Válida	Lista con las cadecas de esa provincia	Lista con las cadecas de esa provincia	Satisfactorio
GetcadecasXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Tabla C.2. Servicio Comunicacion

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetMensajeInternacionalXprovincia	Válida	Lista con los centros de mensajería internacional de esa provincia	Lista con los centros de mensajería internacional de esa provincia	Satisfactorio
GetMensajeInternacionalXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetServiciosInternetXprovincia	Válida	Lista con los centros que prestan el servicio de internet en esa provincia	Lista con los centros que prestan el servicio de internet en esa provincia	Satisfactorio
GetServiciosInternetXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetTelefoniaCelularXprovincia	Válida	Lista con los centros de telefonía celular de esa provincia	Lista con los centros de telefonía celular de esa provincia	Satisfactorio

Continúa en la página siguiente

Tabla C.2. continuación de la página anterior

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetTelefoniaCelularXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Tabla C.3. Servicio Cultura

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetBibliotecasXprovincia	Válida	Lista con las bibliotecas de esa provincia	Lista con con las bibliotecas de esa provincia	Satisfactorio
GetBibliotecasXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetCinesXprovincia	Válida	Lista con los cines de esa provincia	Lista con los cines de esa provincia	Satisfactorio
GetCinesXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetGaleriasArteXprovincia	Válida	Lista con las galerías de arte de esa provincia	Lista con las galerías de arte de esa provincia	Satisfactorio
GetGaleriasArteXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetLibreriasXprovincia	Válida	Lista con las librerías de esa provincia	Lista con las librerías de esa provincia	Satisfactorio
GetLibreriasXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetTeatrosXprovincia	Válida	Lista con los teatros de esa provincia	Lista con los teatros de esa provincia	Satisfactorio
GetTeatrosXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Tabla C.4. Servicio Evento

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetEventosXprovincia	Válida	Lista con los eventos en esa provincia	Lista con los eventos en esa provincia	Satisfactorio
GetEventosXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetLocacionesEventosXprovincia	Válida	Lista con los locales para eventos de esa provincia	Lista con los locales para eventos de esa provincia	Satisfactorio
GetLocacionesEventosXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Tabla C.5. Servicio Excursion

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetExcursionesXprovincia	Válida	Lista con las excursiones en esa provincia	Lista con las excursiones en esa provincia	Satisfactorio
GetExcursionesXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Tabla C.6. Servicio Hospedaje

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetHospedajeXprovincia	Válida	Lista con las agencias de hospedajes de esa provincia	Lista con las agencias de hospedajes de esa provincia	Satisfactorio
GetHospedajeXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Tabla C.7. Servicio Informacion

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetInformacionXprovincia	Válida	Lista con los centros de información de esa provincia	Lista con los centros de información de esa provincia	Satisfactorio
GetInformacionXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Tabla C.8. Servicio Museo_Monumento

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetMuseosMonumentosXprovincia	Válida	Lista con los museos y monumentos de esa provincia	Lista con los museos y monumentos de esa provincia	Satisfactorio
GetMuseosMonumentosXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Tabla C.9. Servicio Naturaleza

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetAreasProtegidasXprovincia	Válida	Lista con las áreas protegidas en esa provincia	Lista con las áreas protegidas en esa provincia	Satisfactorio
GetAreasProtegidasXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetCazaXprovincia	Válida	Lista con las agencias de caza de esa provincia	Lista con las agencias de caza de esa provincia	Satisfactorio
GetCazaXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Tabla C.10. Servicio Nautica

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetBuceoXprovincia	Válida	Lista con las agencias de buceo de esa provincia	Lista con las agencias de buceo de esa provincia	Satisfactorio
GetBuceoXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetMarinasXprovincia	Válida	Lista con las marinas de esa provincia	Lista con las marinas de esa provincia	Satisfactorio
GetMarinasXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Tabla C.11. Servicio Patrimonio

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetPatrimoniosXprovincia	Válida	Lista con los patrimonios de esa provincia	Lista con los patrimonios de esa provincia	Satisfactorio
GetPatrimoniosXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Tabla C.12. Servicio Playa

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetPlayasXprovincia	Válida	Lista con las playas de esa provincia	Lista con las playas de esa provincia	Satisfactorio
GetPlayasXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Tabla C.13. Servicio Religion

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetInstitucionesReligiosasXprovincia	Válida	Lista con las instituciones religiosas de esa provincia	Lista con las instituciones religiosas de esa provincia	Satisfactorio
GetInstitucionesReligiosasXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Tabla C.14. Servicio Salud

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetHospitalesXprovincia	Válida	Lista con los hospitales de esa provincia	Lista con los hospitales de esa provincia	Satisfactorio
GetHospitalesXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetFarmaciasXprovincia	Válida	Lista con las farmacias de esa provincia	Lista con las farmacias de esa provincia	Satisfactorio
GetFarmaciasXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetOpticasXprovincia	Válida	Lista con las ópticas de esa provincia	Lista con las ópticas de esa provincia	Satisfactorio
GetOpticasXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Tabla C.15. Servicio Tienda

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetTiendaXprovincia	Válida	Lista con las tiendas de esa provincia	Lista con las tiendas de esa provincia	Satisfactorio
GetTiendaXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Tabla C.16. Servicio Transportacion

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetAerolineasXprovincia	Válida	Lista con las agencias de aerolíneas de esa provincia	Lista con las agencias de aerolíneas de esa provincia	Satisfactorio
GetAerolineasXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio

Continúa en la página siguiente

Tabla C.16. continuación de la página anterior

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetAeropuertosXprovincia	Válida	Lista con los aeropuertos de esa provincia	Lista con los aeropuertos de esa provincia	Satisfactorio
GetAeropuertosXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetBicicletasXprovincia	Válida	Lista con los lugares donde se alquilan bicicletas en esa provincia	Lista con los lugares donde se alquilan bicicletas en esa provincia	Satisfactorio
GetBicicletasXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetCrucerosXprovincia	Válida	Lista con las terminales de cruceros de esa provincia	Lista con las terminales de cruceros de esa provincia	Satisfactorio
GetCrucerosXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetFerrocarrilesItinerariosXprovincia	Válida	Lista con los itinerarios de los trenes de esa provincia	Lista con los itinerarios de los trenes de esa provincia	Satisfactorio
GetFerrocarrilesItinerariosXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetFerrocarrilesTerminalesXprovincia	Válida	Lista con las terminales de ferrocarriles de esa provincia	Lista con las terminales de ferrocarriles de esa provincia	Satisfactorio
GetFerrocarrilesTerminalesXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetGasolinerasXprovincia	Válida	Lista con las gasolineras de esa provincia	Lista con las gasolineras de esa provincia	Satisfactorio
GetGasolinerasXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetOmnibusNacItinerariosXprovincia	Válida	Lista con los itinerarios de los omnibus nacionales de esa provincia	Lista con los itinerarios de los omnibus nacionales de esa provincia	Satisfactorio
GetOmnibusNacItinerariosXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetOmnibusUrbItinerariosXprovincia	Válida	Lista con los itinerarios de los omnibus urbanos en esa provincia	Lista con los itinerarios de los omnibus urbanos en esa provincia	Satisfactorio
GetOmnibusUrbItinerariosXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
Continúa en la página siguiente				

Tabla C.16. continuación de la página anterior

Operación	Entrada	Respuesta Esperada	Respuesta Devuelta	Resultado Final
GetOmnibusTerminalesXprovincia	Válida	Lista con las terminales de omnibus de esa provincia	Lista con las terminales de omnibus de esa provincia	Satisfactorio
GetOmnibusTerminalesXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio
GetTaxisXprovincia	Válida	Lista con las agencias de taxis de esa provincia	Lista con las agencias de taxis de esa provincia	Satisfactorio
GetTaxisXprovincia	No válida	Un mensaje de error	Mensaje de Error	Satisfactorio