

Integración de Procesos de Negocio aplicando Servicios Web

Un Modelo para el BPI en el dominio de las PyMEs

Picón Darío¹

Fontana Fernando¹

Adriana Martín²

¹ Alumno Ingeniería en Sistemas
Dpto. de Cs. Exactas y Naturales

² Docente/Investigador Cátedra Desarrollo de Aplicaciones Web
Dpto. de Cs. Exactas y Naturales

Universidad Nacional de la Patagonia Austral, Unidad Académica Caleta Olivia (UNPA-UACO)
Ruta 3. Acceso Norte. Caleta Olivia. Santa Cruz. Argentina
e-mails: {dariounpa//fernandoeffontana//adrianaelba.martin}@gmail.com

Abril 2014

Resumen

La Integración de Procesos de Negocio (*Business Process Integration* – BPI), utiliza una gran cantidad de servicios distribuidos, por lo que la comunicación entre estos servicios es clave para el buen funcionamiento del sistema. En este modelo de cooperación, la arquitectura Cliente-Servidor tradicional ya no es suficiente para la implementación de soluciones que soporten comunicación entre aplicaciones distribuidas, independientemente de la plataforma y del lenguaje de programación que utilizan estas aplicaciones. Por su parte, las Arquitecturas Orientadas a Servicios (*Service Oriented Architectures* - SOA) proveen una estructura que posibilita el modelado de procesos y conexiones interorganizacionales. Mientras que la Gestión de Procesos de Negocio (*Business Process Management* - BPM) es el conjunto de sistemas de software, herramientas y metodologías para gestionar tales requerimientos y, el Lenguaje de Ejecución de Procesos de Negocio (*Business Process Execution Language* - BPEL), es un lenguaje de orquestación de servicios que permite definir la forma en que cooperan entre sí los Servicios Web para alcanzar la lógica de negocio. En este escenario, existen buenas herramientas para asistir desde lo conceptual y desde lo práctico a la Integración de Procesos de Negocio aplicando Servicios Web. Entonces, la problemática se plantea al momento de vincular estas herramientas de manera apropiada para facilitar el proceso de definición e implementación de este tipo de sistemas y en particular, en el ámbito de las pequeñas y medianas empresas (PyMEs).

En este trabajo se propone un modelo que hace posible el BPI mediante Servicios Web de una manera ágil y práctica, facilitando su implementación. El mismo será aplicado metodológicamente a un Caso de Estudio en el dominio de las PyMEs, con el propósito de evaluar su eficiencia, evidenciar sus beneficios y hallar oportunidades de mejora.

Palabras clave

BPI, Servicios Web, SOA, BPMN, BPEL, BPM.

Introducción

En la actualidad, las organizaciones están orientadas a procesos de negocios que atraviesan distintas unidades funcionales. Usualmente, cuentan con varias aplicaciones que dan soporte, de manera puntual a sus diferentes áreas. Es decir, cada una de estas aplicaciones de software fue implementada para asistir a las operaciones de una determinada área (o división) de la organización, pero sin tener en cuenta la relación entre los procesos. Además, hay que considerar que estas aplicaciones están desarrolladas en diferentes lenguajes, que usan diferentes tecnologías y, se despliegan en diferentes plataformas de hardware y sistemas operativos con interfaces de usuario inconsistentes y de baja calidad. Como consecuencia, el resultado es una funcionalidad aislada, múltiples instancias de los mismos datos, actividades manuales redundantes, respuestas ineficientes para los clientes y además, costos más altos.

La Integración de Procesos de Negocio (BPI) y los Servicios Web se han convertido en los últimos años en términos muy utilizados en el ámbito de las empresas, dado que estas no sólo requieren integrar sus procesos de negocio sino también interactuar en tiempo real con sus socios, proveedores y clientes a través de Internet (1). Dichas aplicaciones hacen uso de una gran cantidad de servicios distribuidos, por lo que la comunicación entre éstos es clave para el buen funcionamiento del negocio. Por estos motivos, ha surgido la necesidad de pensar en nuevas arquitecturas de recursos informáticos para la implementación de este tipo de soluciones que le permitan a las organizaciones adaptarse rápidamente a los cambios que se producen en su entorno, mediante el desarrollo de procesos de negocio multifuncionales, facilitando la comunicación entre las aplicaciones de software y de acuerdo a la lógica de negocio prescripta por los procesos (2).

El presente trabajo tiene como objetivo realizar primero una revisión de la literatura para determinar el estado del arte en enfoques, tecnologías y herramientas que den soporte al BPI. Seguidamente, proponer un modelo en el que se aborde paso a paso el proceso de BPI de una manera sencilla y transparente para los desarrolladores, en conformidad con arquitecturas distribuidas y orientadas a servicios, que soporten estas diferentes tecnologías. Por último, desarrollar un Caso de Estudio en el dominio de las PyMEs que ponga a prueba y refuerce el modelo presentado.

El resto del documento se organiza de la siguiente manera: en la Sección 1, se analiza la necesidad actual de las PyMEs en torno al BPI y una revisión del estado del arte en BPI y BPM. En la Sección 2, se describe la arquitectura SOA basada en Servicios Web y los estándares involucrados para integrar aplicaciones distribuidas. En la Sección 3, se presenta un modelo de guía que provee las bases conceptuales, métodos y herramientas que se necesitan para alcanzar el BPI y obtener un producto completamente modular y funcional. Mientras que en la Sección 4, se presenta un caso práctico de implementación en el marco de una PyME aplicando el modelo propuesto. En la Sección 5, se muestran los resultados obtenidos en el Caso de Estudio. Finalmente, en la Sección 6, se presentan las conclusiones, las nuevas tendencias en materia de BPI (y las tecnologías y estándares de soporte), las cuáles vamos a considerar en trabajo futuro.

Trabajos Relacionados

Si bien existen numerosos trabajos relacionados con la problemática de aplicar el BPI en una organización, hemos decidido citar a los siguientes porque han contribuido con diferentes enfoques que promueven abordar el proceso de integración de procesos de negocio de una manera sistemática y organizada, lo que nos ha permitido reforzar el modelo de integración propuesto.

Reyes, en (3) fusiona las técnicas “Business Process Management: Rapid Analysis&Design” (BPM: RAD) y “Polymita”, resultando una metodología concreta y práctica para la implementación de proyectos BPM.

Delgado en (4) desarrolla una metodología implementando un sistema siguiendo un enfoque Process-Enabled SOA.

Bazán en (5) luego de un análisis comparativo, supera la falencia de los métodos tradicionales en lo que respecta a la integración, tendiendo hacia el paradigma orientado a procesos. Delinea una nueva visión global que identifica las etapas y su interacción, para cubrir tanto el ciclo de vida de los procesos de negocio como el del software de una manera unificada. Propone una metodología con el objetivo de facilitar la construcción de procesos de mejora continua y flexibilidad ante los cambios. Presenta los enfoques SOA y BPM como propuestas complementarias entre sí que sustentan el cambio de punto de vista en la manera de entender las organizaciones.

Por otro lado, se observó que las publicaciones que difunden los productos comerciales más populares, se hallan frecuentemente actualizadas y lo hacen sesgados hacia aspectos técnicos, sin mayores aplicaciones a un marco metodológico específico.

Mientras que estos aportes son dirigidos a empresas de gran envergadura y corporaciones, la propuesta metodológica descrita en el presente pretende alcanzar aplicabilidad en PyMEs, organizaciones que por lo general no trascienden barreras culturales. Este contexto reduce complejidades tecnológicas, principalmente en lo que concierne a seguridad y performance.

1. Estado del arte en BPI

BPI nació como la evolución de EAI (*Enterprise Application Integration*) (6). La primera generación de EAI se enfocó en la conversión e intercambio de datos entre aplicaciones. Luego, se añadieron conexiones punto a punto entre aplicaciones de manera que puedan compartir información dentro y fuera de la organización. Finalmente, al comprender la importancia de los procesos de negocio en el cumplimiento de los objetivos organizacionales, se buscó una manera de integrar las aplicaciones en un nivel más alto de abstracción, dando surgimiento al BPI. Éste modelo permite el intercambio de información entre aplicaciones como parte de un proceso de negocio, que es controlado a través de BPM.

1.1 Escenario actual

Las organizaciones constan de departamentos (áreas o divisiones), especializados en cumplir ciertas funciones, como por ejemplo contabilidad, marketing, producción, etc. (7). La forma más sencilla de representar gráficamente la estructura funcional de una organización es a través de un organigrama. Pero, existe otra perspectiva, desde la cual una organización puede ser vista como un conjunto interrelacionado de procesos. En este caso, a través de un mapa de procesos se puede representar gráficamente todos los procesos que existen en una organización y la relación que hay entre ellos.

Los procesos de negocio son un conjunto de actividades lógicamente relacionadas que llevadas a cabo permiten alcanzar un determinado objetivo. Dichos procesos de negocio pueden atravesar distintos departamentos (áreas o divisiones) de la organización y potencialmente a distintas organizaciones. Bajo este esquema organizacional, el alcance de un proceso de negocio puede limitarse a un determinado departamento de una empresa, puede abarcar (atravesar) varios departamentos o requerir la colaboración de otras empresas (8). Por ejemplo, el procesamiento de un pedido de cliente normalmente requiere cruzar las fronteras de varios departamentos: (i) ventas, para tomar la orden del pedido, (ii) planificación, para planificar la fabricación del producto (o la reposición del inventario), (iii) producción, para

fabricarlo (de ser necesario), (iv) transporte, para despacharlo y entregarlo y, (v) contabilidad, para procesar la venta y su respectivo cobro (7).

En (9) (7), se hace una distinción entre los dos posibles tipos de procesos de negocio:

1. Procesos ejecutables. Los cuales describen los procesos empresariales reales que son internos a la organización y que se especifican por completo.
2. Procesos abstractos. Los cuales son partes de un proceso de negocio de una empresa que están expuestas a procesos externos en el contexto de las interacciones interempresariales. Un proceso abstracto está especificado parcialmente y no es ejecutable.

Debido al valor que representa la información en las grandes organizaciones y las divisiones individuales dentro de ellas, han construido (o adquirido) aplicaciones para satisfacer las demandas del negocio. Sin embargo, muchos de estos sistemas se hicieron para satisfacer requerimientos particulares basados en problemas operacionales específicos de un área funcional, sin tomar en cuenta la relación entre los procesos (10). Aplicando esta conducta, y a medida que las organizaciones han ido definiendo, desarrollando e implementando sistemas heterogéneos, desarticulados e independientes entre sí, y se volvió más evidente la necesidad de buscar medios de comunicación entre estos sistemas. En la actualidad, la dinámica del mundo de los negocios, demanda un completo acceso a la información para poder alinearla con personas y procesos de negocios a través de distintas tecnologías y organizaciones. Es por ello, que para satisfacer estas demanda surge un interés creciente de integrar a las organizaciones como un todo, creándose una variedad de metodologías (o disciplinas), así como herramientas más potentes para concretar esta integración (10).

1.2 Desafíos en BPI

Actualmente un gran número de PyMEs se encuentran operando con entornos heterogéneos compuestos por plataformas y sistemas dispares. En este contexto, la conexión de estas aplicaciones y su vinculación con los procesos de negocio es un enorme desafío para la mayoría de las organizaciones, debido a que:

- Las aplicaciones y sistemas clave para las operaciones del negocio no viven aisladas, sino que se interrelacionan con otras aplicaciones y sistemas, tanto de la propia organización, como externos a la misma. Pero a menudo son islas desconectadas entre sí, y muchos procesos de negocio que se apoyan en estos sistemas, acaban siendo ineficientes y poco gestionados.
- Muchos procesos de negocio necesitan gestionar y coordinar transacciones de larga duración, tales como pedidos y órdenes de servicio, en las que participan diversos departamentos y sistemas de información de la propia organización, así como también, entidades externas prestadoras de servicios.

La **Figura 1** ilustra esta situación describiendo un proceso de negocio que atraviesa diferentes departamentos de una organización, cada uno operando con diferentes sistemas de información y aislados unos de otros. En este caso, la organización funcional responde a la clásica teoría de administración, centrada en cada función de negocio por separado y trabajando de forma aislada.

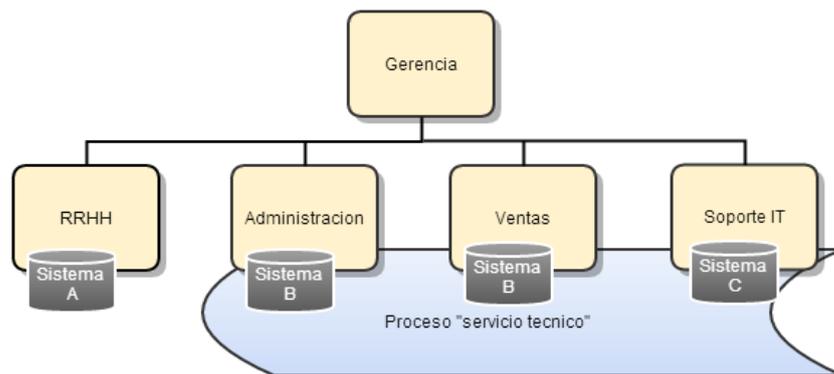


Figura 1. Relación entre unidades funcionales y procesos de negocio

Este enfoque ha llevado a las organizaciones a pensar en una nueva filosofía que considere al proceso de negocio completo, de punta a punta, buscando optimizar e integrar las funciones y procesos de negocio. En la práctica esta tendencia condujo a estructuras organizativas orientadas a proyectos, mediante las cuales los equipos de proyecto se ensamblan a partir de diferentes unidades funcionales para manejar todos los aspectos de una transacción comercial; o a organizaciones matriciales, que proveen un enfoque múltiple tanto en función como en proyecto (7).

1.3 Automatización de los procesos de negocio

Como fue mencionado anteriormente, las organizaciones poseen un conjunto de aplicaciones empresariales. Estas aplicaciones están orientadas a apoyar funciones específicas de un departamento (área o división) de la organización. Un ejemplo típico es la aplicación para el área financiera, otra para el área de recursos humanos y así en las otras áreas de la organización (11).

Un proceso puede ser realizado por una sola persona, o dentro de un mismo departamento. Sin embargo, los más complejos fluyen en la organización a través de diferentes departamentos (10). Es así que un proceso puede atravesar varios departamentos de una organización, y esto implica atravesar por cada una de las aplicaciones de software con la que opera cada uno de estos departamentos. Los procesos de negocio pueden ser controlados y administrados por un sistema basado en software. Los procesos de negocio automatizados de esta manera se denominan *Workflow*. Esta automatización resulta en una importante potenciación de los beneficios de dichos procesos, obteniéndose mejoras en cuanto a rendimiento, eficiencia y productividad de la organización (12). Los *Workflows* permiten la ejecución controlada y coordinada de un conjunto de actividades de forma semi-automática (o automática) en un motor de procesos, también denominado servidor de procesos. En las actividades del proceso existe la posibilidad del intercambio de información, ya sea dentro del proceso o hacia el exterior. En estos motores, los participantes de un proceso pueden ser usuarios finales como así también aplicaciones. Los motores ejecutan procesos definidos en un lenguaje particular (por ejemplo BPMN), que permite definir el flujo de información, la secuencia de actividades y sus condiciones de ejecución (11).

Integrar los procesos de negocio de las organizaciones con aplicaciones empresariales partiendo de un modelo de alto nivel, requiere la participación de elementos tecnológicos y de negocio. Desde el punto de vista tecnológico, significa llevar el proceso a un motor de ejecución de *Workflow*, comunicarlo con aplicaciones de diferentes tecnologías y ubicar instancias del proceso. En la perspectiva de negocio, implica establecer en qué actividades y de qué forma participa una aplicación y mantener consistencia en la información tanto del

proceso como de las aplicaciones (11). Es claro que la clave para el éxito de este proceso, es lograr una real y efectiva integración de todos los recursos y procesos de la organización, una tarea nada sencilla de concretar ante las viejas prácticas organizacionales que propician el aislamiento y la apropiación por parte de los departamentos de sus recursos y sistemas.

1.4 Estándares de implementación de BPM

BPM (*Business Process Management*) es el nombre de un conjunto de sistemas de software, herramientas y metodologías enfocadas hacia la manera en que las organizaciones identifican, modelizan, desarrollan, distribuyen y gestionan tales procesos de negocios. Mientras que los procesos de negocio requieren de la representación de sus interacciones, los servicios Web por sí mismos, no soportan la posibilidad de modelar los procesos y sus conexiones interorganizacionales. Esto trae la necesidad de orquestar servicios Web, es decir, establecer patrones para gestionar las interacciones entre ellos. Entonces, a los efectos de implementar la interacción entre varios servicios Web para lograr un objetivo de negocio, los desarrolladores pueden utilizar WS-BPEL (*Web Service Business Process Execution Language*). Este lenguaje permite establecer la forma en que van a cooperar los diferentes Servicios Web para realizar un proceso de negocio (9) y así centralizar el control de la invocación de éstos de acuerdo con la lógica del proceso, definiéndose cuál, cómo y cuándo se ejecutará un Servicio Web determinado.

De esta manera, se obtiene una visión global del sistema distribuido y se manejan las interacciones entre los servicios individuales para integrar los Servicios Web, sin preocuparse por las cuestiones propias a la implementación de los mismos. El hecho de que WS-BPEL esté basado en XML, permite que la programación pueda realizarse de manera tradicional, pero en el uso cotidiano, se programa de forma gráfica, para facilitar a los responsables de negocio definir y analizar los procesos de negocio implementados. La notación más utilizada para realizar esta tarea es BPMN (*Business Process Management Notation*), la cual fue concebida para ser entendida de manera rápida por todas las personas involucradas en los procesos de negocio, desde los gerentes (sin conocimiento técnico) hasta los programadores. BPMN permite ver y organizar los pasos que se ejecutan de manera coordinada por los participantes de un proceso para obtener un objetivo de negocio. Estas herramientas permiten modelar y gestionar eficientemente los Procesos de Negocio, por ello, el uso conjunto de las mismas es considerada una cadena de valor conocida como: "The BPMN-XPDL-BPEL value chain", donde:

BPMN (*Business Process Management Notation*), es una notación gráfica estandarizada que permite el modelado de procesos de negocio, en un formato de flujo de trabajo (Workflow). BPMN fue inicialmente desarrollada por la organización Business Process Management Initiative (BPMI), y es actualmente mantenida por el OMG (*Object Management Group*) (13). Se trata de una notación gráfica que describe los pasos y actividades de un proceso de negocio. Modela tanto la secuencia de actividades como los datos (o mensajes) intercambiados entre los distintos participantes. Los procesos modelados a través de BPMN son almacenados en formato XPDL, siendo esta una manera de traducir el formato gráfico (para ser leído por personas) a un formato legible por máquinas y permitir el intercambio entre distintas herramientas.

XPDL (*XML Process Definition Language*), es un formato de archivo basado en XML que representa el "dibujo" de la definición del proceso. Guarda el tamaño y las coordenadas X e Y del nodo. Además, los nodos pueden especificar atributos tales como roles, descripción de actividades, timers, llamadas a Servicios Web, entre otros. XPDL ofrece una manera estándar para representar procesos de negocio, de tal manera que puedan ser importados/exportados por cualquier editor BPMN que implemente el estándar, favoreciendo la portabilidad entre diagramas diseñados por herramientas de distintos proveedores. En (14), se presenta uno de los primeros trabajos en los que se trata la relación entre BPMN y XPDL y se establece un

mapeo (correspondencia) entre ellos.

WS-BPEL (*Web Service Business Process Execution Language*), es un lenguaje ejecutable con sus variables y operaciones. Las operaciones permiten enviar y recibir mensajes SOAP (*Simple Object Access Protocol*) y tiene un gran soporte para XML y transformaciones XML. Además, permite de manera fácil invocar múltiples Servicios Web al mismo tiempo y sincronizar los resultados. BPEL no tiene soporte gráfico, es decir, no especifica cómo deben ser los diagramas que especifican los procesos de negocio. El objetivo de BPEL es ofrecer una forma de orquestar Servicios Web, la secuencia de interacciones subyacente y el flujo de datos punto a punto.

El sustento de esta cadena de valor radica en el hecho de que a partir del diseño de un proceso de negocio con BPMN utilizando una herramienta en conformidad con el estándar XPDL es posible finalmente traducirlo a BPEL para ser ejecutado en un motor BPEL de cualquier fabricante. En (15) se pone énfasis en resaltar la diferencia entre XPDL y BPMN señalando que son cosas totalmente diferentes para propósitos completamente diferentes. XPDL es usado para llevar un diseño BPMN de una herramienta de diseño a otra con total transparencia, en cambio BPEL se usa para comunicar el proceso ejecutable al motor, siendo generalmente la herramienta de diseño la encargada de traducir el diseño a BPEL. XPDL puede proporcionar este intercambio de diseño, ya que mantiene una correspondencia uno a uno con el diagrama de proceso BPMN original, y de hecho no es requerida una “conversión”. Un diagrama BPMN se puede almacenar en XPDL y a partir del XPDL recuperar el diagrama original, sin ningún tipo de pérdida. Por otro lado, BPMN también provee una correspondencia formal a lenguajes de ejecución como BPEL, sin embargo este mapeo no es trivial y es de un solo sentido: se puede tomar un diagrama BPMN y producir un BPEL, pero es muy difícil (o prácticamente imposible) recuperar un diagrama BPMN a partir de su BPEL (15).

2. La arquitectura SOA y los Servicios Web

La Arquitectura Orientada a Servicios (SOA) es un paradigma para la creación y uso de procesos de negocio. Los procesos de negocio se encuentran encapsulados dentro de componentes levemente acoplados, autónomos y abstractos, llamados servicios (16). Estos servicios son ofrecidos y publicados por el proveedor del servicio que adquiere los procesos de negocio (17). Por su parte, el proveedor usa WSDL (*Web Services Description Language*) para definir las interfaces externas de los Servicios Web ofrecidos. Con WSDL se expresa la localización de los Servicios Web, las descripciones de las operaciones que realiza cada uno (y como invocarlas) y las funciones de negocio que cumple cada Servicio Web. En general, estas descripciones son publicadas en los registros de Servicios Web, para que los clientes puedan acceder a ellas y hacer uso de los mismos. Los clientes utilizan los registros de Servicios Web para encontrar un servicio adecuado, es decir, uno que se ajuste a los requerimientos del cliente (8). Por otra parte, los registros de Servicios Web son opcionales y se implementan utilizando el estándar UDDI (*Universal Description, Discovery, and Integration*). Así, los clientes y proveedores se comunican con el protocolo de mensajes SOAP, escrito en XML y utilizado por los clientes para enviar solicitudes y por los proveedores para enviar las respectivas respuestas a éstos. Este protocolo es utilizado para intercambiar mensajes y realizar llamadas a procesos remotos (*Remote Procedure Call - RPC*) en un ambiente distribuido de la forma más transparente y sencilla posible.

De acuerdo a lo anteriormente mencionado, en la arquitectura SOA nos encontramos con tres roles fundamentales y conformando la estructura básica SOA tal como ilustra la **Figura 2**:

- El proveedor del Servicio Web (Service Provider) que utiliza WSDL.
- El cliente del Servicio Web (Service Requester).

- El registro de Servicios Web (Service Broker - UDDI).

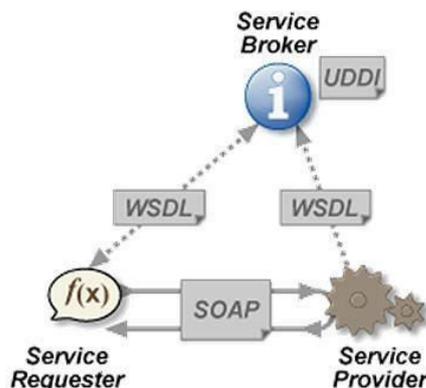


Figura 2. Estructura Básica de SOA

Siguiendo los principios de SOA, los Servicios Web proporcionan mecanismos que permiten a las aplicaciones comunicarse entre sí, independientemente de sus lenguajes y plataformas de programación. La manera de hacerlo es empaquetando las aplicaciones de software en servicios reusables con interfaces estándares e invocables, que puedan ser llamadas en secuencias definidas para formar procesos de negocios. Así, a través de mecanismos estandarizados para apoyar la comunicación entre los servicios y la capacidad de combinar servicios para soportar funcionalidades complejas, los Servicios Web logran superar los típicos problemas de integración de sistemas (2). La invocación de servicios a través del estándar SOAP facilita el intercambio de mensajes entre los servicios. Sin embargo, SOAP no apoya la coordinación de las actividades para gestionar de manera efectiva los procesos de negocio, que pueden incluir interacciones complejas, las cuales a su vez requieren la ejecución de operaciones específicas de las distintas aplicaciones de una manera sistemática. Estas interacciones complejas pueden incluir una secuencia de invocaciones a servicios que será especificada utilizando WS-BPEL (18). En las Secciones 2.1 y 2.2 se describen los componentes de la estructura SOA: WSDL y SOAP.

2.1 WSDL

En una Arquitectura Orientada a Servicios, SOAP ofrece la comunicación básica, pero no nos dice que mensajes deben enviarse para poder interactuar con éxito con un servicio. Esta es la función de WSDL (19). El Lenguaje de Descripción de Servicios Web (WSDL - *Web Services Description Language*) es un lenguaje basado en XML para describir la interfaz pública de los Servicios Web; eso significa que detalla los protocolos y los formatos de los mensajes necesarios para interactuar con estos servicios. WSDL 2.0 permite separar la descripción de la funcionalidad abstracta que ofrece un servicio, de los detalles concretos de una descripción del servicio, tales como el "cómo" y "dónde" se ofrece esta funcionalidad (20).

2.1.1 Estructura del WSDL

WSDL se divide en dos partes claramente diferenciadas: una abstracta y una concreta con el objetivo de promover la reutilización de la descripción y separar ciertas cuestiones de diseño.

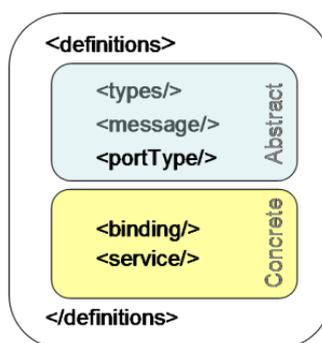


Figura 3. Estructura del WSDL

- La parte abstracta está formada por los elementos <types>, <message> y <portType> (o <interface> en 2.0), y define el servicio en función de los mensajes que envía y recibe. Los mensajes son descritos independientemente del formato de transporte y se definen los tipos de datos, usando por lo general XML Schema (20).
- La parte concreta, le corresponden a los elementos <binding> y <service> y define qué protocolo de comunicación usar (por ejemplo SOAP sobre HTTP), cómo llevar a cabo la interacción entre servicios y dónde terminar la comunicación (la dirección de red) (19).

En la **Tabla 1** se describen los elementos que forman parte de la estructura del WSDL.

Tabla 1. Tabla de elementos del WSDL

DESCRIPCIÓN	ELEMENTO WSDL
Un documento WSDL es como cualquier documento XML y se basa en los esquemas, por lo que debe comenzar con dicha etiqueta.	<?xml version="1.0">
Comienzo del documento, este tag agrupa a todos los demás elementos.	<definitions>
Se definen los tipos de datos utilizados en los mensajes. Se utilizan los tipos definidos en la especificación de esquemas XML.	<types>
Se definen los métodos y parámetros para realizar la operación. Cada message puede consistir en una o más partes (parámetros). Las partes pueden ser de cualquiera de los tipos definidos en la sección anterior.	<message>
Esta sección es la más importante, ya que definen las operaciones que pueden ser realizadas, y los mensajes que involucran (por ejemplo el mensaje de petición y el de respuesta).	<portType>
Se definen el formato del mensaje y detalles del protocolo para cada portType.	<binding>
En el tag <service> podemos tener los siguientes atributos e hijos: <ul style="list-style-type: none"> • name: indica el nombre del servicio web. • <documentation>: Es opcional. En ella se puede incluir una descripción del servicio. • <port>: Aquí se indica la dirección y el tipo de acceso de los servicios web. 	<service>

Utilizando estos elementos no es posible conocer qué hace un servicio, pero sí disponer

de la información necesaria para interactuar con él (funciones, mensajes de entrada/salida, protocolos, etc).

2.2 SOAP

El Protocolo Simple de Acceso a Objetos (*Simple Object Access Protocol* - SOAP) (21) define un mecanismo para la comunicación con los Servicios Web a través de Internet. Especifica el formato de los mensajes que se intercambian entre el Service Requester, el Service Provider, y el servicio de registro de servicios.

XML-RPC¹ es un protocolo basado en llamada a procedimiento remoto (*Remote Procedure Call* - RPC) que utiliza XML para codificar sus llamadas y HTTP como mecanismo de transporte. SOAP deriva de XML-RPC (22). Surge junto con WSDL (23) para dar respuesta a la necesidad de los Servicios Web en el intercambio de información estructurada (XML) por medio de un mecanismo simple, consistente y extensible. Mientras SOAP se centra en la codificación de los mensajes y la definición de la función de llamadas remotas en XML, WSDL proporciona un mecanismo para describir un conjunto de llamadas a operaciones remotas como un puerto que puede ser abordado mediante un mensaje. SOAP y WSDL conjuntamente permiten la creación de contenedores de Servicios Web entre aplicaciones a través de un lenguaje de definición de interfaz estandarizada. SOAP especifica el protocolo para invocar remotamente métodos en servidores, servicios, componentes y objetos (RPC). Para formatear el mensaje usa el XML Information Set y para la negociación y transmisión de los mismos usa la capa de aplicación del protocolo de transferencia de hipertexto (HTTP) (8) o Simple Mail Transfer Protocol (SMTP). Establece una conexión entre emisor y receptor tratando los mensajes XML como solicitudes de servicios remotos. El modelo SOAP separa el procesamiento de la infraestructura y el procesamiento de la aplicación de mensajes. La **Figura 4** muestra la estructura básica de un mensaje SOAP.

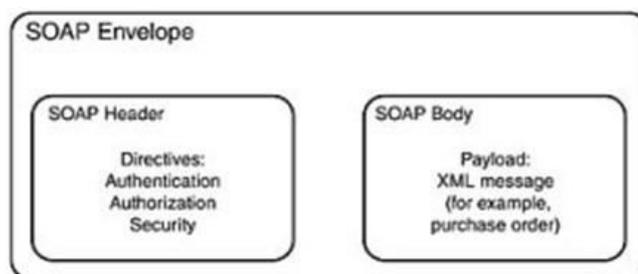


Figura 4. Estructura básica de un mensaje SOAP

Un mensaje SOAP tiene dos partes:

- Encabezado (SOAP header): metadatos para administrar y/o asegurar el paquete tales como tipo de contenido, identidad de emisor, etc.
- Cuerpo (SOAP body): carga útil del mensaje en XML bajo dos "estilos" posibles (24):
 - RPC-style: tienden a ser simples parámetros para facilitar llamar a un método remoto.
 - Document-style: tendiendo a ser documentos XML ricos; recomendados para WS B2B (*Business to Business*).

Ciertas cabeceras (25) previamente especificadas por HTTP permiten a los firewalls y

¹RPC con uso de XML para llamada a procedimiento remoto.

proxys filtrar los mensajes SOAP autorizados.

Se han desarrollado tecnologías complementarias a SOAP tales como MTOM (*Message Transmission Optimization Mechanism*) y SOAP-RRSHB (*Resource Representation SOAP Header Block*) (26), para optimizar el rendimiento en el envío de los mensajes en formato binario XOP (*XML-binary Optimized Packaging*) (27) y los recursos que se transmiten en esos mensajes.

Las posibles interacciones en el contexto de SOAP y WSDL, se esquematizan en la **Figura 5** pudiendo ser:

- La operación unidireccional (one-way) WSDL estándar es soportada como escenario útil en comunicaciones asincrónicas donde el cliente envía la petición sin esperar la respuesta del servidor que la ejecuta y sin devolver datos ni confirmación de éxito de su ejecución.
- La operación solicitud-respuesta (request-response) WSDL estándar es más comúnmente utilizada. Se envía una solicitud, se ejecuta la operación en el servidor y se devuelve la respuesta. Mientras se está cursando la ejecución se mantiene la conexión establecida por el cliente en espera de la respuesta. Ante el agotamiento del tiempo de espera de HTTP, se resetea el socket con algún error.

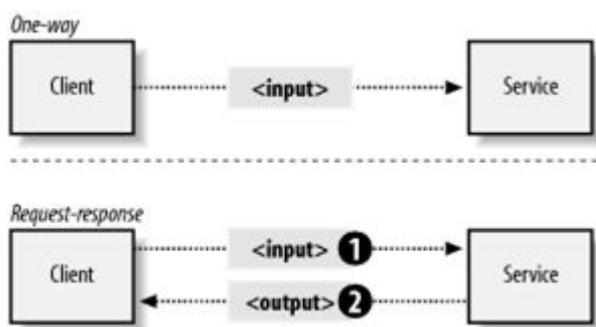


Figura 5. Interacción peer-to-peer usando SOAP y WSDL

Otras operaciones tales como la de solicitud-respuesta y notificación son soportadas por implementaciones especializadas WSDL (o extensiones del elemento de enlace WSDL). Para poder manejar estas comunicaciones, las notificaciones e interacciones de solicitud-respuesta deben asociarse con un ID de correlación.

Por último, cabe señalar que SOAP y WSDL no proporcionan mecanismos para la secuenciación de los mensajes más allá de pares de mensajes individuales (por ejemplo, solicitud-respuesta). Un proveedor de servicio puede decidir en qué nivel de granularidad desea exponer sus operaciones internas.

3. Modelo para la implementación de BPI en una PyME

En esta sección se propone un modelo para abordar la integración de procesos de negocio en el ámbito de las PyMEs con una visión integradora de procesos y servicios. Es aplicable a toda organización donde se presenten claras posibilidades de automatización e integración de procedimientos.

El modelo pretende servir de guía al desarrollador y facilitar su tarea, disgregando el proceso de implementación de una solución BPI en 6 etapas diferenciadas que le permitan abordar la complejidad de una manera organizada. Además es independiente del BPMS (*Business Process Management Suite*) con el cual se automatizarán los procesos diseñados.

Al ser un modelo simplificado y versátil, omite las actividades iniciales en un proyecto tales como el análisis de oportunidad y retorno de inversión, la definición de objetivos y alcance del proyecto, y la especificación de requerimientos. Será el modelo de ciclo de vida adoptado para el proyecto quien sirva de base y de marco a estas actividades. Es decir, nuestro modelo describirá una serie de etapas que se aconsejan ejecutar de manera secuencial independientemente del ciclo de vida utilizado. Sin embargo, dada la naturaleza compleja y cambiante de los requerimientos del negocio en el contexto de las PyMEs actuales, es recomendable que se siga un enfoque (ciclo de vida) iterativo incremental como forma de lidiar con estos cambios, y con fuerte enfoque en la generación de productos intermedios como forma de ir obteniendo productos claves en el avance del mismo que provean visibilidad sobre el desarrollo.

Para implementar este modelo de integración de sistemas se recurre a una arquitectura SOA. Esta tecnología nos posibilita empaquetar las aplicaciones de software como servicios reutilizables para ser accesibles a través de interfaces estandarizadas por otras aplicaciones o mediante el uso de un navegador web (2).

A continuación se describen las 6 etapas que forman parte del modelo propuesto:

Etapas 1: Identificación y análisis de los procesos de negocios

El objetivo de esta etapa es comprender y describir el/los procesos de negocios afectados por la solución que se va a desarrollar.

Se realizan las siguientes actividades que hacen a una mayor comprensión del contexto organizacional donde se va aplicar la solución BPI:

- Identificación y análisis de los Procesos de Negocio involucrados.
- Análisis de los sistemas de software existentes.
- Diseño de una estrategia de implementación de la solución que sirva de “hoja de ruta” para los miembros del equipo de desarrollo durante las siguientes etapas del modelo.

No es necesario identificar todos los procesos de la organización y esquematizar un diagrama de procesos, solo se requieren los implicados en el desarrollo.

Puede ser de utilidad representar los procesos identificados como un Caso de Uso del negocio, que inicialmente será descrito en forma textual y luego modelarlo en UML con un diagrama de actividad.

Etapas 2: Implementación de los Servicios Web

La idea de un servicio es identificar un ámbito bien definido de funcionalidad y hacerlo accesible de forma uniforme y completa, independientemente de dónde se use (5). En el ámbito de una PyME se trata de identificar los sistemas de software existentes que proveen la funcionalidad e información que requiere el proceso de negocios y hacerlos disponibles en forma de Servicio Web para poder ser accedidos por dicho proceso.

Para cada servicio identificado se realiza el contrato funcional con WSDL, es decir, se define la interfase pública que establece la manera en que el Servicio Web se comunicará con el proceso de negocio u otros Servicios Web.

Así, quedan especificados para cada Servicio Web los métodos que deben ser implementados para proveer el servicio acordado para la interfase. Para cada operación en la interfase se debe especificar: a) nombre del método, b) parámetros requeridos por el método, con nombre, tipo y descripción y c) valor retornado, indicando el nombre, tipo y descripción entre otros.

Se deben implementar los componentes proveedores de servicios (Service Provider), e implementar el ligamiento de servicios en las Application Frontend (Service Requester) y en los servicios que llaman a otros servicios; esto último en caso de que se presenten configuraciones complejas en las que un Servicio Web invocado por un Proceso de Negocio invoque a su vez a otro Servicio Web.

Hay un asunto clave que no debe ser dejado de lado al brindar una solución de este tipo: la granularidad del servicio (2). Una solución que involucra la integración de aplicaciones por medio de Servicios Web debe identificar el nivel de granularidad es adecuado, es decir, decidir si exponer la aplicación completa como un servicio o solo un conjunto acotado de funciones dentro de la aplicación.

Esto es importante porque la decisión correcta puede maximizar la reusabilidad y permitir incorporar servicios individuales rápidamente para lograr la funcionalidad necesaria. Eso no quita que no se puedan integrar en un futuro, de hecho SOA es sumamente flexible en esto, pero considerarlo a tiempo reducirá tiempo y costos.

Etapa 3: Modelado del proceso de negocios

El objetivo de esta etapa es representar gráficamente los procesos utilizando la notación BPMN, es decir, modelar cada uno de los procesos identificados y detallados en los casos de uso del negocio mediante BPMN, obteniendo un diagrama del proceso de negocio.

Se toma como base la descripción inicial de los procesos realizada en la Etapa 1.

Lo fundamental es definir la Orquestación de Servicios, es decir, explicitar la secuencia de invocaciones a Servicios Web necesaria para la ejecución de los procesos de negocio. Para serán requeridos los WSDL de los Servicios Web producto de la etapa anterior.

Etapa 4: implementación de la interface de interacción con usuarios

El objetivo de esta etapa es implementar los formularios de entrada que permitan a los usuarios interactuar con el proceso de negocio.

Existen procesos de negocios automatizados en su totalidad y que no requieren la intervención humana, sin embargo en algunas situaciones es el usuario el disparador o iniciador del proceso de negocio y en otras tantas es requerida la intervención del usuario en una etapa intermedia del proceso para que éste pueda continuar.

Etapa 5: Despliegue

En esta etapa se lleva a cabo el despliegue, en la plataforma elegida, de los Procesos de Negocio y Servicios Web que han resultado de las etapas anteriores.

Los procesos de negocio, como se mencionó, se modelan en BPMN y es el BPMS quien se encarga de hacer la derivación automática a BPEL, haciendo al proceso ejecutable y permitiéndole desplegar en una maquina o motor de ejecución BPEL, generalmente denominado Process Server o Servidor de Procesos. Los Servicios Web, en cambio, se deben desplegar en servidores web HTTP.

La arquitectura de referencia utilizada para el modelo propuesto (SOA), no genera

ningún tipo de implicancia con la distribución física o topología de los Servicios Web y los procesos de negocio. Es posible desplegar todos los componentes en un único equipo físico o distribuirlos en varios, sin embargo lo habitual es ejecutar los Servicios Web en los mismos equipos donde se encuentran las aplicaciones de software de donde consumen información y los procesos de negocio en otro equipo que contenga el BPMS.

A pesar de esto último, no existen reglas estrictas para el despliegue y la decisión puede estar fundada en aspectos de seguridad, performance y disponibilidad, además de la infraestructura de hardware que posea la PyME.

Es siempre útil para futuras referencias del personal IT y ante un posible mantenimiento, conservar siempre actualizado un diagrama de despliegue que refleje la configuración vigente, es decir, que describa la forma en la que los componentes que forman parte de la solución fueron desplegados en equipos físicos.

Etapas 6: Administración y seguimiento

Esta etapa se vincula con las etapas de medida de rendimiento y administración del ciclo de vida de los procesos de negocio.

Se llevan a cabo actividades de seguimiento y control tales como el monitoreo en tiempo real del proceso, la evaluación del progreso, performance, status y estado de finalización, y la revisión del historial de ejecuciones del proceso, entre otras.

Los BMPS proveen las salidas pertinentes para poder realizar una adecuada lectura de indicadores.

Cabe aclarar que el modelo de BPI propuesto es aplicable dentro de los límites de una organización pero también se hace extensible a procesos que exceden las barreras organizacionales y se abocan a crear una relación transparente entre socios de negocio (como la interconexión en tiempo real entre cliente y proveedor). Cuando hablamos de “procesos que exceden las barreras organizacionales”, técnicamente se refiere a que la solución desarrollada (propiedad de la misma organización) pasa a ser ofrecida a sus socios de negocio como un servicio.

Todo esto es posible gracias a las ventajas y beneficios de Internet, junto con la tecnología de los Servicios Web que se despliegan en ella. El hecho de que SOAP no imponga un modelo de programación para definir la semántica de la aplicación, permite que sea utilizado en una gran variedad de soluciones.

4. Caso de estudio

Con el fin de validar el modelo propuesto se aplicó una solución de BPI en una PyME. Se propone como escenario de aplicación del modelo el de una PyME dedicada a la producción de productos (bienes o servicios) particulares, que opera dentro de la jurisdicción nacional.

Esta organización posee una única sucursal o casa matriz, con tres áreas funcionales que responden a la Dirección General: Área A: Planeamiento, Área B: Producción y Área C: Comercialización.

El área A opera con un sistema legado de escritorio desarrollado en Java, con un DBMS MySQL. El área B opera con un sistema web desarrollado en PHP, con un DBMS PostgreSQL. Tanto el sistema Java como el PHP operan de manera independiente para dar soporte cada uno a su correspondiente área y se encuentran totalmente aislados, es decir no existe ningún tipo de comunicación entre ellos.

El proceso de negocio X: "gestión de ordenes de servicio" atraviesa las áreas funcionales A: Planeamiento y B: Producción y requiere, para una correcta ejecución, consumir información de los sistemas Java y PHP. La **Figura 6** representa gráficamente la organización funcional de la empresa modelo.

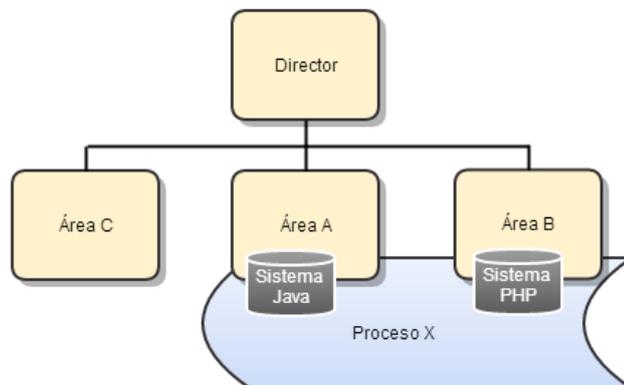


Figura 6. Organización funcional de la empresa del caso de estudio

La Dirección General creyó conveniente por una cuestión de eficiencia automatizar el proceso X. Una manera alternativa de integrar este proceso de negocio era desarrollar un nuevo sistema desde cero que reemplazará al sistema Java y PHP, sin embargo esta alternativa de reingeniería no fue viable por razones de costo y tiempo.

4.1 Breve descripción de la solución

La solución aconsejada y en conformidad con el modelo propuesto fue colocar el proceso de negocios X: "gestión de ordenes de servicio" en un motor BPEL para lograr su automatización y desarrollar el código que permitiera exponer a las dos aplicaciones como un Servicio Web para poder ser invocadas por el proceso de negocios.

Precisamente, cada Servicio Web debió implementar los métodos necesarios para consumir información del sistema subyacente. Estos métodos fueron invocados luego desde el proceso de negocio X: "gestión de ordenes de servicio" durante su ejecución y por medio de WS-BPEL se representó la lógica de ejecución del proceso y el orden de invocación de los métodos de los Servicios Web.

Por último, fue necesario exponer el proceso de negocio como un WS (Service Provider) y desarrollar una sencilla aplicación web que implemente un WS consumidor (Service Consumer) para que un usuario pudiese interactuar con el proceso de negocios.

A los efectos de simplificar la implementación consideramos que nuestro proceso X se limitara a invocar a un Servicio Web que implemente un único método, en vez de invocar a los dos Servicios Web que consumieran información de los sistemas Java y PHP. Para este fin se programó un Servicio Web que implemente el método *saludador(nombre, apellido)*, que recibiera los argumentos indicados y retornase un saludo concatenando el nombre y el apellido de esta manera: "Hola nombre apellido".

El resto del caso de estudio se centra en el desarrollo de los Servicios Web y en el modelado y ejecución del proceso X a través de una herramienta BPMS siguiendo el modelo propuesto.

4.2 Pasos previos: La herramienta BPMS

Antes de comenzar a aplicar el método fue sumamente indispensable seleccionar la herramienta BPMS y configurar el entorno de desarrollo.

BPM es la disciplina para definir, administrar y ejecutar procesos de negocio como un activo corporativo. Un BPMS provee la tecnología que implementa una o más de estas funciones centrales (28).

BPMS es la suite de tecnologías BPM, lo que incluye todos los módulos funcionales, las capacidades técnicas y la infraestructura de apoyo, integradas en un único entorno que realiza todas las funciones de la tecnología BPM de manera perfecta, sin fisuras. “BPMS es el paquete completo”.

Un BPMS debe proveer (28):

- Modelización de procesos: permite capturar los requerimientos de negocio en su etapa inicial y dejarla disponible durante el resto del proceso de desarrollo.
- Ejecución de procesos: la máquina de ejecución de procesos de la suite BPM importa el proceso modelado (definido usando BPEL) y luego ejecuta y administra las instancias de procesos para alcanzar los requerimientos operacionales.
- Monitoreo de procesos: esta capacidad incluye ver el resumen de los procesos en ejecución, de los completados, ver estados de procesos, suspender y reanudar procesos, dar alertas y reasignar procesos.
- Monitorear la actividad de negocio: (*Business Activity Monitoring* - BAM) analiza los eventos generados por la actividad de negocio y permite dar métricas.

Se tomó como referencia el análisis de herramientas realizado en (29) a los efectos de seleccionar la adecuada para la implementación de este caso de estudio. La elegida fue *Intalio BPMS*. Entre las virtudes que justificaran su elección resaltó el esquema adaptable que proporciona al usuario, el soporte a un conjunto amplio de estándares de la industria, ser de código abierto en su totalidad, la existencia de una amplia comunidad de usuarios y desarrolladores que colaboran con mejoras, detección y corrección y la disponibilidad de abundante documentación y screencasts que permiten mantener capacitado al usuario.

Intalio BPMS (30) es un software Open Source basado en *Java J2EE*², en frameworks y arquitecturas ampliamente utilizadas en la industria del software que proporciona una amplia plataforma de tipo empresarial para diseñar, desplegar los procesos de negocio más complejos.

Intalio BPMS cuenta con cada una de las cuatro funcionalidades que un buen BPMS debe proveer:

- Modelización de procesos, a través de *Intalio Designer*. Una herramienta para el diseño visual de los procesos de negocio basada en *Eclipse*.
- Ejecución de procesos, a través de *Intalio Server*. Un servidor de aplicaciones de alto rendimiento para la ejecución de procesos de negocio, basado en *Tomcat*.
- Monitoreo de procesos y monitorear la actividad de negocio, a través de *Intalio Console*. Ofrece la capacidad de gestionar todos los procesos e instancias de los procesos. Es posible monitorear el estado de un proceso, iniciarlo, desplegarlo, conocer el historial de ejecuciones y como han finalizado, y acceder a sus archivos de configuración y archivos de registro, entre otras prestaciones.

Las aplicaciones creadas con herramientas BPMS que soportan el estándar XPD, *Intalio BPMS* en una de ellas, son portables lo que nos permite cambiar de herramienta en la

²Java 2 Platform Enterprise Edition

mitad de un proyecto si fuera necesario y reutilizar diseños creados durante otros proyectos o contextos organizacionales.

4.3 Etapa 1: Identificación y análisis de los procesos de negocios

En esta etapa nos abocamos al análisis del proceso X, el cual debía permitir a un usuario (humano) completar un formulario web con su nombre y apellido como datos de entrada y luego de enviarlo recibir como respuesta un mensaje de confirmación de comunicación exitosa mediante un mensaje personalizado.

4.4 Etapa 2: Implementación de los Servicios Web

Existen dos enfoques básicos para el desarrollo de los Servicios Web desde cero: por una lado tenemos la aproximación top-down y por otro la bottom-up (31).

En el enfoque top-down primero se crea el WSDL y a partir de éste se genera el código del servicio web.

La descripción del Servicio Web es creada en XML (WSDL) antes de que éste sea implementado, así el servicio se describe completamente en términos de lo que hace, como puede ser llamado y que resultados devuelve, pero aún no se encuentra implementado. Esta especificación (WSDL) es luego usada como guía para escribir el código que implementa el servicio.

En el enfoque bottom-up el código que implementa el servicio web se escribe primero, luego a partir de éste se crea la descripción en XML (WSDL).

Existen ciertas ventajas y desventajas asociadas al uso de uno u otro enfoque de acuerdo a la capacidad y experiencia del programador, entorno de desarrollo y la complejidad del WS entre otras. Sin embargo se podrá optar por utilizar arbitrariamente uno u otro enfoque.

Para nuestro caso de estudio fueron aplicados los dos enfoques.

En primera instancia fueron implementados los dos Servicios Web usando un enfoque top-down, elegido arbitrariamente. Se desarrolló:

- un Servicio Web proveedor con PHP utilizando la clase SoapServer de su librería estándar (*Standard PHP Library - SPL*).
- un cliente o consumidor del Servicio Web con PHP utilizando la clase SoapClient de su librería estándar (SPL). Concretamente se trata de un cliente de servicios Web desplegado en un Servidor web Apache que utiliza la funcionalidad ofrecida por el proceso de negocio.

En segunda instancia, se modeló el proceso de negocio que invoca los Servicios Web utilizando un enfoque bottom-up.

Los procesos de negocios automatizados que se ejecutan en un motor BPEL constituyen en sí mismo un Servicio Web, y como tal pueden ser invocados por un Servicio Web cliente.

Al comenzar el diseño del proceso con la herramienta *Intalio Designer* aún no se cuenta con el WSDL, recién una vez finalizado el modelado se dispone del WSDL, que es generado automáticamente por la herramienta.

Respecto a la granularidad de los Servicios Web, por tratarse de un Caso de Estudio simplificado de la organización real, nos hemos limitado a implementar solo los métodos requeridos por el proceso X.

4.4.1 Creación del WSDL

Necesariamente los WSDL del Servicio Web cliente y proveedor debieron ser creados por el desarrollador. Únicamente el WSDL del proceso de negocio fue creado automáticamente, como se mencionó.

Crear el archivo WSDL de un Servicio Web manualmente programando en XML puede ser fuente de errores. Para ello el *Intalio Designer* al estar basado en Eclipse provee la funcionalidad que permite crearlo de manera mucho más sencilla utilizando un asistente visual. El sitio oficial de Eclipse, en su sección de ayuda provee un completo tutorial para crear el WSDL desde cero (32).

4.5 Etapa 3: Modelado del proceso de negocio

El proceso de negocio, como se mencionó anteriormente, se modeló con aplicación *Intalio Designer*, utilizando la notación BPMN.

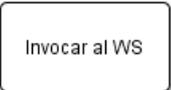
No se pretende en este informe dar una visión exhaustiva de la notación BPMN, solo se describirán brevemente los elementos utilizados para modelar el proceso X de nuestro caso de estudio (33); **Tabla 2**. Para mayor información se puede remitir a especificación oficial de BPMN (34) provista por la OMG (13).

Se comenzó definiendo un Pool por cada participante del proceso. Un pool justamente representa a un participante en el proceso y son utilizados como un mecanismo para organizar las actividades del proceso dentro de categorías visuales diferenciadas con el fin de denotar diferentes capacidades funcionales o responsabilidades.

El pool “WS Cliente” representa la interfaz con el usuario, es el Servicio Web cliente encargado de invocar al proceso de negocio, que conceptualmente es un proceso automatizado ejecutándose en un motor BPEL, pero a los efectos prácticos y de implementación se trata de un Servicio Web que encapsula el comportamiento y lógica de negocios del proceso.

El pool “Proceso de Negocio” representa al proceso de negocio propiamente dicho y el pool “Servicio Web” representa al servicio web proveedor, el cual se invoca para realizar la concatenación del nombre y apellido ingresados.

Tabla 2. Notación BPMN utilizada

	Evento de inicio. Un evento de este tipo causa que el proceso comience.
	Una actividad es un tipo de trabajo genérico que una empresa o compañía realiza. Una tarea es usada para representar una actividad en un bajo nivel de abstracción.
	Evento de fin. Un evento de mensaje de fin indica que un mensaje es enviado al término del proceso.
	Secuencia de flujo normal. Una secuencia de flujo es usada para mostrar el orden en el que las actividades del proceso son realizadas.
	Flujo de mensajes. Un flujo de mensaje es usado para mostrar el flujo entre dos participantes que son preparados para enviar y recibir los mensajes.

Luego se modelaron en cada Pool las actividades que realizan cada participante del proceso y

a través de las fechas se representó el orden de ejecución de las actividades y el flujo de información.

El contenido de los mensajes SOAP intercambiados entre los tres participantes se especificó utilizando la utilidad *Data Mapper* incluida en el *Intalio Designer*. Esta asignación de datos se realiza de una manera gráfica a través de un asistente.

Los tipos de datos contenidos en los mensajes fueron definidos usando XML Schema³, al igual que cuando se crea un archivo WSDL. En esta instancia la herramienta también provee una manera visual de hacerlo.

La **Figura 7** muestra el diagrama BPMN del proceso X.

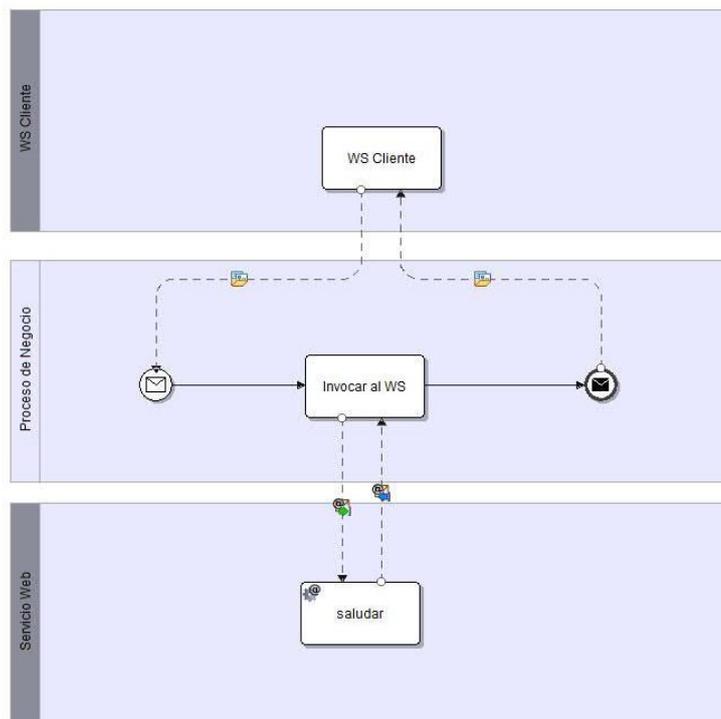


Figura 7. Notación BPMN utilizada

Una vez que el proceso de negocios fue totalmente definido se procedió a ejecutar la operación *Deploy*, la cual valida el modelo BPMN, genera el código BPEL correspondiente de manera totalmente automática y define los archivos WSDL y XSD (*XML Schema Definition*) para que el proceso de negocios X (ahora convertido en un Servicio Web) pueda ser invocado por el Servicio Web cliente.

Así, un proceso BPEL quedó expuesto como un servicio definido con WSDL y puede ser invocado por otro Servicio Web.

La operación de *Deploy* anteriormente mencionada, se realizó a través de la herramienta *Intalio Designer*. La función principal de esta operación es validar el modelo BPMN y desplegar el WSDL y XSD en el *Intalio Server*, también denominado “servidor de procesos”.

Finalmente se colocó una copia del WSDL y XSD en el “servidor web cliente”. Éstos hacen las veces de contrato para permitir que el servicio web cliente pueda invocar al proceso de negocios X.

³ Describe la estructura y las restricciones de los contenidos de los documentos XML.

4.5.1 Ejecución del proceso de negocio en motor BPEL

BPEL es un lenguaje de orquestación de servicios Web basado en XML, que sirve para definir la forma en que van a cooperar los Servicios Web para realizar el proceso X.

Existen algunas ventajas que trae aparejada la utilización de BPEL. Primero, permite concentrarse en la interacción de los Servicios Web y no de la funcionalidad de los mismos (definida en las especificaciones WSDL de cada uno). Segundo, expresa la lógica de colaboración y la secuencia de eventos, mientras que los Servicios Web proveen las funcionalidades individuales. Tercero, permite exponer a los procesos de negocio como servicios, haciendo del servicio de orquestación una operación recursiva que permite crear servicios complejos de otros más simples (35).

La programación del proceso de negocio puede realizarse de manera tradicional (ya que está basado en XML), pero en el uso cotidiano, BPEL se programa de forma gráfica utilizando la notación BPMN, y es la propia herramienta de diseño (*Intalio Designer* en nuestro caso) la encargada de atravesar la cadena de valor BPMN-XPDL-BPEL para obtener un proceso de negocio completamente portable y capaz de ejecutarse en un motor BPEL. Así, un proceso BPEL se expone como un servicio definido con WSDL y permite ser invocado por otro WS (28).

4.6 Etapa 4: Implementar la interface de interacción con usuarios

Se desarrolló una sencilla aplicación web en PHP que implementó un WS consumidor (Service Consumer) para que un usuario pueda interactuar con el proceso de negocios.

4.7 Etapa 5: Despliegue

El servicio web cliente se ejecutó en un servidor web HTTP Apache, desplegado en un equipo llamado “*servidor web cliente*”. El servicio web proveedor se ejecutó también en un servidor web HTTP Apache, pero desplegado en otro equipo, llamado “*servidor web proveedor*”.

El proceso de negocios X se ejecutó en un servidor *Intalio* desplegado también en otro equipo, llamado “servidor de procesos”.

El modelo de despliegue puede verse en la **Figura 8**.

Los servidores se comunican entre sí a través de mensajes SOAP sobre HTTP.

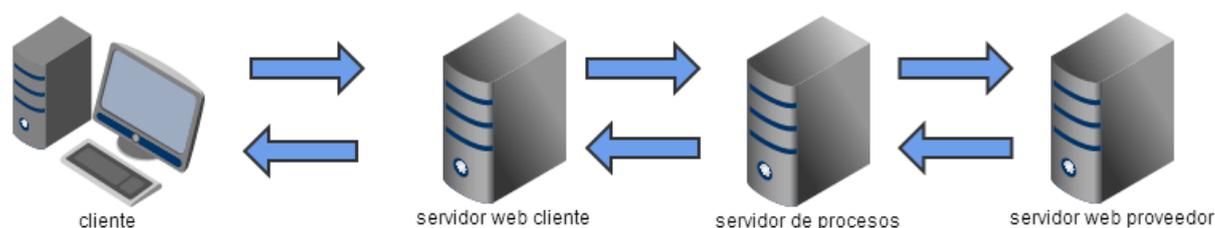


Figura 8. Modelo de despliegue

Es útil mencionar que no existe ningún tipo de restricción que nos lleve a desplegar los Servicios Web y el proceso de negocios de la manera indicada. Hubiese sido posible por ejemplo hacerlo todo en un solo equipo que tenga instalado un servidor web HTTP (corriendo sobre el puerto TCP 80) donde se desplieguen el servicio web cliente y el proveedor, y un

motor BPEL (sobre el puerto TCP 8080) donde se despliegue el proceso de negocio.

A continuación se detalla la secuencia en la que se produce la comunicación al ejecutar el proceso X. Se puede complementar gráficamente con la **Figura 7** y la **Figura 8**.

1. El usuario establece una comunicación con el “servidor web cliente” y se le presenta un formulario que le solicita el ingreso de su nombre y apellido.
2. El “servidor web cliente” donde se ejecuta el servicio web cliente realiza la invocación del proceso de negocios enviándole un mensaje de tipo Request al “servidor de procesos”, pasándole como argumento el nombre y el apellido ingresados.
3. El proceso de negocios que se ejecuta en el “servidor de procesos” realiza un llamada al método Saludador (nombre, apellido) del servicio web proveedor que se ejecuta en el “servidor web proveedor”.
4. El servicio web proveedor realiza la concatenación del nombre y apellido recibidos como argumento y los devuelve al proceso que lo invocó.
5. El proceso de negocios envía un mensaje de tipo Response al servicio web cliente con resultado de la solicitud.
6. El cliente visualiza en el navegador el saludo como respuesta a su petición.

4.8 Etapa 6: Administración y seguimiento

La secuencia de comunicación arriba citada fue monitoreada a través del *Intalio Console*, que hace las veces de dashboard o panel de control y permite al administrador estar al tanto del progreso del proceso y evaluar su performance.

En implementaciones de procesos interorganizacionales es común instaurar para su monitoreo una infraestructura de dashboard federado (8) de manera que permita compartir la actividad entre las organizaciones involucradas.

5. Discusión

El modelo propuesto permitió abordar una solución BPI de una manera ordenada y pautar claramente el ciclo de vida de los procesos de negocio, reduciendo la brecha entre estos y su realización en un software. Consecuentemente acercó distancias entre los analistas de procesos del área de negocios y el área de tecnología de la PyME.

En el desarrollo del Caso de Estudio se aplicaron cada una de las etapas del modelo propuesto con resultados satisfactorios que permitieron evidenciar los beneficios descriptos a continuación:

- Facilita la definición de procesos de negocios flexibles y adaptables a cambios organizacionales.
- Es apto para la reingeniería de los procesos de la organización.
- Favorece la identificación de funcionalidades transversales y el reuso de sistemas de software existentes en una PyME, definiendo un modelo de integración que facilite el acceso desacoplado a tales activos.

El enfoque divide la aplicación en dos capas estrictamente desacopladas: la capa superior (el proceso de negocio) representa la lógica de flujo de la aplicación, mientras que la capa inferior (los servicios web) representa la lógica de funcionamiento de la aplicación. Esta estructura de capas tiene varias ventajas sobre los paradigmas más convencionales. El proceso de negocio subyacente, así como los servicios web invocados pueden ser modificados

ante un cambio en la lógica de negocio sin sufrir ningún impacto el resto de los elementos del sistema. Así, este modelo proporciona una gran flexibilidad para incorporar cambios y lo hace sumamente escalable.

Al ser modular permite trabajar la solución en dos ambientes separados uno del otro sin tener que preocuparse demasiado por la integración. En uno encontraremos directores y analistas de procesos abocados a la capa superior y en otro un equipo de desarrolladores a cargo de la capa inferior.

La solución propuesta utilizando una arquitectura SOA basada en Servicios Web no requiere de la implementación de UDDI cuando los servicios son utilizados puertas adentro y no se hacen públicos para otras organizaciones. Eso no implica que a futuro pueda hacerse.

Cabe mencionar que este modelo no queda exento de mejoras por abordar. Se pudo observar que al implementar las tecnologías de servicios Web (con o sin orquestación mediante BPEL) se agregan fuentes de errores, ya que si falla una parte de la arquitectura, el proceso se ve afectado. También dificulta la detección de errores tanto en un ambiente de desarrollo como en producción: Es una tarea compleja poder detectar en qué punto del proceso se ha producido el error. Las herramientas de monitoreo dan indicios de los fallos y posibles diagnósticos pero no siempre alcanzan para recuperar el sistema en un tiempo razonable. Cuando ocurren hechos inesperados como la congestión o caída de un servidor que ejecuta un servicio Web que forma parte del flujo actual de trabajo, se afecta la performance general del sistema (29).

6. Conclusiones y Trabajo Futuro

En este trabajo hemos analizado la relación entre los Servicios Web y los procesos de negocio. Demostramos que un Servicio Web se puede usar como la implementación de una actividad dentro de un proceso de negocio, y que un proceso a su vez puede exteriorizarse como Servicio Web. Esta cualidad permitirá conformar jerarquías o composiciones de Procesos de Negocio para brindar soluciones BPI menos acopladas y reusables. Aunque en este enfoque de Servicios Web es fácil exponer una aplicación como un servicio, alguna aplicación de software existente en la organización podría requerir modificaciones.

El alcance de la gestión de procesos de negocio (BPM) se ha abordado conforme al modelo propuesto desde una perspectiva integradora, partiendo de la especificación de los procesos de negocio basado en un modelo de programación asociado, hasta el monitoreo y seguimiento de los procesos desplegados.

Por medio de un caso de implementación fue posible ubicar a *Intalio BPMS* como unas de las suites más potentes que provee el mercado actual de software libre y trabaja en conformidad con todos los estándares BPM.

Comparándola con la tradicional arquitectura cliente-servidor, SOA se focaliza en la organización de los procesos de negocios más que en la infraestructura tecnología en sí, por ello da ventaja competitiva a las empresas que la implementan, por su capacidad de reaccionar más rápido a los cambios en los requerimientos de negocio. Tanto el software empresarial como la Web en general se basan en un diseño “*message-based*” conformado la Web de los Servicios. La estandarización propuesta por la W3C⁴ en cada tecnología que interviene es clave en la evolución conjunta de las mismas.

Un desafío a futuro es extender el modelo propuesto a un ciclo de vida aplicable a proyectos que involucren BPI a través de una arquitectura SOA y Servicios Web, que abarque

⁴World Wide Web Consortium

desde los procesos previos al desarrollo hasta los procesos integrales, tales como la gestión de proyectos, riesgos y configuración pero desde una perspectiva concretamente orientada a servicios.

Otro aspecto posible a investigar surge de la necesidad de buscar maneras de incrementar la calidad, los tiempos de desarrollo y la reusabilidad en soluciones de integración de sistemas mediante el uso de patrones de integración empresarial (*Enterprise Integration Patterns*) (36) que proporcionen maneras adecuadas de integrar los Servicios Web para resolver problemas comunes junto con buenas prácticas en el WSF (*Web Services Framework*) de base.

A pesar de que popularmente se generaliza el concepto de servicio web con el de servicio Web basado en SOAP, es posible también diseñar servicios web con XML-RPC o tomar REST como modelo de diseño.

Al momento de elegir entre XML-RPC, SOAP o REST surge el requisito de evaluar la situación particular y el entorno empresarial, ya que todas son opciones válidas e interesantes como forma de conexión de Servicios Web.

Tecnológicamente, las nuevas tendencias nos llevan a implementar SOA mediante REST (37), el cual es un estilo arquitectural de software para sistemas hipermedia distribuidos tales como la web, se centra en el uso de los estándares HTTP y XML para la transmisión de datos sin la necesidad de contar con una capa adicional, como lo hace SOAP (38). Mientras que SOAP se caracteriza por seguir un RPC-style, REST tiende a un estilo data-centric u orientado a recursos, independiente del protocolo (39). Quedamos atentos a darle crédito a este tema por su prometedor futuro y rápida adopción por las empresas más grandes de la actualidad a la vanguardia en SOC (Service Oriented Computing).

Glosario de Siglas y Acrónimos

BAM: Business Activity Monitoring

BPEL: Business Process Execution Language

B2B: Business To Business

WS-BPEL: Web Service Business Process Execution Language

BPI: Business Process Integration

BPM: Business Process Management

BPMN: Business Process Management Notation

BPMS: Business Process Management Suite

EAI: Enterprise Application Integration

EIP: Enterprise Integration Patterns

HTTP: Hypertext Transfer Protocol

J2EE: Java 2 Platform Enterprise Edition

IETF: Internet Engineering Task Force

MTOM: Message Transmission Optimization Mechanism

PyME: Pequeña y Mediana Empresa

REST: Representational State Transfer

RPC: Remote Procedure Call

RRSHB: Resource Representation SOAP Header Block. Bloque de Cabecera SOAP de Representación de Recursos

SOA: Service Oriented Architecture

SOAP: Simple Object Access Protocol

SOC: Service Oriented Computing

SPL: Standard PHP Library

UDDI: Universal Description, Discovery and Integration

W3C: World Wide Web Consortium

WS: Web Service

WSDL: Web Services Description Language

WSF: Web Services Framework

XPDL: XML Process Definition Language

XSD: XML Schema Definition

XOP: XML-binary Optimized Packaging

Agradecimientos

Este trabajo ha recibido el soporte del Proyecto de Investigación UNPA 29/B167 "Identificación, Desarrollo y Uso de Soluciones Web centradas en el Usuario".

Referencias

1. **DeveloperWorks, IBM.** Introducción a SOA y Servicios Web. [En línea] [Citado el: 29 de 03 de 2014.] <<http://www.ibm.com/developerworks/ssa/webservices/newto/websvc.html>>.
2. **Karthikeyan Umapathy, Sandeep Purao.** Systems Integration and Web Services. s.l. : IEEE, 2010.
3. **Reyes, José Villasís, Gualotuña, Tatiana y Hinojosa, Cecilia.** Metodología para el análisis, diseño e implementación de procesos con tecnología BPM.
4. **Delgado, Andrea, González, Laura y Piedrabuena, Federico.** Desarrollo de aplicaciones con enfoque SOA. Montevideo, Uruguay : Universidad de la República, Facultad de Ingeniería, Instituto de Computación.
5. **Bazán, Patricia.** Un modelo de integrabilidad con SOA y BPM. s.l. : Facultad de Informática Universidad Nacional de La Plata, 2009.
6. **Raut, Ashutosh y Basavaraja, Ashwin.** Enterprise Business Process Integration. s.l. : IEEE, 2003.
7. **Hafedh, Mli, Tremblay, Guy y Bou jaoude, Guitta.** Business Process Modeling Languages: Sorting Through the Alphabet Soup. Montreal, Canada : s.n., 2010.

8. **Leymann, F., Roller, D. y Schmidt, M. T.** Web services and business process management. s.l. : IBM Systems Journal, 2002. Vol. vol 41, No 2.
9. **OASIS.** Web Services Business Process Execution Language Version 2.0. s.l. : OASIS Standard, April 2007.
10. **Martinez, Billy Crosby Orellana Orantes y Cristian Eneida Vazquez.** Integración de procesos de negocios. San Salvador, El Salvador C.A. : s.n., 2006.
11. **Álvarez, Gustavo.** Integración de procesos de negocio de alto nivel con soluciones EAI. Colombia : Universidad de los Andes.
12. **Zorzan, Fabio A. y Riesco, Daniel.** Transformación de Procesos BPMN a su Implementación en BPEL utilizando QVT. s.l. : WICC 2010, XII Workshop de Investigadores en Ciencias de la Computación, 2010.
13. **Object Management Group.** Workflow Management Facility Request for Proposals. OMG Framingham, MA : RFP Document Number cf/97-05-03, May 1997.
14. **White, Stephen A.** XPD and BPMN. SeeBeyond, United States : Extracted with permission from the Workflow Handbook 2003 published by Future Strategies Inc., in collaboration with the WfMC.
15. **Swenson, Keith.** Collaborative Planning & Social Business. *The BPMN-XPDL-BPEL value chain*. [En línea] May de 2006. [Citado el: 29 de 03 de 2014.] <<http://socialbiz.org/2006/05/26/bpmn-xpdl-and-bpel/>>.
16. **Krafizg, D., Banke, K. y Slama, D.** *Enterprise SOA - Service Oriented Architecture Best Practices*. USA : Pearson Education, Inc., 2005.
17. **Valani, D., Chawan, R. K. y Ghazali, S.** SOA Enabled BPM Rule Engine. TCET, Numbai, India : ICWET 2011, 2011.
18. **Leymann, Frank y Roller, Dieter.** Business processes in a web services world. *A quick overview of BPEL4WS*. 2002.
19. **Curbera, Francisco, y otros.** Unraveling the Web Services Web - An Introduction to SOAP, WSDL, and UDDI. 2002.
20. **W3C.** Web Services Description Language (WSDL) Version 2.0 Part 1: Core Language. *W3C Recommendation 26 June 2007*. [En línea] June de 2007. [Citado el: 29 de 03 de 2014.] <<http://www.w3.org/TR/2007/REC-wsdl20-20070626/>>.
21. —. Simple Object Access Protocol (SOAP) 1.1. *W3C Note 08 May 2000*. [En línea] May de 2000. [Citado el: 29 de 03 de 2014.] <<http://www.w3.org/TR/2000/NOTE-SOAP-20000508/>>.
22. **Cerami, Ethan.** Top Ten FAQs for Web Services. [En línea] 2002. [Citado el: 29 de 03 de 2014.] <<http://webservices.xml.com/pub/a/ws/2002/02/12/webservicefaqs.html>>.
23. **W3C.** Web Services Description Language (WSDL) 1.1. W3C Note. [En línea] March de 2001. [Citado el: 29 de 03 de 2014.] <<http://www.w3.org/TR/wsdl>>.

24. **Hostetler, Glenn y Hasznos, Sandor.** SOAP - Here's What Every Project Manager Should Know About It. [En línea] [Citado el: 29 de 03 de 2014.] <<http://www.practicingsafetechs.com/TechsV1/SOAP/>>.

25. **Goland, Y.** HTTP Extensions for Distributed Authoring. s.l. : WEBDAV IETF RFC 2528, Febrero, 1999.

26. **W3C.** Guía Breve de Servicios Web. [En línea] [Citado el: 29 de 03 de 2014.] <<http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>>.

27. **Gudgin, Martin Microsoft, y otros.** XML-binary Optimized Packaging. *W3C Recommendation 25 January 2005.* [En línea] [Citado el: 29 de 03 de 2014.] <<http://www.w3.org/TR/2005/REC-xop10-20050125/>>.

28. **Bazán, Patricia.** BPEL: una propuesta para el uso de Web Services. s.l. : LINTI. Laboratorio de Investigación en Nuevas Tecnologías Informáticas. Facultad de Informática, Universidad Nacional de La Plata.

29. **Baroni, Federico y Lerman, Ricardo.** Desarrollo de aplicaciones de comercio electrónico utilizando servicios web y herramientas BPEL. s.l. : UTN Facultad Regional Santa Fe. Argentina.

30. **Intalio.** Intalio BPMS. *Intalio / bpms, the first open source Business Process Management Suite.* [En línea] [Citado el: 29 de 03 de 2014.] <<http://www.intalio.com/products/bpms/overview/>>.

31. **IBM.** Feature Pack for Web Services - Top-down development of a JAX-WS service. s.l. : IBM Corporation 2007, 2007.

32. **Eclipse.** Eclipse documentation - Archived Release. *Creating a new WSDL file.* [En línea] [Citado el: 29 de 03 de 2014.] <<http://help.eclipse.org/>>.

33. **Polančič, Gregor.** Business Process Modeling with BPMN 2.0. [En línea] [Citado el: 29 de 03 de 2014.] <<http://www.slideshare.net/grepol/modelling-with-bpmn-2-0>>.

34. **Object Management Group.** Business Process Model And Notation (BPMN) Version 2.0. [En línea] January de 2011. [Citado el: 29 de 03 de 2014.] <<http://www.omg.org/spec/BPMN/2.0/>>.

35. **Lapadula, Alessandro, Pugliese, Rosario y Tiezzi, Francesco.** Using formal methods to develop WS-BPEL applications. 2011.

36. **Woolf, G. Hohpe and B.** Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. s.l. : Addison-Wesley, 2004.

37. **Fielding, Roy Thomas.** Architectural Styles and the Design of Network-based Software Architectures. University of California, Irvine : s.n., 2000. Vol. Chapter 5. Representational State Transfer (REST).

38. **Pautasso, Cesare.** Rest vs. Soap: Making the Right Architectural Decision. s.l. : SOA Symposium 2008, Amsterdam, 2008.

39. **Fielding, Roy T.** Untangled musings of Roy T. Fielding. [En línea] Octubre de 2008. [Citado el: 29 de 03 de 2014.] <<http://roy.gbiv.com/untangled/2008/rest-apis-must-be-hypertext-driven>>.

Anexo I: Código de formulario de interacción con usuario

index.php

```
1. <h1>Ingrese su nombre y apellido para ser saludado</h1>
2. <p>Este es un cliente de WS y al presionar enviar el proceso de negocio (que se encuentra corriendo en el BPEL engine) es llamado como un web service y este proceso a su vez <br/>
3. se comunica con el web service "saludador" (que se encuentra corriendo en un servidor web apache). <br/>
4. Luego de ejecutarse el WS "saludador" el proceso devuelve el saludo al cliente de WS que inicialmente lo invoco.</p>
5.
6. <form name="form" method="get" action="cliente.php">
7.     <p>Nombre: <input type="text" name="imp_nombre"></p>
8.     <p>Apellido: <input type="text" name="imp_apellido"></p>
9.     <p><input type="submit" name="boton" value="Enviar"></p>
10. </form>
```

cliente.php

```
1. <?php
2.
3. ini_set("soap.wsdl_cache_enabled", "0");
4.
5. $nombre=$_GET["imp_nombre"];
6. $apellido=$_GET["imp_apellido"];
7. $params = array('nombre' => $nombre, 'apellido'=>$apellido);
8.
9.
10. $s=new SoapClient("http://localhost/DAW/saludador/WS-Proceso_de_Negocio.wsdl");
11.
12.
13. $res=$s->EventStartMessage($params);
14.
15. echo "<h2>";
16. echo $res->saludo; //lo devuelve como un objeto (object)
17. echo "</h2>";
18. ?>
19.
20. <a href="http://localhost/DAW/saludador/index.php" title="volver a principal">
    Volver a saludador</a>
```

Anexo II: WSDLs

saludador.wsdl

```
1. <?xml version="1.0" encoding="UTF-8" standalone="no"?>
2. <wSDL:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="http://
www.example.org/saludador/" xmlns:wSDL="http://schemas.xmlsoap.org/wsdl/" xmlns:xsd="ht
tp://www.w3.org/2001/XMLSchema" name="saludador" targetNamespace="http://www.example.or
g/saludador/">
3.   <wSDL:types>
4.     <xsd:schema targetNamespace="http://www.example.org/saludador/">
5.       <xsd:element name="saludar" type="tns:saludarType">
6.
7.       </xsd:element>
8.       <xsd:element name="saludarResponse"
9.         type="tns:saludarResponseType">
10.
11.       </xsd:element>
12.
13.       <xsd:complexType name="saludarType">
14.         <xsd:sequence>
15.           <xsd:element name="nombre" type="xsd:string"></xsd:element>
16.           <xsd:element name="apellido" type="xsd:string"></xsd:element>
17.         </xsd:sequence>
18.       </xsd:complexType>
19.
20.       <xsd:complexType name="saludarResponseType">
21.         <xsd:sequence>
22.           <xsd:element name="saludo" type="xsd:string"></xsd:element>
23.         </xsd:sequence>
24.       </xsd:complexType>
25.     </xsd:schema>
26.   </wSDL:types>
27.   <wSDL:message name="saludarRequest">
28.     <wSDL:part element="tns:saludar" name="parameters"/>
29.   </wSDL:message>
30.   <wSDL:message name="saludarResponse">
31.     <wSDL:part element="tns:saludarResponse" name="parameters"/>
32.   </wSDL:message>
33.   <wSDL:portType name="saludador">
34.     <wSDL:operation name="saludar">
35.       <wSDL:input message="tns:saludarRequest"/>
36.       <wSDL:output message="tns:saludarResponse"/>
37.     </wSDL:operation>
38.   </wSDL:portType>
39.   <wSDL:binding name="saludadorSOAP" type="tns:saludador">
40.     <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
41.     <wSDL:operation name="saludar">
42.       <soap:operation soapAction="http://www.example.org/saludador/saludar"/>
43.       <wSDL:input>
44.         <soap:body use="literal"/>
45.       </wSDL:input>
46.       <wSDL:output>
47.         <soap:body use="literal"/>
48.       </wSDL:output>
49.     </wSDL:operation>
50.   </wSDL:binding>
51.   <wSDL:service name="saludador">
52.     <wSDL:port binding="tns:saludadorSOAP" name="saludadorSOAP">
53.       <soap:address location="http://localhost/DAW/saludador/servidor.php"/>
54.     </wSDL:port>
55.   </wSDL:service>
56. </wSDL:definitions>
```

WS-Proceso_de_Negocio.wsdl

```

1. <?xml version='1.0' encoding='utf-8'?>
2. <wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:plnk="http://docs
   .oasis-
   open.org/wsbpel/2.0/plnktype" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:
   bpel="http://docs.oasis-
   open.org/wsbpel/2.0/process/executable" xmlns:vprop="http://docs.oasis-
   open.org/wsbpel/2.0/varprop" xmlns:this="http://saludador.com/WS/Proceso_de_Negocio" xm
   lns:xs="http://www.w3.org/2001/XMLSchema" xmlns:Servicio_Web="http://saludador.com/WS/S
   ervicio_Web" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
   instance" xmlns:tns="http://www.example.org/SaludadorSchema" xmlns:saludador="http://ww
   w.example.org/saludador/" xmlns:diag="http://saludador.com/WS" xmlns:WS_Cliente="http:/
   /saludador.com/WS/WS_Cliente" xmlns:xml="http://www.w3.org/XML/1998/namespace" targetNa
   mespace="http://saludador.com/WS/Proceso_de_Negocio">
3.   <wsdl:types>
4.     <xs:schema elementFormDefault="qualified" targetNamespace="http://saludador.com
   /WS/Proceso_de_Negocio">
5.       <xs:import namespace="http://www.example.org/SaludadorSchema" schemaLocatio
   n="SaludadorSchema.xsd"/>
6.       <xs:element name="EventStartMessageRequest" type="tns:saludarType"/>
7.       <xs:element name="EventStartMessageResponse" type="tns:saludarResponseType"
   />
8.     </xs:schema>
9.   </wsdl:types>
10.  <wsdl:message name="EventStartMessageRequest">
11.    <wsdl:part name="body" element="this:EventStartMessageRequest"/>
12.  </wsdl:message>
13.  <wsdl:message name="EventStartMessageResponse">
14.    <wsdl:part name="body" element="this:EventStartMessageResponse"/>
15.  </wsdl:message>
16.  <wsdl:portType name="ForWS_Cliente">
17.    <wsdl:operation name="EventStartMessage">
18.      <wsdl:input message="this:EventStartMessageRequest" name="EventStartMessage
   "/>
19.      <wsdl:output message="this:EventStartMessageResponse" name="EventStartMessa
   geResponse"/>
20.    </wsdl:operation>
21.  </wsdl:portType>
22.  <wsdl:binding name="CanonicBindingForWS_Cliente" type="this:ForWS_Cliente">
23.    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"
   />
24.    <wsdl:operation name="EventStartMessage">
25.      <soap:operation style="document" soapAction="http://saludador.com/WS/Proces
   o_de_Negocio/ForWS_Cliente/EventStartMessage"/>
26.      <wsdl:input name="EventStartMessage">
27.        <soap:body use="literal"/>
28.      </wsdl:input>
29.      <wsdl:output name="EventStartMessageResponse">
30.        <soap:body use="literal"/>
31.      </wsdl:output>
32.    </wsdl:operation>
33.  </wsdl:binding>
34.  <wsdl:service name="CanonicServiceForWS_Cliente">
35.    <wsdl:port name="canonicPort" binding="this:CanonicBindingForWS_Cliente">
36.      <soap:address location="http://localhost:8080/ode/processes/Saludador/WS/Pr
   oceso_de_Negocio/WS_Cliente"/>
37.    </wsdl:port>
38.  </wsdl:service>
39. </wsdl:definitions>

```

Anexo III: web service provider

servidor.php

```
1. <?php
2. class Trabajo {
3.     private $nom;
4.
5.     public function saludar($obj){ //me trae un objeto
6.         $nom=$obj->nombre;
7.         $ape=$obj->apellido;
8.         $params = array('saludo' => 'Hola '.$nom.' '.$ape);
9.         return $params; //lo devuelvo al proceso como un array
10.    }
11.
12. }
13. $server=new SoapServer("saludador.wsdl");
14. $server->setClass("Trabajo");
15.
16. $server->handle();
17.
18. ?>
```


Intalio Console

Showing instances 1 - 25 of 25 matching instances.

Pages: First Previous 1 Next Last
Items per page: 10 20 50 100 ...

Process	State	Failures	Started	Last Active
Proceso de Negocio [V36]	Completed		2014-04-07 23:51:54	2014-04-07 23:51:55
Proceso de Negocio [V36]	Completed		2014-04-07 23:48:43	2014-04-07 23:48:46
Proceso de Negocio [V36]	Completed		2014-03-16 11:53:14	2014-03-16 11:53:15
Proceso de Negocio [V36]	Completed		2014-03-16 11:50:42	2014-03-16 11:50:45
Proceso de Negocio [V36]	Completed		2013-11-18 12:15:40	2013-11-18 12:15:41
Proceso de Negocio [V36]	Completed		2013-11-18 00:21:29	2013-11-18 00:21:30
Proceso de Negocio [V36]	Completed		2013-11-17 11:41:09	2013-11-17 11:41:11
Proceso de Negocio [V36]	Completed		2013-11-16 22:12:05	2013-11-16 22:12:06
Proceso de Negocio [V36]	Completed		2013-11-16 22:11:48	2013-11-16 22:11:53
Proceso de Negocio [V36]	Completed		2013-10-03 17:30:59	2013-10-03 17:30:59
Proceso de Negocio [V36]	Completed		2013-10-03 17:30:48	2013-10-03 17:30:48
Proceso de Negocio [V36]	Completed		2013-10-03 17:29:43	2013-10-03 17:29:44
Proceso de Negocio [V36]	Completed		2013-10-03 17:29:37	2013-10-03 17:29:38
Proceso de Negocio [V36]	Completed		2013-10-03 17:24:50	2013-10-03 17:24:51
Proceso de Negocio [V36]	Completed		2013-10-03 17:23:26	2013-10-03 17:23:27
Proceso de Negocio [V36]	Completed		2013-10-03 17:22:47	2013-10-03 17:22:48
Proceso de Negocio [V36]	Completed		2013-10-03 17:22:04	2013-10-03 17:22:04
Proceso de Negocio [V36]	Completed		2013-10-03 17:20:47	2013-10-03 17:20:48
Proceso de Negocio [V36]	Completed		2013-10-03 17:18:47	2013-10-03 17:18:48
Proceso de Negocio [V36]	Completed		2013-10-03 17:18:12	2013-10-03 17:18:12
WS1 Process [V21]	Failed		2013-10-02 20:04:28	2013-10-02 20:10:06
WS1 Process [V21]	Terminated	1	2013-10-02 20:02:59	2013-10-02 20:03:00
WS1 Process [V21]	Terminated	4	2013-10-02 20:02:59	2013-10-02 20:03:00