

**ARQUITECTURA ORIENTADA A SERVICIOS: PRINCIPIOS Y  
FUNDAMENTOS**

**PATRICIA KATHERINE ALVAREZ ROMERO  
REMBERTO MANUEL CARO PEÑA**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR  
FACULTAD DE INGENIERÍA  
PROGRAMA DE INGENIERÍA DE SISTEMAS  
CARTAGENA DE INDIAS D.T. Y C.  
2009**

**ARQUITECTURA ORIENTADA A SERVICIOS: PRINCIPIOS Y  
FUNDAMENTOS**

**PATRICIA KATHERINE ALVAREZ ROMERO  
REMBERTO MANUEL CARO PEÑA**

MONOGRAFÍA PRESENTADA COMO REQUISITO PARA OPTAR AL  
TÍTULO DE:

**INGENIERO DE SISTEMAS**

DIRECTOR  
**GIOVANNI VASQUEZ**

**UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR  
FACULTAD DE INGENIERÍA  
PROGRAMA DE INGENIERÍA DE SISTEMAS  
CARTAGENA DE INDIAS D.T Y C.**

**2009**

**Nota de aceptación**

---

---

---

---

---

**Firma del presidente del jurado**

---

**Firma del jurado**

---

**Firma del jurado**

**Cartagena, Diciembre de 2009**

**ARTICULO 107 DEL REGLAMENTO ACADÉMICO DE LA UNIVERSIDAD  
TECNOLÓGICA DE BOLÍVAR:**

La universidad se reserva el derecho de propiedad intelectual de todos los trabajos de grado aprobados, los cuales no pueden ser explotados comercialmente sin autorización.

**Cartagena de Indias D.T. y C Diciembre de 2009**

Señores:

**COMITÉ DE REVISIÓN DE MONOGRAFÍA  
UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR**

La Ciudad

Apreciados señores:

Por medio de la presente nos permitimos informarles que la monografía titulada **“ARQUITECTURA ORIENTADA A SERVICIOS: PRINCIPIOS Y FUNDAMENTOS”** ha sido desarrollada de acuerdo a los objetivos establecidos.

Como autores del proyecto consideramos que el trabajo es satisfactorio y amerita ser presentado para su evaluación.

Atentamente,

---

**REMBERTO MANUEL CARO PEÑA**  
Código: T00015376

---

**PATRICIA ALVAREZ ROMERO**  
Código: T00012222

**Cartagena de Indias D.T. y C Diciembre de 2009**

Señores:

**COMITÉ DE REVISIÓN DE MONOGRAFÍA  
UNIVERSIDAD TECNOLÓGICA DE BOLÍVAR**

La Ciudad

Apreciados señores:

Por medio de la presente me permito informarles que la monografía titulada **“ARQUITECTURA ORIENTADA A SERVICIOS: PRINCIPIOS Y FUNDAMENTOS”** ha sido desarrollada de acuerdo a los objetivos establecidos.

Como director considero que el trabajo es satisfactorio y amerita ser presentado para su evaluación.

Atentamente,

---

**GIOVANNI VASQUEZ**  
**Ingeniero de Sistemas,**  
**Magíster en Ciencias Computacionales,**  
**Certificado como Programador Tecnología Java.**

## **AUTORIZACIÓN**

**Cartagena de Indias D.T. y C Diciembre de 2009**

Yo Remberto Manuel Caro Peña, identificado con la cedula de ciudadanía numero 73.207.053 de Cartagena (Bolívar) autorizo a la Universidad Tecnológica de Bolívar para hacer uso de mi trabajo de grado y publicarlo en el catalogo online de la biblioteca.

---

**REMBERTO MANUEL CARO PEÑA**  
**C.C. 73.207.053 de Cartagena**

## **AUTORIZACIÓN**

**Cartagena de Indias D.T. y C Diciembre de 2009**

Yo Patricia Katherine Álvarez Romero, identificado con la cedula de ciudadanía numero 45.546.053 de la ciudad de Cartagena (Bolívar) autorizo a la Universidad Tecnológica de Bolívar para hacer uso de mi trabajo de grado y publicarlo en el catalogo online de la biblioteca.

---

**PATRICIA KATHERINE ALVAREZ ROMERO**  
**C.C. 45.546.053 de Cartagena**



*Dedico este trabajo a la persona que  
incondicionalmente me ha  
patrocinado, apoyado y motivado  
durante todos estos años de  
estudio y trabajo.  
Gracias Madre!*

***Remberto Caro Peña***

*Dedico este trabajo de investigación*  
*Al motor de mi vida, mi hija*  
*MICHELLE*  
**Patricia Álvarez Romero**

## AGRADECIMIENTOS

Agradezco a **Dios** por darme la vida, energía y salud necesarias para recorrer este camino.

A mi madre, **Rosa**, ser humano excepcional e incondicional que me ha apoyado en todo momento y diversas formas, sin ti madre esto no hubiera sido posible.

A mi jefe, amigo y patrocinador, **Juan Carlos Otoy**, gracias por creer en mí y darme la oportunidad de trabajar y estudiar.

A mi mentor, **Moisés Quintana**, quien sencillamente me enseñó a programar bien, eso marcó definitivamente los logros que he alcanzado hasta hoy.

A mi profesor y amigo **Giovanni Vásquez**, quien me enseñó los conceptos de la ingeniería de software, estos han sido y seguirán siendo esenciales en el desarrollo diario de mi trabajo.

A mis amigos **Patricia Álvarez y Cesar Rivas**, con los que he compartido, disfrutado, peleado y estudiado durante los últimos nueve años.

A todos los demás (compañeros, ex-novias, profesores, conocidos, etc.) que de una u otra manera me ayudaron durante toda la vida universitaria.

**Remberto Caro Peña**

## AGRADECIMIENTOS

Gracias a Dios

por permitirme llegar hasta este punto y cumplir una meta más.

A mis padres Elizabeth y Erick

por sus diversas formas de apoyo a lo largo de mi vida y guiarme por el camino de la educación.

A la persona que me ha acompañado incondicionalmente y con la que he compartido alegrías, tristezas y momentos inolvidables, Alfredo que me ha apoyado e impulsado a finalizar este trabajo.

A toda mi familia y amigos

que se encuentran esperando este momento

A mi compañero de monografía Remberto Caro

por su participación en este trabajo de investigación, conformar un excelente equipo y ser un gran amigo.

A Giovanni Vásquez

por su asesoría y dirección en el trabajo de investigación

y durante mis años de estudio

A cada uno de los maestros

que hicieron parte de mi desarrollo profesional durante mi carrera, sin su ayuda y conocimientos no estaría en donde me encuentro ahora.

Y a todas aquellas personas que de una u otra forma, colaboraron en la realización de esta investigación,

hago extensivo mi más sincero agradecimiento.

## CONTENIDO

<b>AGRADECIMIENTOS.....</b>	<b>11</b>
<b>LISTA DE FIGURAS.....</b>	<b>15</b>
<b>GLOSARIO .....</b>	<b>16</b>
<b>RESUMEN.....</b>	<b>20</b>
<b>INTRODUCCIÓN.....</b>	<b>22</b>
<b>LA ARQUITECTURA ORIENTADA A SERVICIOS .....</b>	<b>24</b>
<b>1.1. ANTECEDENTES DE LA ARQUITECTURA ORIENTADA A SERVICIOS.....</b>	<b>25</b>
<b>1.2. FUNDAMENTACION TEORICA.....</b>	<b>29</b>
<b>1.3. PRINCIPIOS DE LA ORIENTACION A SERVICIOS .....</b>	<b>37</b>
1.3.1. Componentes de una SOA.....	41
<b>1.4. CAPAS DE SERVICIO .....</b>	<b>46</b>
1.3.2. Capa de servicios de aplicación .....	48
1.3.3. Capa de servicios de negocio.....	51
1.3.4. Capa de servicios de orquestación .....	53
<b>ANALISIS ORIENTADO A SERVICIOS .....</b>	<b>56</b>
<b>2.1. INTRODUCCION AL ANALISIS.....</b>	<b>57</b>
<b>2.2. PASOS PARA REALIZAR UN ANALISIS ORIENTADO A SERVICIOS.....</b>	<b>60</b>
2.2.1. Paso 1: definir las necesidades de automatización de negocios.....	61
2.2.2. Paso 2: Identificar los sistemas de automatización.....	61
2.2.3. Paso 3: Modelo de servicios candidato.....	62
<b>2.3. BENEFICIOS DE UNA SOA CENTRADA EN NEGOCIOS.....</b>	<b>62</b>
2.3.1. Preparación de negocios para un proceso de orquestación.....	64

2.3.2.	Servicio de negocio posibilitan la reutilización.....	65
2.3.3.	Los servicios de negocios pueden realizar la orientación a servicios empresarial. ....	65
<b>2.4</b>	<b>DERIVACION DE LOS SERVICIOS DE NEGOCIO. ....</b>	<b>66</b>
2.4.1.	Fuentes de donde los servicios de negocio pueden ser derivados.....	67
2.4.1.1	Modelos de administración de procesos de negocio (BPM models).....	68
2.4.1.2	Modelos entidad .....	73
2.4.2.	Tipos de derivación de servicios de negocio. ....	75
2.4.2.1	Servicio de negocio centrado en tareas. ....	75
2.4.2.2	Servicio de negocio centrado en entidades. ....	76
2.4.2.3	Servicios de negocio y orquestación. ....	79
	<b>MODELADO ORIENTADO A SERVICIOS .....</b>	<b>81</b>
<b>3.1.</b>	<b>MODELADO DE SERVICIOS .....</b>	<b>82</b>
<b>3.2.</b>	<b>PASOS PARA MODELAR SERVICIOS.....</b>	<b>83</b>
<b>3.3.</b>	<b>CLASIFICACION DE LA LOGICA DEL MODELO DE SERVICIO .....</b>	<b>100</b>
3.3.1.	El modelo SOE.....	101
3.3.2.	El modelo de negocio y tecnológico empresarial.....	103
3.3.3.	Bloques de construcción para el modelado básico .....	103
	<b>DISEÑO ORIENTADO A SERVICIOS .....</b>	<b>105</b>
<b>4.1.</b>	<b>INTRODUCCION AL DISEÑO ORIENTADO A SERVICIOS.....</b>	<b>106</b>
<b>4.2.</b>	<b>COMPOSICION DE UNA SOA.....</b>	<b>108</b>
<b>4.2.</b>	<b>DISEÑO DE SERVICIOS DE NEGOCIO CENTRADO EN ENTIDADES.....</b>	<b>109</b>
<b>4.3.</b>	<b>DISEÑO DE SERVICIOS DE APLICACIÓN .....</b>	<b>115</b>
	<b>RECOMENDACIONES.....</b>	<b>119</b>
	<b>CONCLUSIONES .....</b>	<b>122</b>
	<b>BIBLIOGRAFÍA .....</b>	<b>124</b>

## LISTA DE FIGURAS

ESQUEMA DE UNA ARQUITECTURA ORIENTADA A SERVICIOS.....	34
RELACIÓN ENTRE LAS ENTIDADES.....	36
DOMINIOS DE LA LÓGICA DE APLICACIÓN Y DE NEGOCIO .....	38
CAPAS DE SERVICIO.....	40
INTERFACE DE SERVICIOS .....	41
UNIDAD DE TRABAJO.....	41
OPERACIONES DE SERVICIO .....	42
INTERRELACIÓN DE COMPONENTES .....	43
DEFINICIÓN DE COMPONENTES.....	44
CAPAS DE SERVICIO DETALLADAS.....	47
CAPA DE SERVICIOS DE APLICACIÓN .....	48
CAPA DE SERVICIOS DE NEGOCIO.....	52
CAPA DE SERVICIOS DE ORQUESTACIÓN .....	55
PASOS PARA REALIZAR UN ANÁLISIS ORIENTADO A SERVICIOS .....	60
PARTES DE UN PROCESO QUE PUEDEN SER ENCAPSULADAS POR UN SERVICIO DE NEGOCIO.....	70
RELACION ENTRE LAS ENTIDADES.....	74
PASOS PARA REALIZAR UN MODELADO ORIENTADO A SERVICIOS .....	83
PROCESO DE PRESENTACIÓN DE FACTURA .....	85
CANDIDATOS A SERVICIO .....	92
CANDIDATOS A SERVICIOS (REVISADOS).....	97
EL MODELO SOE .....	102
PASOS PARA REALIZAR UN DISEÑO ORIENTADO A SERVICIOS.....	107

## GLOSARIO

**Aplicación.** (Del lat. *applicatio*, *-ōnis*). f. *Inform.* Programa preparado para una utilización específica, como el pago de nóminas, formación de un banco de términos léxicos, etc.

**Arquitectura.** Consiste en el diseño de componentes de una aplicación generalmente utilizando patrones de arquitectura, que permiten visualizar la interacción entre las entidades de un negocio y además poder ser validado, por medio de diagramas de secuencia. En general la arquitectura describe cómo se construirá una aplicación de software, documentándose utilizando diagramas.

**Artefacto.** Es un producto tangible resultante del proceso de desarrollo de software. Algunos artefactos como los casos de uso, diagrama de clases u otros modelos UML ayudan a la descripción de la función, la arquitectura o el diseño del software. Otros se enfocan en el proceso de desarrollo en sí mismo, como planes de proyecto, casos de negocios o enfoque de riesgos. El código fuente compilado para el testeo se suele considerar un artefacto, ya que el ejecutable es necesario para el plan de testeo.

En ocasiones un artefacto puede referirse a un producto terminado, como el código o el ejecutable, pero más habitualmente se refiere a la documentación generada a lo largo del desarrollo del producto en lugar del producto en sí.

**Atributos.** Son los que están asociados a clases y objetos, ellos describen la clase o el objeto de alguna manera. Un atributo puede tomar un valor definido por un dominio enumerado. Un dominio es simplemente un conjunto de valores específicos.



**BPEL.** WS-BPEL (en castellano, *Lenguaje de Ejecución de Procesos de Negocio*), es un lenguaje estandarizado por OASIS para la composición de servicios web. Está desarrollado a partir de WSDL y XLANG, ambos lenguajes orientados a la descripción de servicios Web. Básicamente, consiste en un lenguaje basado en XML diseñado para el control centralizado de la invocación de diferentes servicios Web, con cierta lógica de negocio añadida que ayuda a la programación en gran escala (*programming in the large*).

**Componente.** Los componente de Software son todo aquel recurso desarrollado para un fin concreto y que puede formar solo o junto con otro/s, un entorno funcional requerido por cualquier proceso predefinido. Son independientes entre ellos, y tienen su propia estructura e implementación. Si fueran propensos a la degradación debieran diseñarse con métodos internos propios de refresco y actualización.

**Flujo de trabajo.** (Workflow en inglés) es el estudio de los aspectos operacionales de una actividad de trabajo: cómo se estructuran las tareas, cómo se realizan, cuál es su orden correlativo, cómo se sincronizan, cómo fluye la información que soporta las tareas y cómo se le hace seguimiento al cumplimiento de las tareas. Generalmente los problemas de flujo de trabajo se modelan con redes de Petri.

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

**Ontología.** En informática hace referencia a la formulación de un exhaustivo y riguroso esquema conceptual dentro de uno o varios dominios dados; con la finalidad de facilitar la comunicación y la compartición de la información entre diferentes sistemas y entidades.

**Plataforma.** En informática y tecnología, plataforma se refiere al sistema operativo o a sistemas complejos que a su vez sirven para crear programas, como las plataformas de desarrollo.

**Servicio web.** (Del inglés de *Web Service*) es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet. La interoperabilidad se consigue mediante la adopción de estándares abiertos. Las organizaciones OASIS y W3C son los comités responsables de la arquitectura y reglamentación de los servicios Web. Para mejorar la interoperabilidad entre distintas implementaciones de servicios Web se ha creado el organismo WS-I, encargado de desarrollar diversos perfiles para definir de manera más exhaustiva estos estándares.

**Sistema.** (lat. *systema*) es un conjunto de elementos interrelacionados e interactuantes entre sí.

**SOAP.** (siglas en inglés de *Simple Object Access Protocol*) es un protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de un protocolo creado por David Winer en 1998, llamado XML-RPC. SOAP fue creado por Microsoft, IBM y otros y está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los servicios Web.

**UDDI.** son las siglas del catálogo de negocios de Internet denominado *Universal Description, Discovery and Integration*. El registro en el catálogo se hace en XML. es uno de los estándares básicos de los servicios Web cuyo objetivo es ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se

describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios Web del catálogo de registros.

**WEB.** (Del ingl. *web*, red, malla). f. *Inform.* Red informática. La web es un sistema de hipertexto que utiliza Internet como su mecanismo de transporte.

**WSDL.** (siglas en inglés de *Web Services Description Language*) describe la interfaz pública a los servicios Web. Está basado en XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.

**XML.** (siglas en inglés de *Extensible Markup Language*), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.

**XML schema.** es un lenguaje de esquema utilizado para describir la estructura y las restricciones de los contenidos de los documentos XML de una forma muy precisa, más allá de las normas sintácticas impuestas por el propio lenguaje XML. Se consigue así una percepción del tipo de documento con un nivel alto de abstracción. Fue desarrollado por el World Wide Web Consortium (W3C) y alcanzó el nivel de recomendación en mayo de 2001.

## RESUMEN

SOA es mucho más que una arquitectura de TI, es un nuevo enfoque para las empresas que intentan unir tecnología y negocio. El mercado de este nuevo enfoque ha estado centrado en modelos de integración en donde las empresas aprovechaban la centralización y estandarización de los recursos tecnológicos. Pero esto es un concepto que evoluciona, y por ello, las Arquitecturas Orientadas a Servicios van a experimentar un cambio de paradigma, pasando de un modelo centrado en la integración a un modelo centrado en los procesos de negocio.

La arquitectura antes mencionada es por tanto un marco de trabajo para el desarrollo de software y su implantación. Y aunque la mayoría de las definiciones identifican la utilización de Servicios Web (SOA y WSDL) en su implementación, una arquitectura orientada a servicios se puede implementar utilizando cualquier tecnología basada en servicios.

La finalidad de la arquitectura es conseguir combinar distintos módulos funcionales para generar aplicaciones de carácter específico, proviniendo todos de servicios preexistentes, cuanto mayor sea la funcionalidad proporcionada por estos módulos, menor será el número de interfaces necesarios para alcanzar el objetivo deseado y cada interfaz conlleva un gasto de procesamiento adicional; sin embargo, cuando los módulos son excesivamente grandes resulta complicada su reutilización. Consecuentemente es necesario alcanzar el nivel de granulación adecuado.

La expectativa creada por esta nueva arquitectura es que el coste marginal de creación de la enésima aplicación sea cero dado que el software requerido se encontrará ya disponible para satisfacer los requisitos; tan sólo se requerirá la coordinación entre los elementos.

Una de las claves de esta arquitectura es que no existen interacciones entre los módulos que se encuentren embebidas. La interacción entre los servicios, los cuales son independientes, es especificada por los usuarios. De ahí la necesidad de que los servicios presenten mayor funcionalidad que las tradicionales funciones o clases de los lenguajes de programación evitando la desbordante complejidad que la existencia de miles de objetos granulares supondría para el diseñador. Los servicios en sí mismos se desarrollan mediante el uso de lenguajes de programación como C, C++, Java, etc. Estos servicios presentan un acoplamiento bajo, en oposición a las funciones que componen un fichero ejecutable. Los servicios se ejecutan en el interior de un envoltorio (wrapper) como Java o .NET, que se encarga de la gestión de la memoria, las invocaciones de los servicios, y el uso de tipos de datos.

Cada vez existe un mayor número de compañías de software que están ofreciendo servicios a cambio del pago de alguna tasa. En el futuro, los sistemas basados en la arquitectura orientada a servicios podrán estar compuestos de los servicios ofertados por estas empresas en combinación con otros creados por la propia organización o institución. La gran ventaja es que se reparte el coste de desarrollo entre los participantes, y éstos disfrutan y promueven la estandarización tanto en la propia empresa como entre empresas. En particular, el sector vinculado a los viajes dispone de un amplio número de servicios y datos bien definidos y documentados, suficientes para que cualquier desarrollador pueda crear software para agencias de viajes usando únicamente servicios software ya existentes. Otros sectores como el financiero, están realizando grandes progresos en esta dirección.

La arquitectura mencionada se rige sobre un principio fundamental, la orientación al servicio. En este entorno, los servicios pueden ser utilizados sin conocimiento alguno de las características de la plataforma sobre la que se despliegan.

## INTRODUCCIÓN

El acrónimo SOA proviene del inglés Service-Oriented Architecture. Se trata de un modelo de arquitectura que caracteriza el procedimiento para crear y usar los diversos procesos, reunidos en forma de servicios, que configuran un determinado Proceso de Negocio (Un proceso de negocio se puede ver como un conjunto estructurado de tareas, que contribuyen colectivamente a lograr los objetivos de una organización).

Esta arquitectura define y proporciona la infraestructura necesaria para que el intercambio de información y la participación en los procesos de negocio se lleve a cabo con total independencia de la plataforma hardware-software sobre la que trabajan: sistema operativo, lenguaje de programación, características de los equipos, etc.

Y esto se debe a la consideración principal: se trata de alinear los objetivos de negocio con los recursos ofrecidos por las tecnologías de información y comunicación.

Otro aspecto importante es analizar hasta qué punto es posible lograr una reutilización (quizás una de las aspiraciones más antiguas dentro de la comunidad informática), una promesa que apareció con la orientación a objetos y el diseño de componentes y ahora se renueva con la aparición de los “servicios”.

A fin de tener un panorama amplio de lo que realmente significa SOA, es necesario enriquecer nuestros conocimientos con una serie de tecnologías que, aunque disponibles antes de la aparición de esta arquitectura, aparecen ahora usadas en conjunto. Pero también es muy importante que se pueda visualizar el panorama en su conjunto, teniendo en cuenta que es algo más que un conjunto de

tecnologías que la hacen posible; es una nueva forma de visualizar a la empresa como sistema.

Un aspecto clave para lograr esto es asegurar que los procesos y el respaldo de la tecnología permanezcan aislados unos respecto de otros, y es en ese sentido que la arquitectura adquiere toda su importancia. O, intentando parafrasear una frase conocida: “Las tecnologías pasan, pero los procesos quedan”, admitiendo que los procesos tienen una vigencia más larga que las tecnologías.

En este trabajo de investigación abordaremos temas tales como el análisis orientado a servicios, que tiene como objetivo lograr un entendimiento profundo de los requerimientos que se debe satisfacer y proponer para la solución a nivel conceptual, que después las tareas de diseño se volverán en una solución concreta, lista para ser implementada, también expondremos el modelado orientado a servicios que incluye cada uno de los pasos de modelado, el modelo SOE y por último el diseño orientado a servicios en el que encontramos 5 tipos de diseño que cada uno consta de varios pasos para su realización.

**CAPITULO 1**

***LA ARQUITECTURA ORIENTADA A  
SERVICIOS***



**Ilustración Tomada de Referencia 1.**



## **1.1. ANTECEDENTES DE LA ARQUITECTURA ORIENTADA A SERVICIOS.**

En las últimas décadas los departamentos de Tecnologías de la Información<sup>1</sup> (TI) de las empresas han construido una infraestructura que soporta en gran medida la operación de sus empresas y sus clientes. El resultado de este proceso ha sido la creación y mantenimiento de un número considerable de aplicaciones de uso interno, cada una responsable de sus propias tareas.

Los negocios exigen crear aplicaciones cada vez más complejas, en menos tiempo y con menor presupuesto. En muchos casos crear estas aplicaciones requiere de funcionalidades ya antes implementadas como parte de otros sistemas. Ante esta situación los arquitectos de software se enfrentan a dos opciones:

- Tratar de reutilizar la funcionalidad ya implementada en otros sistemas. Una labor difícil de realizar, debido a que estos no fueron diseñados para integrarse o se elaboraron para plataformas y/o tecnologías incompatibles entre ellas.
- Re-implementar la funcionalidad requerida. Aunque implica más tiempo de desarrollo, es en la mayoría de los casos la más fácil y segura.

A pesar de que no sea la más acertada a largo plazo, la segunda opción es la más escogida. Esto trae como resultado:

- Funcionalidad replicada en varias aplicaciones.

---

<sup>1</sup> SENN JAMES A. Information Technology: Principles, Practices, and Opportunities (3rd Edition) (Hardcover - Dec 1, 2003) p 35.

- Dificultad de migración de los sistemas internos, al haber múltiples conexiones desde sistemas que dependen de estos para su funcionamiento.
- Al no haber una estrategia de integración de aplicaciones, se generan múltiples puntos de fallo, que pueden detener la operación de todos los sistemas muy fácilmente.
- Un modelo así, por lo general no escala muy bien.
- El inconveniente final es una pobre respuesta al cambio. Las aplicaciones siguen siendo concebidas desde un principio como islas independientes.

En la arquitectura SOA<sup>2</sup> la funcionalidad deseada se descompone en unidades (servicios) que pueden ser distribuidos en diferentes nodos conectados a través de una red y que, asimismo, son combinados entre sí para alcanzar el resultado deseado. Estos servicios pueden proporcionar datos a otros o llevar a cabo actividades de coordinación entre uno o varios servicios.

Las aplicaciones necesarias para desarrollar los correspondientes procesos de negocio se logran mediante la combinación de colecciones de pequeños módulos llamados servicios. Estos módulos pueden ser empleados por grupos de usuarios provenientes de la propia organización o ajenos a la misma y las nuevas aplicaciones creadas del aprovechamiento de estos servicios muestran mayor flexibilidad y uniformidad. De este modo se consigue un ahorro en el esfuerzo de desarrollo pues se re aprovechan las funcionalidades comunes a las distintas aplicaciones además de favorecer la interacción entre organizaciones dado que se

---

<sup>2</sup> ERL THOMAS, SOA Principles of Service Design (Prentice Hall Service-Oriented Computing Series) p. 38.

logra la homogeneización de la apariencia y del nivel y tipo de datos de entrada para la validación de los usuarios.

En este entorno de trabajo, las unidades básicas son los servicios. Los servicios<sup>3</sup> son unidades de funcionalidad que desarrollan su actividad de forma independiente y que se aproxima al concepto que los humanos asocian a los mismos como puede ser la visualización del estado de una cuenta bancaria, o la emisión de una petición de un tiquete de avión o de tren. En lugar de que los servicios contengan en su código fuente llamadas a otros, se definen protocolos que describen cómo pueden comunicarse entre sí.

Debido a que el término orientado servicio existe desde hace algún tiempo, se ha utilizado en diferentes contextos y con diferentes propósitos. Una constante a través de su existencia ha sido que representa un enfoque distinto para dividir un problema en varias piezas. Lo que esto significa es que la lógica necesaria para resolver un gran problema puede ser mejor construida, llevada a cabo, gestionada si se descompone en una colección de pequeños piezas relacionadas. Cada una de estas piezas se ocupa de una parte específica del problema.

Este enfoque hace que trascienda la tecnología y las soluciones de automatización. Esta es una teoría genérica que puede utilizarse para hacer frente a una variedad de problemas. Lo que distingue al planteamiento orientado a servicios para la separación de las piezas, es la manera en que logra la separación.

Si observamos las ciudades y sus necesidades. Las empresas están llenas de negocios orientados a servicios. Estas empresas realizan aplicaciones orientadas a servicios en la medida en que cada una proporcione un servicio que puede ser

---

<sup>3</sup> ERL THOMAS, SOA Principles of Service Design (Prentice Hall Service-Oriented Computing Series) p. 39.

utilizado por múltiples consumidores. En conjunto, estas empresas abarcan una comunidad de negocios. Este procedimiento tiene sentido para una comunidad de negocios si todos aportan, no para ser desarrollado por una sola empresa proporcionando todos los servicios.

La arquitectura orientada a servicios es un término que representa un modelo<sup>4</sup> en el que la lógica de automatización se descompone en pequeñas, distintas unidades de la lógica. En conjunto, estas unidades abarcan una mayor porción de negocios de automatización de la lógica. Individualmente, estas unidades pueden ser distribuidas.

Incluso si en una comunidad de negocios distribuidos, queremos imponer dependencias entre los servicios, podría inhibir el potencial de las empresas, a pesar de que se desea permitir la interacción y acoplamiento de los demás servicios, se quiere evitar un modelo en el que los servicios tengan una estricta conexión. Al habilitar a las empresas a regir libremente sus servicios individuales, se les permite evolucionar y crecer relativamente independiente unos de otros.

Para conseguir la independencia dentro de nuestro negocio, tenemos que garantizar determinados convenios, por ejemplo, una moneda común para el intercambio de bienes y servicios, un código de construcción que requiere señalización para ajustarse a ciertos parámetros o quizás la exigencia de que todos los empleados hablen el mismo idioma que el nativo de los consumidores. Para obtener la capacidad de ejercer un gobierno autónomo, estos convenios de normalizar los aspectos clave de cada empresa en beneficio de los consumidores se realizan apoyando los negocios individuales.

---

• <sup>4</sup> ERL THOMAS, SOA Principles of Service Design (Prentice Hall Service-Oriented Computing Series) p.43

## **1.2. FUNDAMENTACION TEORICA**

La arquitectura orientada a servicios alienta a las unidades individuales de la lógica a la existencia autónoma, pero no aislados unos de otros. Unidades de la lógica siguen siendo necesarias para ajustarse a una serie de principios que les permiten evolucionar con independencia, sin dejar de mantener una cantidad suficiente de interés común y la normalización. Estas unidades de la lógica se conocen como servicios.

Para mantener su independencia, los servicios encapsulan la lógica dentro de un contexto distinto. Este contexto puede ser específico a una tarea de negocio, una empresa, o alguna otra agrupación lógica.

La cuestión que se aborda para servicios puede ser pequeña o grande. Por lo tanto, el tamaño y el alcance de la lógica representada por los servicios pueden variar. Además, la lógica de servicios puede abarcar la lógica proporcionada por otros servicios. En este caso, uno o más servicios se componen colectivamente.

Por ejemplo, las empresas de soluciones de automatización son típicamente una aplicación de un proceso de negocio. Este proceso se compone de la lógica que ordena la acción de las acciones realizadas por la solución. La lógica se descompone en una serie de pasos que ejecutan en secuencias predefinidas de acuerdo a las normas y condiciones de tiempo de ejecución.

Cuando construimos una solución de automatización de los servicios, cada servicio puede encapsular una tarea realizada por un paso o encapsular una tarea comprimida integrada por un conjunto de pasos. Un servicio puede encapsular la lógica de todo el proceso. En los dos últimos casos, el alcance más amplio representado por los servicios es que puede abarcar la lógica encapsulada por otros servicios.

Para que los servicios puedan utilizar la lógica que encapsulan ellos deben participar en la ejecución de las actividades comerciales. Para ello, deben formar distintas relaciones con aquellos que quieran utilizarlas.

Los servicios pueden ser utilizados por otros programas<sup>5</sup>. A pesar de todo, la relación entre los servicios se basa en el entendimiento para la interacción de los servicios, que debe ser consciente de sí. Esta conciencia se logra mediante el uso de descripciones de servicios.

Una descripción del servicio en su forma más básica, establece el nombre y la ubicación del servicio, así como sus necesidades de intercambio de datos. La manera en que los servicios utilizan las descripciones de servicios resulta de la relación clasificada como débilmente acoplados.

Para los servicios que interaccionan y acoplan algo significativo, deben intercambiar información. Se requiere un marco de comunicaciones capaces de preservar su bajo acoplamiento. Este podría ser un marco de tipo mensajería.

Cuando un servicio envía un mensaje, en su camino se pierde el control de lo que ocurre con el mensaje de entonces. Esa es la razón por la que se requiere que los mensajes existan como unidades independientes de comunicación. Esto significa que los mensajes, como los servicios, deben ser autónomos. A tal efecto, los mensajes pueden equiparse con suficiente inteligencia para auto-regular sus partes de la lógica de procesamiento.

Los servicios proporcionan descripción de servicios y comunicación a través de mensajes utilizando una arquitectura básica. Hasta el momento, esta arquitectura parece similar a las anteriores arquitecturas distribuidas que soportan mensajería

---

<sup>5</sup> Wikipedia®. "Reutilización" [En línea]. [citado: 21-05-2009]. Enciclopedia en línea Wikipedia®. Disponible en Internet: <http://es.wikipedia.org/wiki/Animaci3n>. Última actualización 20-may-2009.

y una separación entre la interfaz y el procesamiento lógico. Lo que la distingue es la forma en que sus tres componentes básicos (servicios, descripciones y mensajes) se han diseñado.

Al igual que la orientación a objetos, la orientación a servicio se ha convertido en un enfoque de diseño que introduce los principios comúnmente aceptados que rigen la ubicación y el diseño arquitectónico de nuestros componentes.

Las aplicaciones de los principios de la orientación a servicios a los procesamientos lógicos, arroja como resultado el procesamiento lógico orientado a servicio. Cuando una solución se compone de unidades de procesamiento lógico orientado a servicio, se convierte en lo que denominamos una solución orientada al servicio.

Cada uno de los principios de orientación a servicios está plenamente explicado más adelante. Con el fin de proporcionar una introducción preliminar, se destacaran algunos de los aspectos clave de estos principios<sup>6</sup> aquí:

- Bajo acoplamiento: mantener una relación de servicios que minimiza las dependencias y sólo exige que se conserve el conocimiento de sí.
- Contrato formal de servicios: los servicios de comunicaciones se adhieren a un acuerdo, tal como se definen colectivamente por una o más descripciones de servicios y los documentos relacionados.
- Autonomía: los servicios tienen control de la lógica que encierran.

---

<sup>6</sup> ERL THOMAS. "Service-Oriented Architecture Concepts, Technology and Design.". 1 ed. [United States] p. 36

- Abstracción: más allá de lo que se describe en el contrato de servicios, los servicios ocultan la lógica del mundo exterior.
- Reusabilidad: la lógica se divide en servicios con la intención de promover la reutilización.
- Capacidad de integración (*Composability*): las colecciones de los servicios pueden ser coordinados y ensamblados para formar servicios compuestos.
- Sin estado (*Statelessness*): reducir al mínimo los servicios de mantenimiento de la información específica de una actividad.
- Capacidad de descubrimiento (*Discoverability*): los servicios están diseñados para ser descriptivos hacia el exterior a fin de que puedan ser encontrados y evaluados a través de los mecanismos disponibles de descubrimiento.

Con un conocimiento de los componentes que conforman la estructura básica y un conjunto de principios de diseño que se pueden utilizar para normalizar la forma y estos componentes, lo único que falta es una plataforma de aplicación que permitirá poner estas piezas juntas para construir la solución para la automatización orientada a servicios. El conjunto de servicios web ofrece la tecnología de dicha plataforma.

Como se emnciono antes, el término "orientado a servicios" y varios modelos de este tipo existían antes de la llegada de los servicios web. Sin embargo, ningún avance tecnológico ha sido tan adecuado y exitoso como los servicios Web.

Todos los principales proveedores de plataformas que actualmente apoyan la creación de soluciones orientadas a servicios, la mayoría lo hacen con el

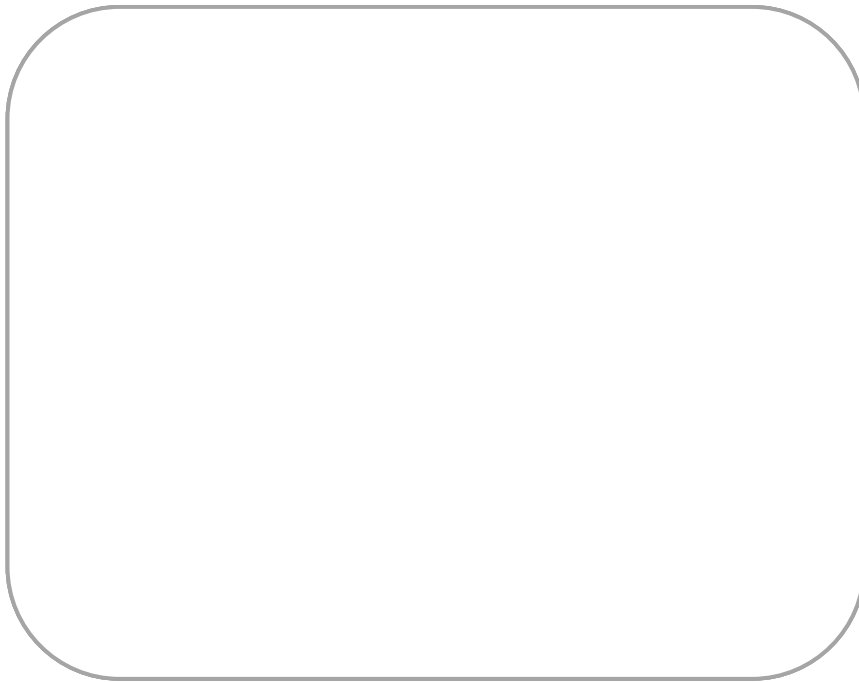


conocimiento de que el apoyo se basa únicamente en el uso de servicios Web. Por lo tanto, es conocido plenamente que el logro de esta arquitectura no requiere los servicios Web, estos son completamente independientes.

Últimamente se ha descrito los distintos componentes de lo que llamamos arquitectura orientada a servicios primitiva. Es etiquetada como tal, ya que representa una tecnología de arquitectura de referencia que cuenta con el apoyo de los principales proveedores de plataformas actuales.

Todas las formas de la arquitectura que se explora de aquí en adelante se basan en ampliar este modelo primitivo. Algunas de las extensiones que discutimos hoy son alcanzables mediante la aplicación de avanzadas técnicas de diseño, mientras que otros dependen de la disponibilidad de pre-definición y especificaciones de servicios web de proveedores de apoyo correspondiente.

Esta arquitectura presenta un modelo de construcción de sistemas distribuidos en el que la funcionalidad demandada será entregada a la aplicación a través de servicios. En la ilustración 1 se muestra el esquema de la arquitectura y los elementos que podrían observarse. El esquema se encuentra dividido en 2 zonas; una que abarca el ámbito funcional de la arquitectura y otra vinculada a la calidad de servicio.



**Ilustración 1. Esquema de una Arquitectura orientada a servicios.**

A continuación se describen brevemente los elementos de la arquitectura:

**FUNCIONES:**

- Transporte: es el mecanismo utilizado para llevar las demandas de servicio desde un consumidor de servicio hacia un proveedor de servicio, y las respuestas desde el proveedor hacia el consumidor.
- Protocolo de comunicación de servicios: es un mecanismo acordado a través del cual un proveedor de servicios y un consumidor de servicios comunican qué está siendo solicitado y qué está siendo respondido.
- Descripción de servicio: es un esquema acordado para describir qué es el servicio, cómo debe invocarse, y qué datos requiere el servicio para invocarse con éxito.
- Servicio: describe un servicio actual que está disponible para utilizar.

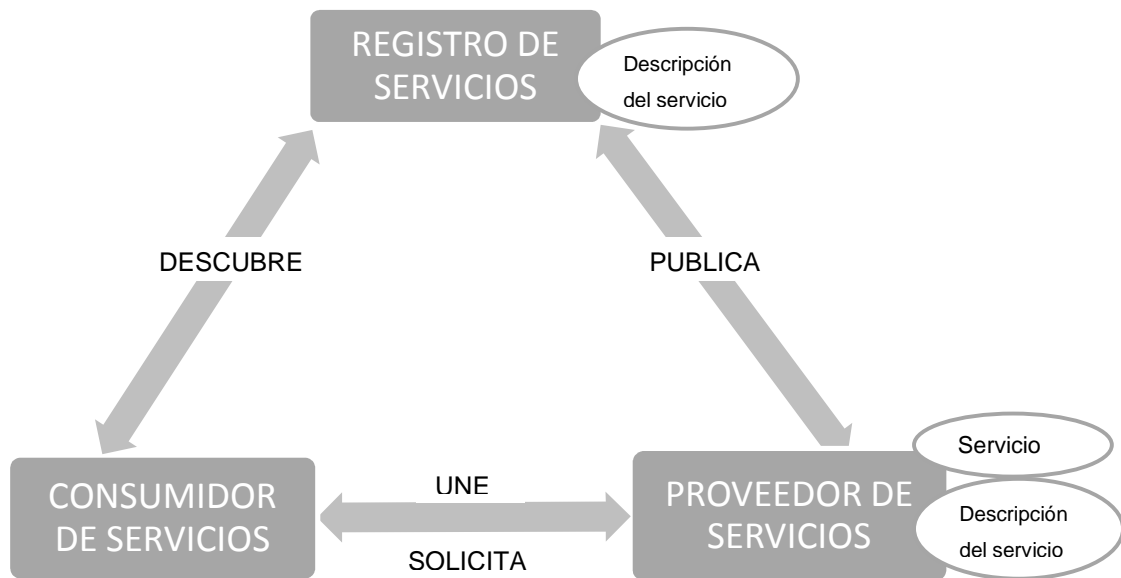
- **Procesos de Negocio:** es una colección de servicios, invocados en una secuencia particular con un conjunto específico de reglas, para satisfacer un requisito de negocio.
- **Registro de Servicios:** es un repositorio de descripciones de servicios y datos que pueden utilizar los proveedores de servicios para publicar sus servicios, así como los consumidores de servicios para descubrir o hallar servicios disponibles.

#### CALIDAD DE SERVICIO:

- **Política:** es un conjunto de condiciones o reglas bajo las cuales un proveedor de servicio hace el servicio disponible para consumidores.
- **Seguridad:** es un conjunto de reglas que pueden aplicarse para la identificación, autorización y control de acceso a consumidores de servicios.
- **Transacciones:** es el conjunto de atributos que podrían aplicarse a un grupo de servicios para entregar un resultado consistente.
- **Administración:** es el conjunto de atributos que podrían aplicarse para manejar los servicios proporcionados o consumidos.

Las colaboraciones en esta arquitectura siguen el paradigma de descubrir, unir y solicitar, donde un consumidor de servicios realiza la localización dinámica de un servicio consultando el registro de servicios para hallar uno que cumpla con un determinado criterio. Si el servicio existe, el registro proporciona al consumidor la interfaz de contrato y la dirección del servicio proveedor.

El siguiente diagrama (Ilustración 2) muestra las entidades (roles, operaciones y artefactos) en una arquitectura orientada a servicios donde éstas colaboran.



**Ilustración 2. Relación entre las entidades.**

Cada entidad puede tomar uno de los tres roles posibles correspondientes a consumidor, proveedor y/o registro:

- Un consumidor de servicios es una aplicación, un módulo de software u otro servicio que demanda la funcionalidad proporcionada por un servicio, y la ejecuta de acuerdo a un contrato de interfaz.
- Un proveedor de servicios es una entidad accesible a través de la red que acepta y ejecuta consultas de consumidores, y publica sus servicios y su contrato de interfaces en el registro de servicios para que el consumidor de servicios pueda descubrir y acceder al servicio.
- Un registro de servicios es el encargado de hacer posible el descubrimiento de servicios, conteniendo un repositorio de servicios disponibles y permitiendo

visualizar las interfaces de los proveedores de servicios a los consumidores interesados.

Las operaciones que pueden llevar a cabo las entidades son:

- **Publicar:** Para poder acceder a un servicio se debe publicar su descripción para que un consumidor pueda descubrirlo y solicitarlo.
- **Descubrir:** Un consumidor de servicios localiza un servicio que cumpla con un cierto criterio consultando el registro de servicios.
- **Une y solicita:** Una vez obtenida la descripción de un servicio por parte de un consumidor, éste lo solicita haciendo uso de la información presente en la descripción del servicio.

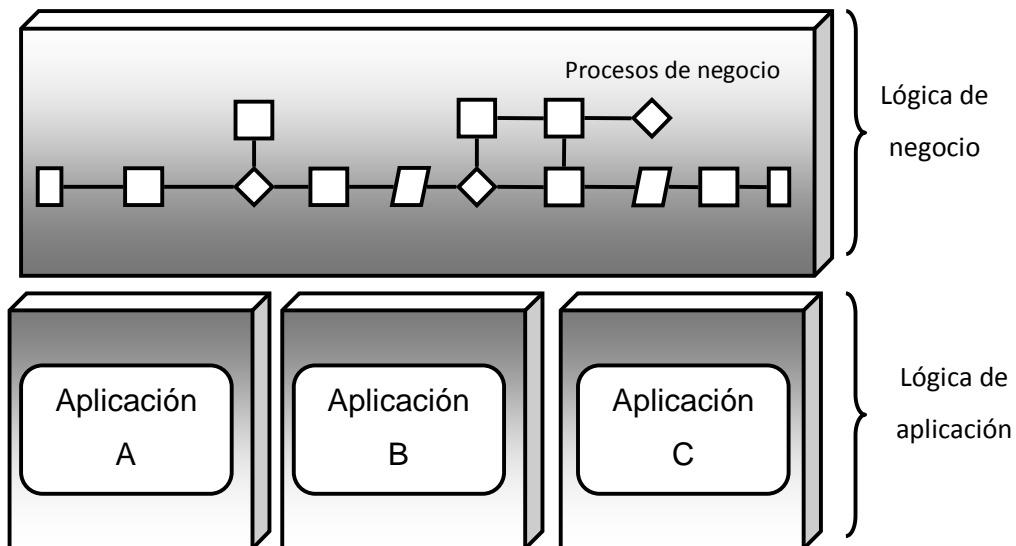
Finalmente, los artefactos en una arquitectura orientada a servicios son:

- **Servicio:** Un servicio que está disponible para el uso a través de una interfaz publicada y que permite ser solicitado por un consumidor de servicios.
- **Descripción de servicio:** Una descripción de servicio específica la forma en que un consumidor de servicio interactuará con el proveedor de servicio, especificando el formato de consultas y respuestas desde el servicio. Esta descripción también puede especificar el conjunto de precondiciones, pos-condiciones y/o niveles de calidad de servicio.

### **1.3. PRINCIPIOS DE LA ORIENTACION A SERVICIOS**

La lógica colectiva que define y maneja una empresa es una entidad evolutiva que constantemente cambia en respuesta a influencias externas e internas. Desde

una perspectiva TI (Tecnologías de la Información)<sup>7</sup>, esta lógica empresarial puede ser dividida en 2 mitades: lógica de negocio y lógica de aplicación, como se observa en la ilustración 3.



**Ilustración 3. Dominios de la lógica de aplicación y de negocio**

Cada una existe en su propio mundo y representan una parte necesaria de la estructura organizacional contemporánea. La lógica de negocio es una implementación documentada de los requerimientos de negocio que se originan desde las áreas de negocio de una empresa. La lógica de negocio esta generalmente estructurada en procesos que expresan estos requerimientos, junto con cualquier restricción asociada, dependencia e influencia externa.

La lógica de aplicación es una implementación automatizada de la lógica de negocio organizada en diversas soluciones tecnológicas. La lógica de aplicación expresa flujos de trabajo de los procesos de negocio a través de sistemas

<sup>7</sup> WOODS DAN, MATTERN THOMAS, Enterprise SOA: Designing IT for Business Innovation (Paperback - April 28, 2006) p 379.

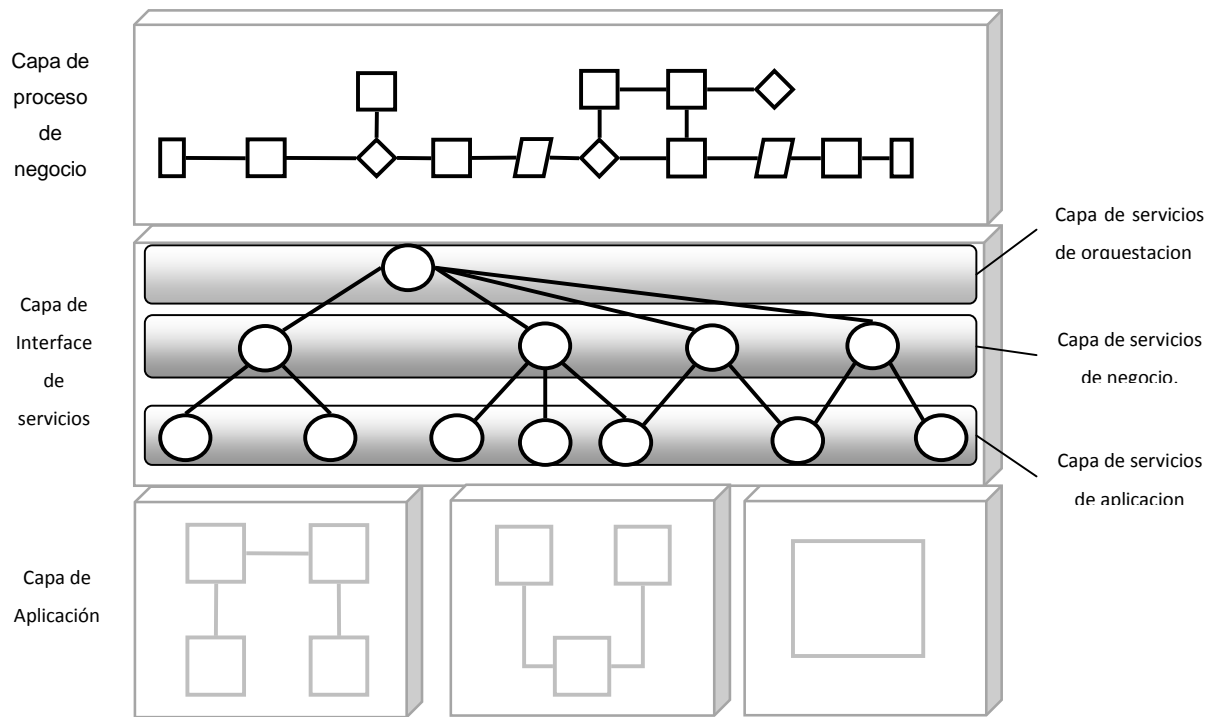
comprados o desarrollados a la medida dentro de los confines de la infraestructura TI de una organización.

La orientación a servicios se aplica a la lógica de empresa. Esta introduce nuevos conceptos que amplían la manera en la cual esta lógica es representada, vista, modelada, y compartida.

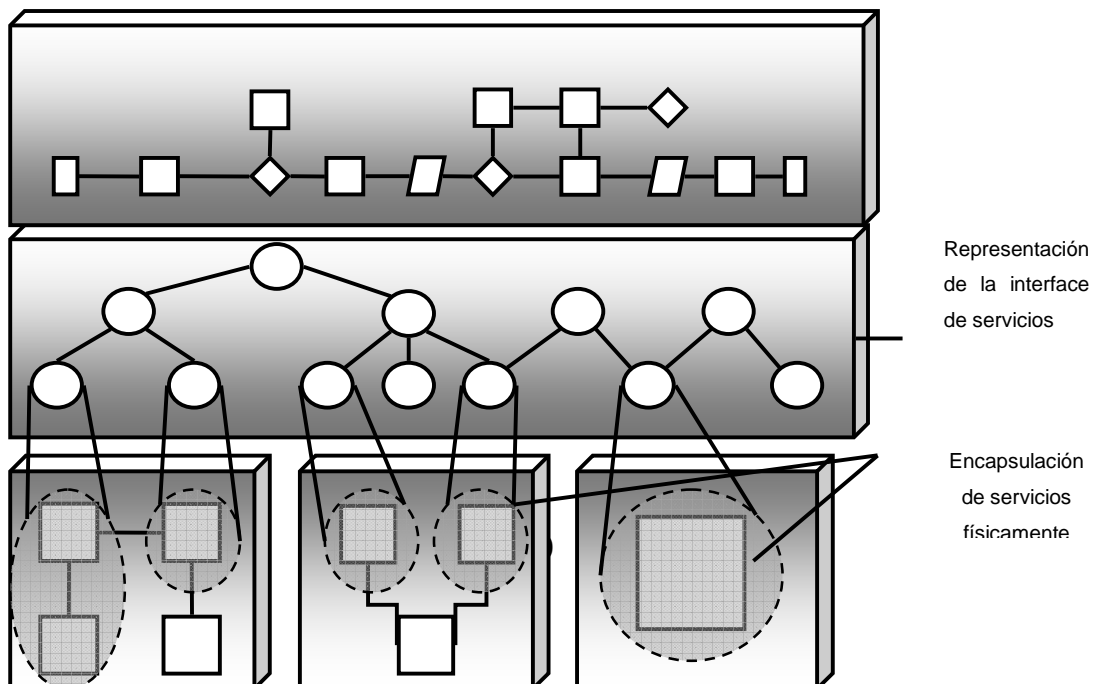
Los conceptos introducidos por la orientación a servicios son materializados a través de la introducción de servicios. Los servicios proveen un alto nivel de abstracción localizado entre las capas de negocio y aplicación. En ese punto, los servicios pueden encapsular lógica de aplicación física así como también lógica de procesos de negocio.

Los servicios modularizan la empresa, formando unidades de lógica independientes que existen dentro de una capa de conectividad común. Los servicios pueden ser distribuidos en capas de tal manera que servicios padre puedan encapsular servicios hijos. Esto permite que la capa de servicios pueda estar compuesta por múltiples capas de abstracción.

A continuación se puede apreciar la ilustración 4 que nos muestra las capas de servicio y como están constituidas y la ilustración 5 donde se plasma el funcionamiento de la interface de servicio.



**Ilustración 4 Capas de Servicio**





## Ilustración 5 Interface de servicios

### 1.3.1. Componentes de una SOA

Una arquitectura orientada a servicios se compone básicamente de cuatro partes, cada una de las cuales poseen una unidad de medida diferente:

- Mensajes (unidades de comunicación): representan los datos requeridos para completar alguna o todas las partes de una unidad de trabajo.
- Operaciones (unidades de trabajo): representa la lógica requerida para procesar los mensajes necesarios para completar una unidad de trabajo. como se muestra en la ilustración 6.

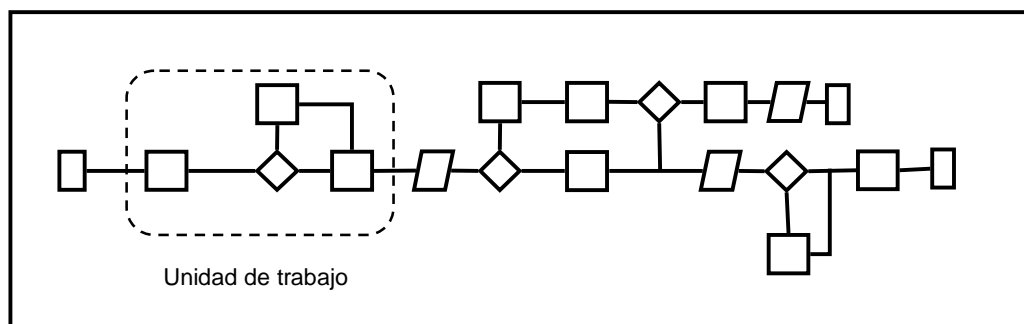
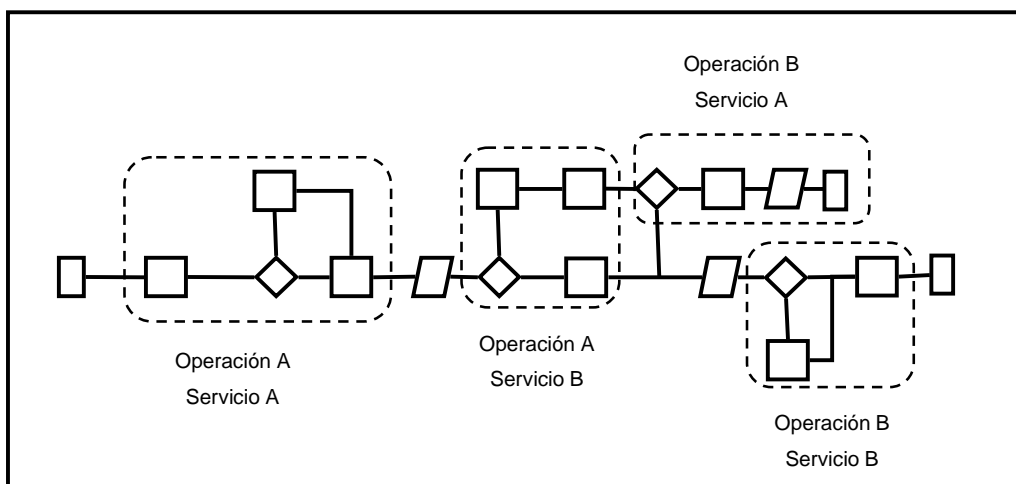


Ilustración 6 Unidad de Trabajo

- Servicios (unidades de procesamiento lógico): representan un grupo de operaciones lógicamente agrupadas capaces de realizar ciertas unidades de trabajo relacionadas.

- Procesos (unidades de lógica de automatización): contienen las reglas de negocio que determinan cuales operaciones del servicio son usadas para completar una unidad de automatización. En otras palabras, un proceso representa una gran porción de trabajo que requiere de la finalización de pequeñas unidades de trabajo. De acuerdo a la ilustración 7 que se plasma a continuación.

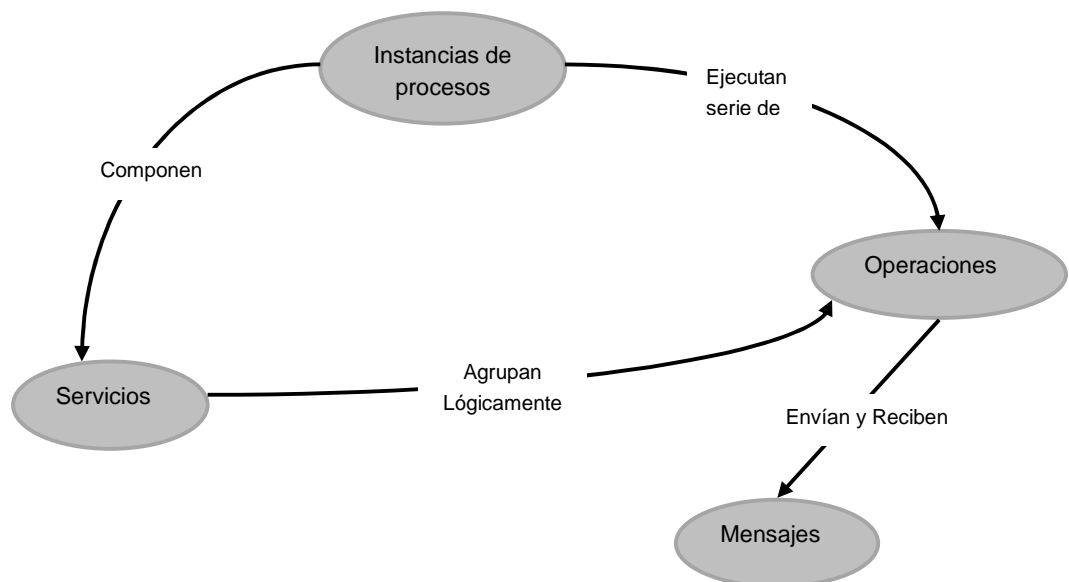


**Ilustración 7 Operaciones de servicio**

Interrelacionen de acuerdo a lo siguiente y visualice la ilustración 8 en la cual se interrelacionan los componentes y la ilustración 9 donde se definen los componentes

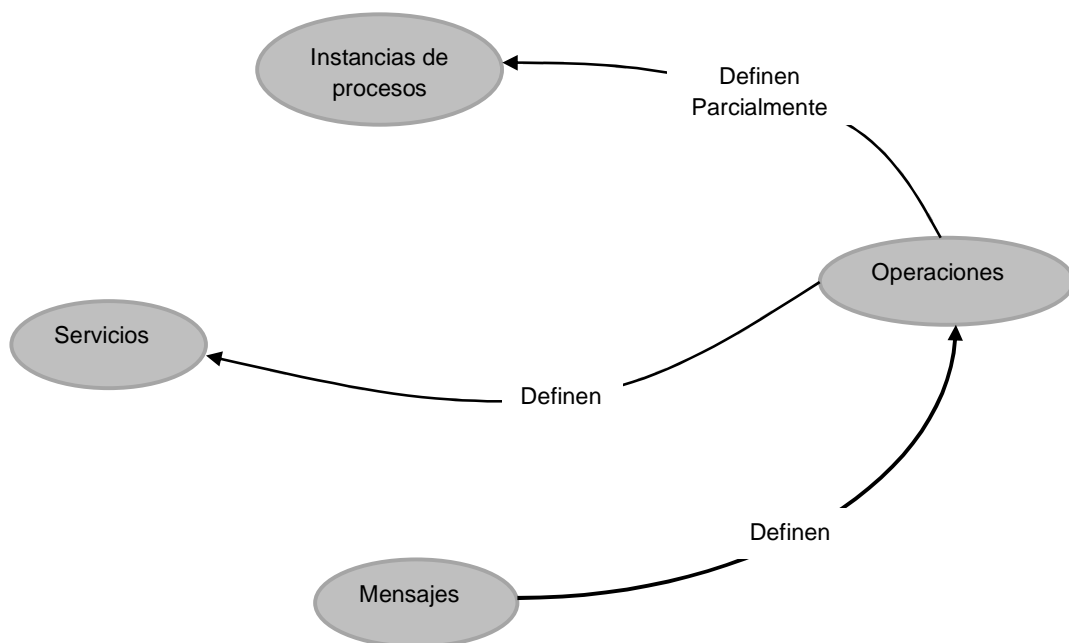
- Una operación envía y recibe mensajes para realizar trabajo.
- Una operación es definida en su mayoría por los mensajes que esta procesa.
- Un servicio agrupa una colección de operaciones relacionadas.

- Un servicio es definido en su mayoría por las operaciones que este comprende.
- Una instancia de proceso puede componer servicios.
- Una instancia de proceso no está necesariamente definida por sus servicios porque esta podría solamente requerir un subconjunto de la funcionalidad ofrecida por los servicios.
- Una instancia de proceso invoca a una serie única de operaciones para completar su automatización.
- Cada instancia de proceso está parcialmente definida por las operaciones de servicio que esta usa.



**Ilustración 8 Interrelación de componentes**

Una arquitectura orientada a servicios es un entorno estandarizado acorde a los principios de la orientación a servicios en el cual un proceso que usa servicios (un servicio orientado a procesos) puede ejecutarse.



**Ilustración 9 Definición de componentes**

Los principios sobre los cuales se basa la orientación a servicios son los siguientes:

- *Los servicios son reusables:* sin importar si la oportunidad de reutilización es inmediata o no, los servicios son diseñados para soportar reutilización potencial.
- *Los servicios comparten un contrato formal:* para que los servicios puedan interactuar, no necesitan compartir nada excepto un contrato formal que describa cada servicio y defina los términos de intercambio de información.
- *Los servicios están débilmente acoplados:* los servicios deben estar diseñados para interactuar sin la necesidad de fuertes dependencias con otros servicios.
- *Los servicios abstraen la lógica subyacente:* la única parte de un servicio que es visible al mundo exterior es la que está expuesta vía el contrato de servicio. La lógica subyacente, más allá de lo que este expresado en las descripciones que comprenden el contrato, es invisible e irrelevante a los solicitantes del servicio.
- *Los servicios pueden componerse de otros servicios:* esto permite a la lógica ser representada en diferentes niveles de granularidad y promueve la reutilización y la creación de capas de abstracción.
- *Los servicios son autónomos:* la lógica gobernada por un servicio reside dentro de unos límites específicos. El servicio tiene control dentro de este límite y no depende de otros servicios para ejecutar su gobierno (sobre la lógica).
- *Los servicios no tienen estado:* los servicios no deberían requerir administrar información de estado, ya que esto podría impedir su habilidad de permanecer débilmente acoplados. Los servicios deberían ser diseñados para no tener estado incluso si esto significa tener que delegar la administración de la información de estado a otra parte.

- *Los servicios son descubribles (exhibición de servicios):* los servicios deberían permitir que sus descripciones sean descubiertas y entendidas por humanos y solicitantes de servicios que podrían ser capaces de hacer uso de su lógica.

#### **1.4. CAPAS DE SERVICIO**

En un modelo empresarial la capa de interfaz de servicio se encuentra entre el proceso de negocios y la capa de aplicación, Aquí es donde reside la conectividad del servicio y por lo tanto este es el área de la empresa en donde las características de la arquitectura son los más prevalentes. Para implementar las características que acabamos de identificar de manera efectiva, debemos abordar algunas preguntas.

- ¿Con que lógica se deberían presentar los servicios?
- ¿Cómo podrían los servicios relacionarse con la lógica existente?
- ¿Cómo los servicios pueden representar mejor la lógica de los procesos de negocio?
- ¿Cómo los servicios pueden ser construidos y a su vez posicionados para promover agilidad?

El elemento común a todas las respuestas, parece ser la última de las cuatro características de la arquitectura: las capas de abstracción.

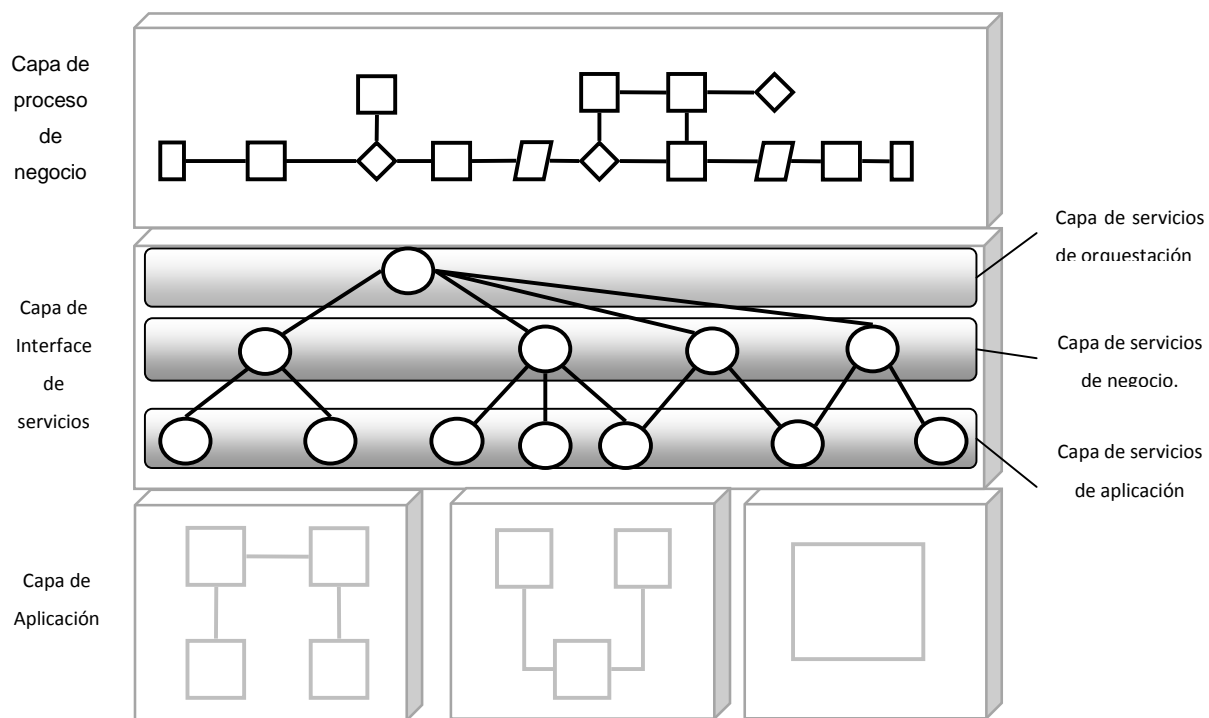
Se ha establecido cómo, aprovechando el concepto de composición, podemos construir las capas de servicios especializadas. Cada una de estas capas puede

abstraer un aspecto específico de la solución, abordando una de las preguntas identificadas. Esto nos alivia de tener que construir un servicio que adapte negocios, aplicaciones y agilidad a la vez.

Las tres capas de abstracción que hemos identificado son las siguientes:

- Capa de servicios de aplicación
- Capa de servicios de negocio
- Capa de servicios de orquestación

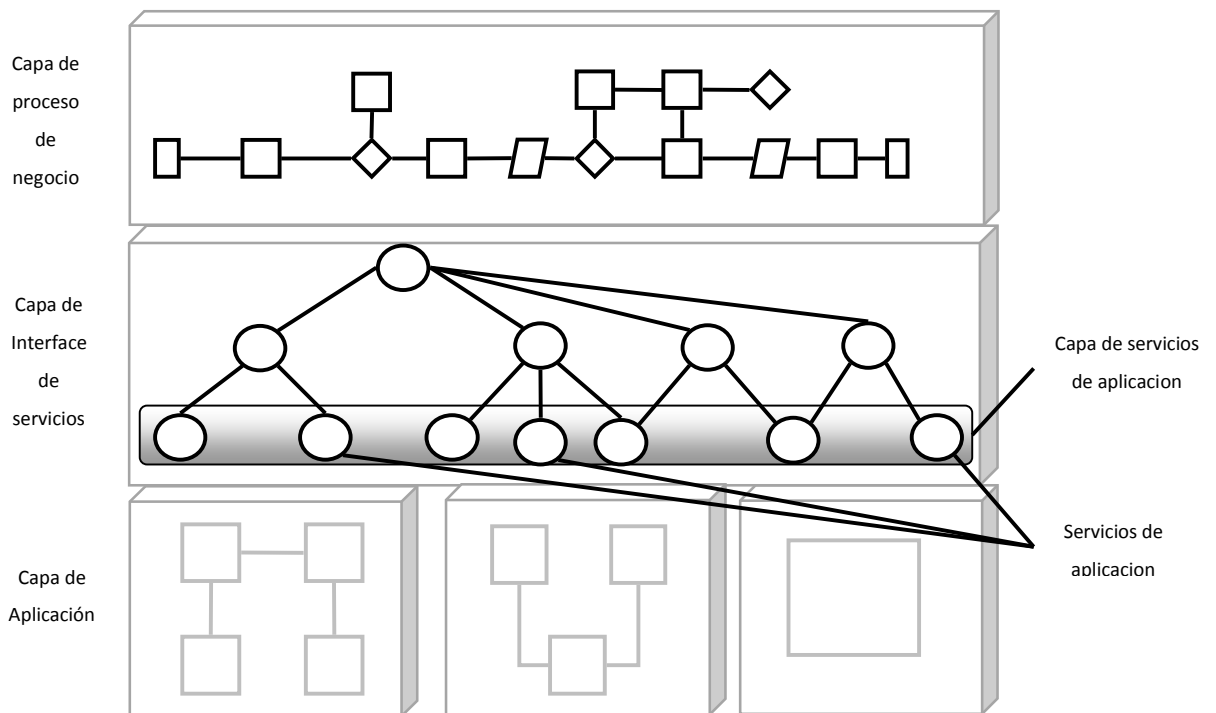
Cada una de estas capas se muestra en la imagen a continuación (ilustración 10).



**Ilustración 10 Capas de servicio detalladas**

### 1.3.2. Capa de servicios de aplicación

La capa de servicios de aplicación<sup>8</sup> establece el inicio del nivel de grupo que existe al expresar la funcionalidad de la tecnología específica. Los Servicio que residen dentro de esta capa pueden ser referidos a un simple servicio de aplicación. Su propósito es proporcionar funciones reutilizables relacionadas con el procesamiento de los datos dentro de los entornos nuevos o primitivos de aplicaciones, véase ilustración 11.



**Ilustración 11 Capa de servicios de aplicación**

<sup>8</sup> ERL THOMAS. "Service-Oriented Architecture Concepts, Technology and Design.". 1 ed. [United States] p 337



Los servicios de aplicación común tienen las siguientes características:

- Estas exponen funcionalidad dentro de un contexto de procesamiento específico.
- Se basan en los recursos disponibles dentro de una determinada plataforma.
- Son la solución agnóstica.
- Son genéricas y reutilizables.
- Pueden ser usadas para lograr punto a punto la integración con otros servicios de aplicación.
- A menudo son incompatibles en términos de la interfaz de granularidad que exponga.
- pueden consistir en la mezcla de costumbres de desarrollo de servicios y de servicios de terceros que han sido adquiridos o arrendados.

Ejemplos típicos de los modelos de servicios ejecutados como servicios de aplicación son las siguientes:

- Servicios utilitarios
- Servicio de encapsulación

Cuando una capa de servicios empresariales existe (como se explica en la sección de la capa de servicios empresariales), existe una fuerte motivación para activar todos los servicios de aplicación dentro de los servicios utilitarios generales. De esta manera se aplica una solución de forma agnóstica, proporcionando operaciones reutilizables que pueden estar compuestas por los servicios de

negocio satisfaciendo los requerimientos de procesamiento del centro de negocios.

Por otra parte, si la lógica de negocio no reside en una capa separada, los servicios de aplicación pueden ser necesarios para la implementación de modelos de servicios más relacionados con la capa de servicios de negocio. Por ejemplo, un servicio de aplicación también puede clasificarse como un servicio de negocio si se relaciona directamente con la aplicación lógica y contiene las reglas de negocio incorporadas.

Servicios que contienen lógica de aplicación y de negocios pueden ser denominados servicios de aplicación híbridos o solo servicios híbridos. Este modelo de servicio se encuentra comúnmente en las arquitecturas distribuidas tradicionales. No se recomienda el diseño en la construcción de capas de abstracción de servicio.

Por último, un servicio de aplicación también puede componer, otros más pequeños servicios de aplicación dentro de una unidad mayor de lógica de aplicación.

Los servicios de aplicación que existen únicamente para permitir la integración entre los sistemas a menudo se conocen como servicios de aplicación de integración o simplemente servicios de integración. Los Servicios de integración a menudo se implementan como controladores.

Debido a que son comunes los residentes de la capa de servicios de aplicación, ahora es un buen momento para introducir el modelo de servicio de encapsulación. Los servicios de encapsulación con mayor frecuencia se utilizan para fines de integración. Estos consisten en servicios que encapsulan, algunas o todas las partes de un entorno heredado para exponer herencias funcionales a

servicios a solicitantes. La forma más frecuente de los servicios de encapsulación es utilizando un adaptador de servicio proporcionado por las herencias de los proveedores.

### **1.3.3. Capa de servicios de negocio**

Si bien los servicios de aplicación son responsables de la representación de la tecnología y la lógica de aplicación, la capa de servicios de negocio<sup>9</sup> introduce un servicio que se trate sólo de representar la lógica de negocio (Ilustración 12), denominada servicios de negocio.

Los servicios de negocios son la sangre vital de la Arquitectura Orientada a servicios contemporánea. Ellos son responsables de la expresión de la lógica de negocio a través de servicios de orientación y llevar la representación de modelos de negocio corporativo en el ámbito de servicios Web.

Los servicios de aplicación se dividen en diferentes tipos de modelos de servicio, simplemente porque las categorías representan un grupo de servicio que expresan la funcionalidad de una tecnología específica. Por lo tanto, una aplicación de servicios puede ser un servicio utilitario, un servicio de encapsulación, u otro servicio.

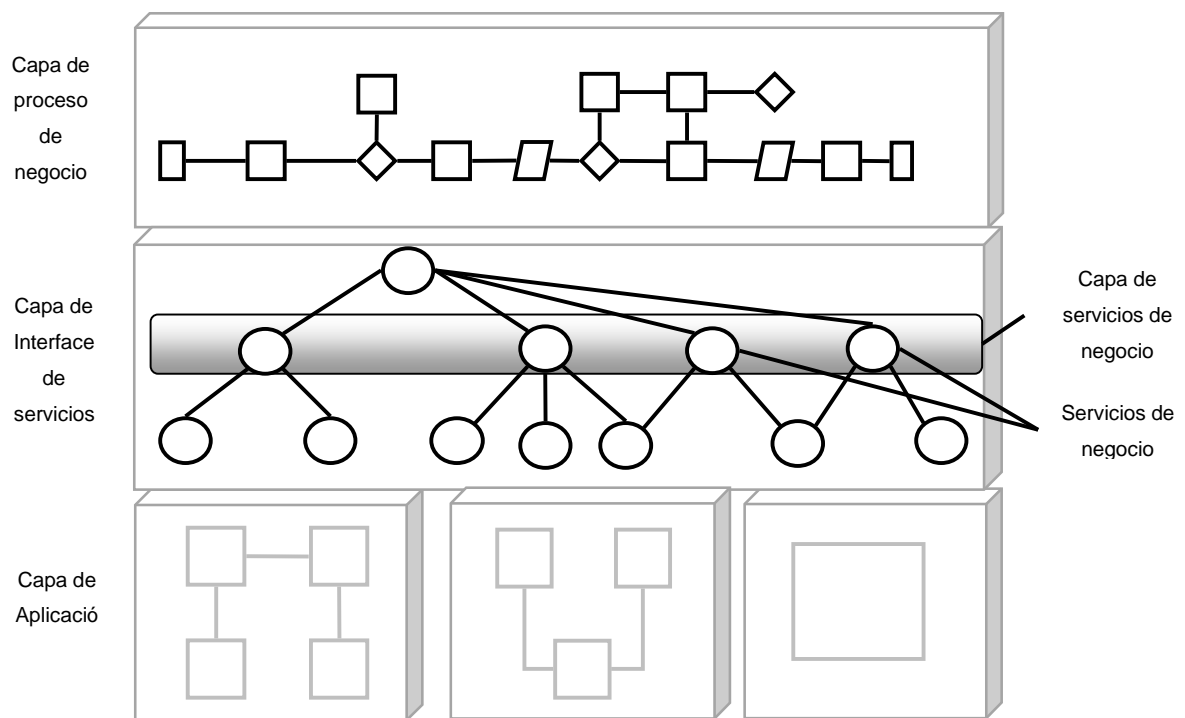
Los servicios de negocios, por otro lado, son siempre una implementación del modelo de servicios de negocio. El único propósito de los servicios de negocios destinados a la capa de servicios de negocio es representar la lógica de negocio en la forma más pura posible.

---

<sup>9</sup> ERL THOMAS. "Service-Oriented Architecture Concepts, Technology and Design.". 1 ed. [United States] p 341

Esto, sin embargo, no les impide implementar otros modelos de servicios. Por ejemplo, los servicios de negocio también pueden ser clasificados como servicios de control y servicios utilitarios.

De hecho, cuando la aplicación lógica es extraída de una capa de servicios de aplicación, es más que probable que los servicios de negocio actuarán como controladores de la composición de servicios de aplicaciones disponibles para la ejecución de su lógica de negocio.



**Ilustración 12 Capa de servicios de negocio**

La capa de abstracción de servicios de negocio conduce a la creación de dos nuevos modelos de servicio de negocio:

- Servicios de negocio centrados en tareas: es un servicio que encapsula la lógica de negocio específica de una tarea o proceso de negocio. Este tipo de servicio en general es necesario cuando la lógica de los procesos de negocio no está centralizada, como parte de una capa de orquestación. Los servicios de negocio centrados en tareas han limitado el potencial de reutilización.
- Servicios de negocio centrados en entidades: es un servicio que encapsula una entidad de negocio (por ejemplo, una factura). Los servicios de negocio centrados en entidades son muy útiles para crear procesos de negocio reutilizables y servicios agnósticos que están compuestos por una capa de orquestación o por una capa de servicio consistente en el servicio de negocio centrado en tareas (o ambos).

Cuando una capa de servicios de aplicación existe, estos dos tipos de servicios de negocio pueden ser posicionados para componer servicios de aplicación para llevar a cabo su lógica de negocio.

#### **1.3.4. Capa de servicios de orquestación**

La Orquestación<sup>10</sup> es más valiosa que un proceso de negocio, ya que nos permite vincular directamente la lógica de proceso de servicio a la interacción dentro del flujo de trabajo de la lógica. Este combina el modelado de procesos de negocio con el modelado y diseño orientado a servicio. Y, porque los lenguajes de orquestación (como WS-BPEL) realizan una gestión de flujo de trabajo a través de un modelo de procesos de negocio, la orquestación lleva a los procesos de

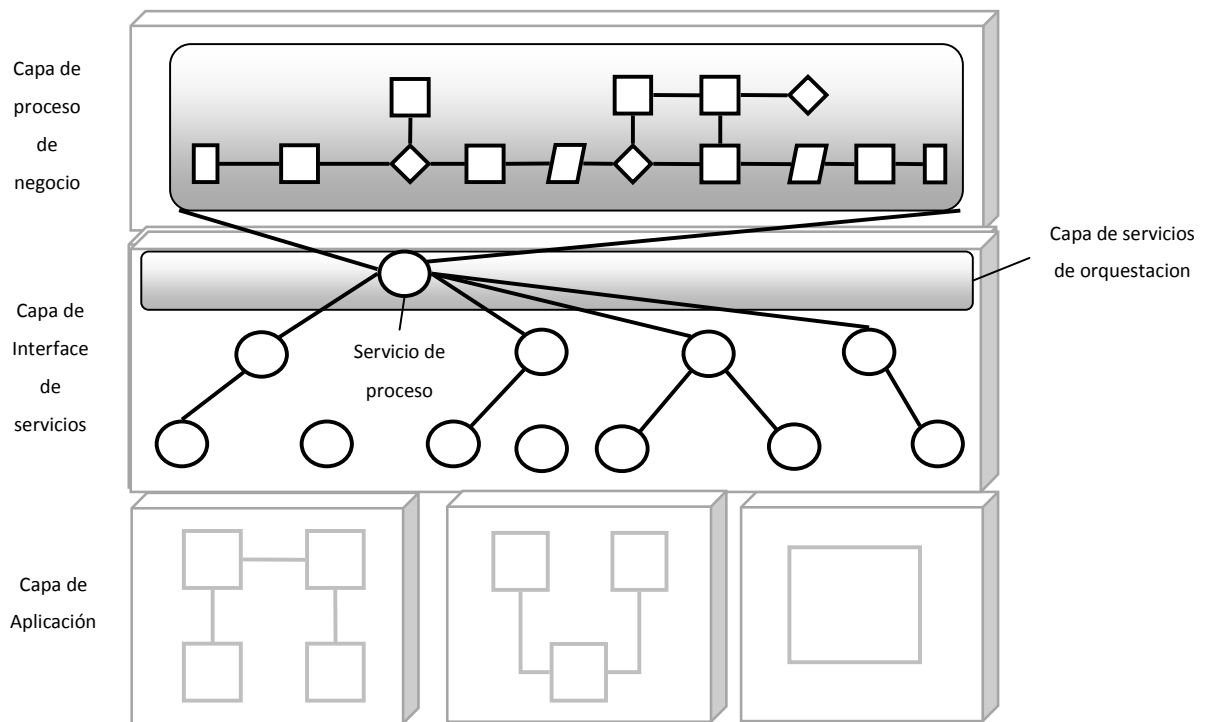
---

<sup>10</sup> ERL THOMAS. "Service-Oriented Architecture Concepts, Technology and Design.". 1 ed. [United States] p 344.

negocio dentro de la capa de servicio, posicionándolo como un controlador de composición maestro.

La capa de servicios de orquestación introduce un nivel de abstracción que alivia la necesidad de otros servicios para la gestión de los detalles de interacción requeridos para garantizar que las operaciones de los servicios se ejecuten en una secuencia específica. Dentro de la capa de servicios de orquestación, los procesos de servicios componen otros servicios que ofrecen conjuntos de funciones específicas, independientes de las reglas de negocio y escenarios específicos de lógica necesaria para ejecutar un proceso de instancia.

Este proceso de servicios es el mismo para los que hemos definido el modelo de proceso de servicio. Por lo tanto, el proceso de servicios también es servicios de control por su propia naturaleza, ya que son necesarios para componer otros servicios para la ejecución de la lógica de procesos de negocio. Los procesos de servicios también tienen el potencial de convertirse en servicios utilitarios en la medida, en que un proceso en su totalidad, sea considerado reutilizable. En este caso, un proceso que permite la orquestación de servicios puede el mismo ser orquestado, así como se encuentra en la ilustración 13.



**Ilustración 13 Capa de servicios de orquestación**

La introducción de una capa de orquestación normalmente trae consigo la exigencia de introducir nuevas middleware en la infraestructura de TI. Los servidores de orquestación no son de ninguna manera triviales y además pueden imponer importantes gastos y complejidad.

## CAPITULO 2

# ***ANALISIS ORIENTADO A SERVICIOS***



Ilustración Tomada de Referencia 4.



## **2.1. INTRODUCCION AL ANALISIS**

El proceso para determinar las necesidades de la automatización de los negocios, puede ser representado a través del análisis orientado a servicios.

El objetivo del análisis orientado a servicios es abordar las siguientes preguntas:

- ¿Cuales servicios necesitan ser construidos?
- ¿Qué lógica debería ser encapsulada en cada servicio?

En la medida en que estas preguntas son contestadas se relaciona directamente con la cantidad de esfuerzo invertido en el análisis. En concreto, la determinación de los niveles de servicio para construir y la forma de abordar su entrega, son fundamentales los puntos de decisión que acaban formando la estructura de todo entorno orientado al servicio.

Los objetivos generales de la realización de un análisis orientado a servicios son los siguientes:

- Definir un conjunto preliminar de manejo del servicio candidato.
- agrupar el manejo de servicios de candidatos en contexto lógico. Estos servicios representan contextos candidatos.
- Definir los límites preliminares de servicio para que no superpongan los servicios existentes o servicios previstos.
- Identificar la lógica encapsulada con el potencial de reutilización.
- Asegurarse de que el contexto de la lógica encapsulada, sea apropiado para su uso.
- Definir la composición de cualquier conocimiento preliminar de los modelos.

La introducción de un nuevo proceso de análisis en un entorno de TI puede ser una cosa difícil. Cada organización ha desarrollado su propio enfoque para el análisis de negocio de automatización de los problemas y sus soluciones. Años de esfuerzo y de documentación ya se han invertido en procesos y modelado bien establecidos de los resultados finales. El proceso descrito no está destinado a sustituir los procedimientos existentes. En lugar de ello, se propone una secuencia de medidas suplementarias, específicas para la prestación de soluciones orientadas a servicios.

El análisis orientado a servicios<sup>11</sup> se puede aplicar a diferentes niveles, dependiendo de cuál de los estilos de estrategias de la arquitectura se use para producir servicios, la estrategia elegida determinará las capas de abstracción que componen el entorno de solución para un servicio de capas.

Desde una perspectiva de análisis, cada nivel tiene diferentes requisitos de modelado. Por ejemplo, la naturaleza de los análisis necesarios para definir los servicios de aplicación es diferente de lo que se necesita para modelar la capa de servicios empresariales.

Por lo tanto, como se ha mencionado anteriormente, un requisito fundamental de este proceso es la elección del estilo de estrategia. Otras preguntas que deben responderse antes de proceder con el análisis orientado a servicios son las siguientes:

- ¿Qué trabajo es necesario para establecer el modelo y la ontología de negocio?
- ¿Qué herramientas de modelado se utilizarán para llevar a cabo el análisis?
- ¿El análisis parte de un plan de transición de SOA?

---

<sup>11</sup> ERL THOMAS. "Service-Oriented Architecture Concepts, Technology and Design." 1 ed. [United States] p. 377

La respuesta a esta última pregunta a menudo dependerá de alcance del proyecto. Este análisis puede ser una etapa prevista en un plan más amplio que se traza con la transición hacia una organización amplia de esta arquitectura. En proyectos más pequeños, el análisis orientado a servicios a si mismo puede incorporar un paso para la planificación de la transición.

## 2.2. PASOS PARA REALIZAR UN ANALISIS ORIENTADO A SERVICIOS

El proceso del análisis orientado a servicios es un sub-proceso del estilo del ciclo de vida de la arquitectura orientada a servicios. Los pasos indicados en la siguiente figura (Ilustración 14) son tareas comunes asociadas a esta fase y se describen con más detalle a continuación:

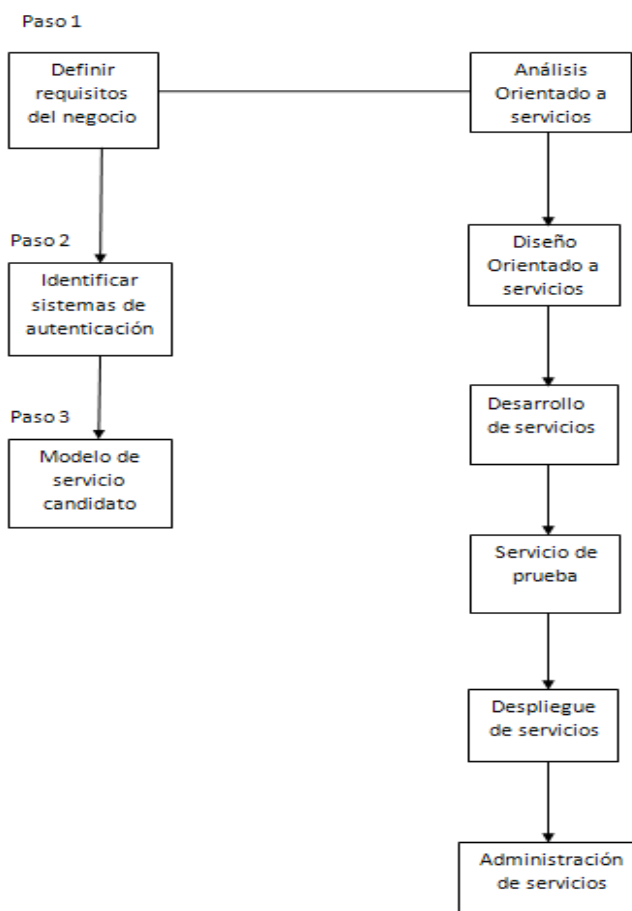


Ilustración 14 Pasos para realizar un análisis orientado a servicios

### **2.2.1. Paso 1: definir las necesidades de automatización de negocios.**

A través de cualquier medio, los requerimientos<sup>12</sup> del negocio son recogidos normalmente, su documentación se requiere para este proceso de análisis para empezar. Dado que el alcance de nuestro análisis se centra en la creación de servicios de apoyo a la solución orientada al servicio, sólo los requisitos relacionados con el ámbito de aplicación de la solución deben ser considerados.

Los Requerimientos del negocio deben ser lo suficientemente maduros para que un alto nivel de automatización de procesos pueda ser definido. Este proceso de documentación de negocio se utilizará como punto de partida del proceso de modelado de servicio que se describe en el paso 3.

### **2.2.2. Paso 2: Identificar los sistemas de automatización.**

Existen aplicaciones que ya son lógicas, en la medida en que la automatización<sup>13</sup> de alguno de los requisitos señalados en el paso 1 haya sido identificada. Mientras que un análisis orientado a servicios no podría determinar con exactitud como los servicios Web podrían encapsular o sustituir la herencia lógica de aplicación, que hace que nos ayude en el alcance de potencial de sistemas afectados.

Los detalles de cómo los servicios Web se refieren a los sistemas existentes son subsanadas en la fase de diseño orientada a servicios.

Por ahora, esta información será usada para ayudar a identificar servicios candidato de aplicación durante el proceso de modelado de servicio que se describen en el paso 3. Este paso es más orientado a apoyar los esfuerzos de modelado de mayor escala de las soluciones orientadas a los servicios.

---

<sup>12</sup> ERL THOMAS. "Service-Oriented Architecture Concepts, Technology and Design." 1 ed. [United States] p.379

<sup>13</sup> ERL THOMAS. "Service-Oriented Architecture Concepts, Technology and Design." 1 ed. [United States] p.380

La comprensión de los entornos heredados afectados sigue siendo útil con una cantidad menor de modelado de servicios, pero una gran cantidad de esfuerzo de investigación no sería necesario en este caso.

### **2.2.3. Paso 3: Modelo de servicios candidato.**

El análisis orientado a servicios introduce el concepto de modelado de servicio que es un proceso en donde se identifica la operación de servicios candidatos<sup>14</sup> y, a continuación, son agrupados en un contexto lógico.

Estos grupos eventualmente toman la forma de un servicio candidato, estos son montados en un modelo provisional compuesto, que representa la combinación lógica de la planificación de aplicaciones orientadas a servicios.

## **2.3. BENEFICIOS DE UNA SOA CENTRADA EN NEGOCIOS.**

La llegada de los servicios web ha difundido la importancia de la apertura de los marcos de las comunicaciones. Como resultado, la mayoría de los profesionales de TI se encuentran en una etapa en la que reconocen la importancia de las tecnologías de servicios Web<sup>15</sup>.

Como hemos establecido anteriormente, la mayoría de los servicios Web que actualmente se está construyendo, son más o menos, una mezcla de aplicación y servicios de negocios. Estos tipos de servicios híbridos son atractivos porque, con un mínimo de esfuerzo, cumplen con los requisitos inmediatos y con claro retorno de la inversión.

---

<sup>14</sup> ERL THOMAS. "Service-Oriented Architecture Concepts, Technology and Design." 1 ed. [United States] p. 381

<sup>15</sup> DAN WOODS, THOMAS MATTERN Enterprise SOA: Designing IT for Business Innovation p. 321

La proliferación de servicios híbrido es el resultado de un enfoque de abajo hacia arriba de modo que se ha convertido en algo común. Que de forma inmediata para todas las anteriores aplicaciones de arquitectura tomen parte de los marcos de servicios de comunicación en la Web abierta.

Los servicios de negocios, por otro lado, a menudo necesitan alguna justificación. Muchos todavía se resisten o desconocen los beneficios de los principios de la introducción orientada a servicios en el ámbito de análisis de negocios.

Es fácil hacer caso omiso del modelado de negocios orientado a servicios y simplemente centrarse en la orientación a servicios en lo que se refiere a la tecnología y la arquitectura técnica. La lógica común de este enfoque es que, independientemente de que los procesos de negocios<sup>16</sup> deban ser automatizados, también pueden dividirse en servicios Web, según sea necesario.

Muchas de las características contemporáneas de la arquitectura, todavía se pueden alcanzar sin el uso de los servicios de negocio. Los beneficios aparentes son superficiales. Hay muy buenas razones para tomarse el tiempo para construir el modelo y los servicios empresariales.

En esta sección se lista una serie de beneficios para incorporar la orientación de servicios al nivel de proceso de negocios. La orientación a servicios lleva a que la estructura de los modelos de procesos de negocios puedan mejorar significativamente la flexibilidad y la agilidad con la que los procesos pueden ser remodelados en respuesta a los cambios. Cuando están debidamente diseñados, los servicios de negocios pueden establecer una gran capacidad de respuesta en la información de los entornos de tecnologías; sensible a los cambios en áreas de organización de negocio que pueden ser eficiente mediante la recomposición de

---

<sup>16</sup> MICHAEL ROSEN, BORIS LUBLINSKY, KEVIN T. SMITH, MARC J. BALCER. Applied SOA: Service-Oriented Architecture and Design Strategies. p.86

un proceso de negocio y su apoyo a la arquitectura de su tecnología (según lo expresado por la capa de servicios de aplicación).

Como impulso de negocio, a menudo hay restricciones del mundo real (infraestructura, limitaciones de seguridad, limitaciones presupuestarias), que requieren que la tecnología en algunos casos retroceda. Esto puede desplazar la carga de la adaptación sobre los modelos de procesos de negocio. Este tipo de requerimiento de agilidad puede satisfacerse por medio de la capa de servicios de negocio, ya que permite a los servicios de negocio adaptarse a los cambios requeridos para el entorno de una organización técnica.

En otras palabras, la aplicación de capa de abstracción de servicio de negocio y tecnología termina estableciendo la posibilidad de una empresa para alcanzar una de las dos vías.

### **2.3.1. Preparación de negocios para un proceso de orquestación.**

El entorno de orientación a servicios se trasladara a base de orquestación en algún momento, cada vez es más importante estar preparados para esta transición. La Orquestación trae consigo conceptos que, cuando se aplique, se encuentran en el núcleo de la arquitectura . Por lo tanto, los actuales procesos de modelado, eventualmente pueden lograr más fácil migrar a un entorno de impulso de orquestación recomendado.



### **2.3.2. Servicio de negocio posibilitan la reutilización.**

La creación de capas de servicios de negocios, promueve la reutilización<sup>17</sup> de la capa de negocios y servicios de aplicación:

- Al modelar la lógica de negocio como parte de servicios con límites explícitos, los procesos de negocio a nivel de reutilización se pueden lograr. La lógica de subprocesos o incluso los procesos completos pueden ser reutilizados, como parte de la lógica de procesos o como parte de complejos procesos (que se traduce en una composición de servicios en su propio derecho).
- Al tomar el tiempo para ajustar adecuadamente la representación de servicios de negocios, el resultado de la capa final de los servicios de negocios es liberar a toda la capa de servicios de aplicación de asumir tareas o actividades específicas de las funciones de procesamiento. Esto permite a los servicios de aplicación posicionarse para convertirse en puros, los servicios reutilizables que facilitan servicios de negocios a través de los límites de solución.

### **2.3.3. Los servicios de negocios pueden realizar la orientación a servicios empresarial.**

El modelado de los servicios de negocio une los principios de orientación de servicios con el modelo de organización de negocio. La perspectiva resultante puede expresar claramente cómo los servicios se relacionan e incorporan el cumplimiento de los requisitos de negocio.

---

<sup>17</sup> ERIC A. MARKS, MICHAEL BELL. Service-Oriented Architecture (SOA): A Planning and Implementation Guide for Business and Technology p. 211

Las aplicaciones de Servicios de negocios dan fuerzas a una organización para ver y reinterpretar conocimientos de negocios en una forma orientada al servicio. Alterar la perspectiva de cómo los procesos de negocio pueden estar estructurados, divididos y modelados es un paso esencial para lograr un entorno en el que la orientación a servicios este normalizado, en constante éxito y comúnmente natural.

Aunque la capa de servicios de negocios puede representar con exactitud un modelo de servicio corporativo a aplicar, se convertirá en obsoleto, una vez se hayan revisado los requisitos empresariales emergentes. En la medida en que se mantiene lineamientos en relación con el estado actual de los modelos de negocio, que seguirá sirviendo como una valiosa vista de la empresa valioso, ya que no existen en abstracto, sino de la forma operativo e implementado .

## ***2.4 DERIVACION DE LOS SERVICIOS DE NEGOCIO.***

Cuando la definición de SOA por parte de la industria no existía y los principios de la orientación a servicios no estaban estandarizados globalmente, tampoco estaban estandarizados los medios de modelado del servicio de negocio. Como con todos los demás aspectos, hay muchas opiniones, y aunque muchos tienen ideas, pocas metodologías concretas han surgido. Existe un grupo seleccionado de enfoques y algunos de estos son más aceptados que otros.

No debería haber un único enfoque para la derivación de servicios, no es inusual encontrar un modelo de negocio detrás de una típica empresa que ha sufrido millones de revisiones, tomando forma a lo largo de años para la adaptación al entorno de negocio, correspondiente a una organización.

Las organizaciones emplean diferentes metodologías, relaciones entidad negocio y vocabularios, resultan divergentes para la estructura de modelo de negocios.

Complementario a esto hay preferencias culturales e influencias de distribuidores de plataformas que resultan de los modelos de negocio a través de diferentes conjuntos de modelado de herramientas e idiomas. El punto de equilibrio se encuentra cuando cada modelo de negocios es único.

#### **2.4.1. Fuentes de donde los servicios de negocio pueden ser derivados.**

El funcionamiento interno de cada organización, independientemente de la estructura o del tamaño, puede descomponerse dentro de un conjunto de servicios de negocio. Ya que un servicio de negocio simple representa una unidad lógica de trabajo y casi ninguna organización se compone de unidades de trabajo.

Lo que difiere, sin embargo es como las organizaciones construyen y documentan el trabajo que realizan. Al inicio de esta sección hicimos hincapié en el hecho de que cada entorno corporativo es único en cuanto a la forma y tamaño de sus modelos de negocios y como los implementan y los mantienen. Por consiguiente, corresponde a la orientación de servicios determinar cuál es el mejor mapa lógico existente para los servicios.

A continuación encontraremos algunos enfoques de análisis de negocios comunes utilizados por muchas Empresas.

#### **2.4.1.1 Modelos de administración de procesos de negocio (BPM models).**

La llegada de BPM a la industria, ha traído como resultado el modelado de procesos y la remodelación de actividades, BPM<sup>18</sup> es el conjunto de servicios y herramientas que facilitan la administración de procesos de negocio.

Por administración de procesos entendemos: análisis, definición, ejecución, monitoreo, y control de los procesos. Este modelo además contempla soporte para la interacción humana, e integración de aplicaciones, y es aquí la diferencia fundamental con la tecnología de WorkFlow existente, que es que el modelo integra en los flujos, a los sistemas. Las soluciones del tipo WorkFlow solo se limitaban a definir el flujo de actividades humanas, o de documentos, y con esto obtener el seguimiento de los procesos, pero en estos casos si un participante del proceso requería como parte de sus actividades ingresar datos en una aplicación, entonces debía salir del ambiente del WorkFlow, levantar la aplicación, y luego de terminada su operación volver al WorkFlow y registrar el cambio de estado, o termino de la actividad. En el modelo mencionado anteriormente todo está integrado en el mismo flujo lo que es más natural para un participante, el completa su actividad dentro del flujo, y tras bambalinas se actualizan los sistemas que se tengan que actualizar. Es decir, los procesos de negocio encapsulan los servicios, los integran y se trabaja en un mismo flujo, como se ve en la Ilustración 15.

En la practica un flujo BPM o modelo de proceso BPM, visualmente es muy parecido a un WorkFlow, la diferencia está en que uno puede notar que ciertas actividades son realizadas por personas, y otras son actividades sistematizadas (realizadas por sistemas), y ambas aparecen en el flujo.

---

<sup>18</sup> JAMES P. LAWLER , H. HOWELL-BARBER Service-Oriented Architecture: SOA Strategy, Methodology, and Technology p.4

El otro “valor agregado” del modelo es que ofrece una solución completa, que abarca todo el ciclo de vida de un proceso de negocio: análisis, modelamiento, ejecución y monitoreo de los procesos.

En BPM<sup>19</sup> el modelo del proceso se convierte en el núcleo de la implementación del proceso como solución tecnológica. El modelo del proceso de negocio (su diseño), que realiza el área de negocios de una empresa, es “en si” lo que se ejecuta sobre el “servidor de procesos” (el motor de BPM). Dicho en otras palabras: la “lógica de negocio” principal que antes bajo las tecnología tradicional se debía programar, y colocar sobre un “servidor de aplicaciones” (tradicional), ahora se reemplaza por un modelo que se sube al “servidor de procesos” con mucho menos intervención del área de TI (menos programación).

En la practica una buena solución debería poder ejecutar un proceso modelado por el área de negocio, sin la necesidad de que TI tenga que programar una sola línea de código, y obtener como solución algo equivalente a un WorkFlow Tradicional (sin integración de sistemas). Luego el área de TI debería tomar este “workflow”, e implementarle los formularios de entrada (de interacción con usuarios), y los “servicios” (las actividades automatizadas) para completarlo en un flujo BPM.

Hacer que un modelo se convierta en un proceso ejecutable requiere de varias tecnologías habilitantes (enabling tools), cuando estas tecnologías se proveen juntas se le llama BPMS, las principales son:

- **Motores de Orquestación:** permiten coordinar la secuencia de actividades según los flujos y reglas del modelo de procesos.

---

<sup>19</sup> MICHAEL ROSEN, BORIS LUBLINSKY, KEVIN T. SMITH, MARC J. BALCER. Applied SOA: Service-Oriented Architecture and Design Strategies. p.22

- **Herramientas de Análisis y Business Intelligence:** permiten analizar la información producto de la ejecución del proceso en tiempo real.
- **Motores de Reglas:** (Rule Engines) ejecuta reglas que permiten abstraer las políticas y decisiones de negocio de las aplicaciones subyacentes.
- **Repositorios:** mantiene los componentes y recursos de los procesos (definiciones, modelos, reglas, etc. ) disponibles para su reutilización en múltiples procesos
- **Herramientas de Simulación y Optimización:** permite a los administradores del negocio, comparar los nuevos diseño de procesos con el desempeño operacional actual.
- **Herramientas de Integración:** permiten integrar el modelo con otros sistemas, con los sistemas legados de la empresa.

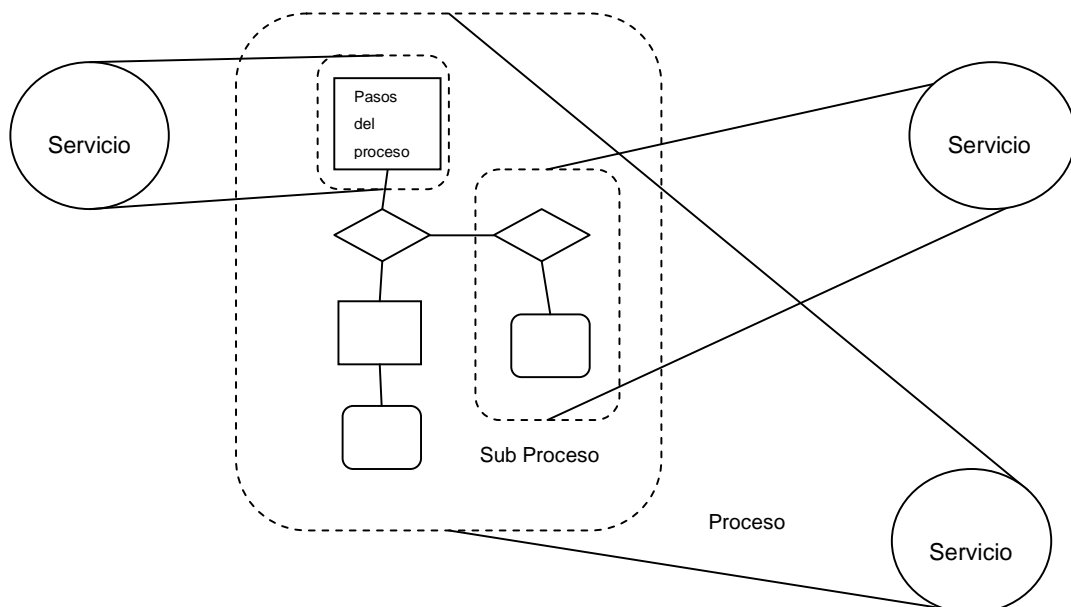


Ilustración 15. Partes de un proceso que pueden ser encapsuladas por un servicio de negocio.

A continuación se crea una empresa para uso de este trabajo llamada ABC y por medio de casos de estudio se mostrara a lo largo de este, como se aplican los procesos que se encuentren especificados en los capitulos.

#### CASO DE ESTUDIO

Se describen dos procesos de la empresa ficticia ABC que actualmente han sido automatizados implementando servicios. Nos enfocaremos en la interacción de los servicios de presentación de factura y completamiento de la orden.

Primero exploraremos el proceso de presentación de factura, este consiste en una serie de pasos que describen como debe ser generada la factura específicamente:

1. El empleado de contabilidad crea y emite una factura electrónica utilizando el sistema de contabilidad.
2. El evento de guardado desencadena una secuencia de comandos que exporta una copia electrónica de la factura a un directorio de red.
3. Un componente desarrollado a la medida, el cual revisa este cada diez minutos, toma el documento y lo transforma en un documento XML.
4. El documento XML de la factura luego es validado, si se considera valido es enviado al servicio de presentación de factura, si la validación falla, el documento es rechazado y el proceso termina.
5. Dependiendo de cuándo fue la última revisión de metadatos el servicio podría emitir una solicitud de recibo de metadatos al sistema de contabilidad (sistema heredado).

6. Si la solicitud de recibo de metadatos es emitida y si esta determina que ningún cambio fue hecho a las descripciones relevantes del servicio de contabilidad, el servicio de presentación de facturas transmite la factura a l sistema de contabilidad usando la garantía de entrega *ExactlyOnce* (mecanismo que asegura que el documento solo será entregado una sola vez, debido a su unicidad inherente). Si la solicitud de recibo de metadatos identifica un cambio en las descripciones del servicio de contabilidad, la factura no es presentada y el proceso termina.

El proceso interno de completamiento de la orden es similar en el sentido que este establece el mismo tipo de relación entre el sistema contable y un servicio Web, solo que en este caso el flujo de datos es revertido, como se describe a continuación:

1. El Servicio de Completamiento de Orden recibe un mensaje SOAP desde el sistema de contabilidad, el cual consiste en un documento de orden de compra.
2. El servicio valida el documento entrante. Si es valido, el documento es pasado a un componente personalizado, sino, un mensaje de notificación de rechazo es enviado al sistema de contabilidad y el proceso termina.
3. El componente transforma el documento XML en una orden de compra que se ajusta al formato nativo del sistema de contabilidad.
4. La orden de compra es introducida al sistema contable mediante su extensión de importación.
5. La orden de compra termina en la cola de trabajo de un empleado de contabilidad, quien entonces procesa el documento.



### 2.4.1.2 Modelos entidad

Las entidades primarias representan el menú de documentos y áreas de operación de una empresa. Por ejemplo, factura, orden de compra, cliente y área de reclamo son entidades con diferentes tipos de negocio. Complementario a esto, la entidad de modelos de organización acuerdan con el propietario las reglas y las políticas de negocio.

Los servicios centrados en entidades son un espejo de los modelos entidad, por contener un conjunto de operaciones genéricas que facilitan diferentes tipos de funciones asociadas al procesamiento de la entidad. La comunicación entre diferentes servicios centrados en entidades puede ser regido por, limitar a la relación de inherencia entre entidades.

#### CASO DE ESTUDIO

ABC ha decidido construir servicios centrados en torno a tareas específicas, por esto los modelos entidad no son usados como fuentes para la derivación de servicios de negocio. Sin embargo, para demostrar el concepto de entidades veremos cuales entidades se encuentran dentro de las áreas de negocio de ABC.

Las entidades de mayor relevancia son:

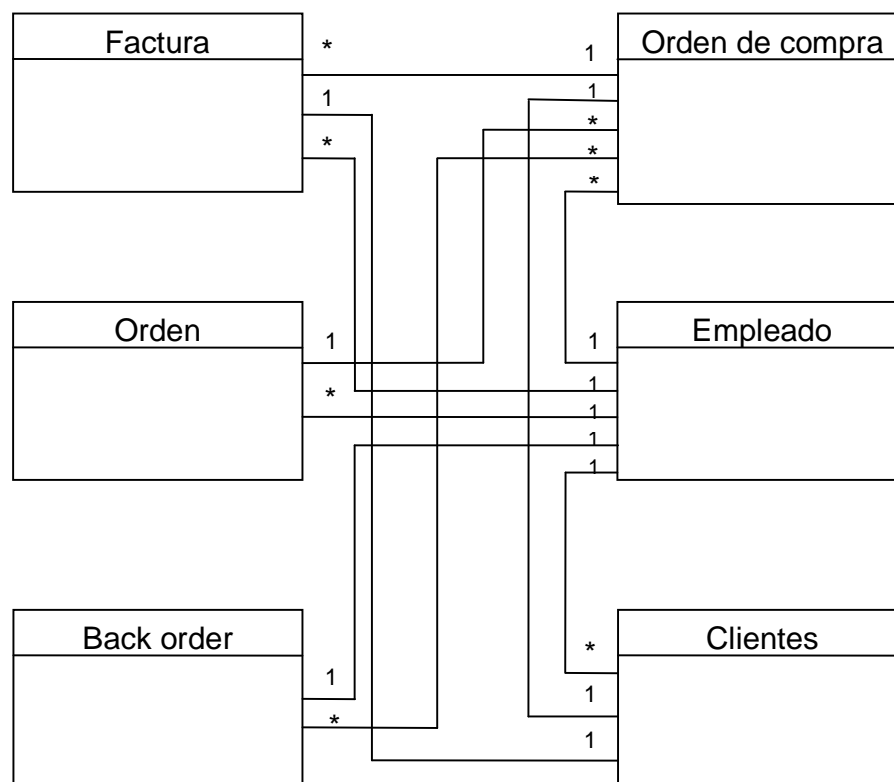
- Factura
- Orden de compra

Algunas entidades involucradas con estos procesos:

- Empleado
- Orden
- Orden incompleta
- Cliente

A continuación veremos algunos ejemplos de cómo las entidades descritas anteriormente se relacionan una a otra, véase ilustración 15.

- Una Orden de Compra puede estar relacionada a cero o muchas facturas.
- Una Orden de Compra solo puede estar relacionada a un único cliente.
- Una Orden puede estar relacionada a una o muchas Órdenes de compra.
- Una Orden incompleta puede estar relacionada a una o más Órdenes de compra.
- Una Factura solo puede estar relacionada un único cliente.
- Un Empleado puede estar relacionado a cero o muchas facturas, Órdenes de compra, Ordenes, Ordenes incompletas, o Clientes.



**Ilustración 17. Relacion entre las entidades.**

## **2.4.2. Tipos de derivación de servicios de negocio.**

Al derivar servicios de dos fuentes, podemos identificar diversos resultados en la creación de distintos tipos de servicios de negocios.

### **2.4.2.1 Servicio de negocio centrado en tareas.**

Este servicio Web ha sido modelado para acomodar procesos de negocio específico. Las operaciones son agrupadas de acuerdo a su relevancia para ejecutar la tarea para el apoyo de los procesos. Típicos ejemplos de servicio de negocio centrado en tareas:

- Verificar factura.
- Obtener historia de reporte.

Cada uno de estos servicios contiene operaciones relacionadas a una tarea en particular que se encuentran dentro de un proceso. El servicio centrado en tareas<sup>20</sup> usualmente resulta del modelado de ejercicios que están centrados en los requerimientos de negocio de una reunión inmediata. Las fuentes incluyen modelos de casos de uso y definiciones de procesos BPM. Mientras menos se requiera esfuerzo de análisis, este tipo de servicio de negocio tiene limitado el potencial de reusabilidad. El modelamiento reusable del servicio de negocio centrado en tareas a menudo requiere de múltiples casos de uso y modelos de proceso de negocio analizados antes del actual modelamiento de los servicios.

---

<sup>20</sup> ERL THOMAS. "Service-Oriented Architecture Concepts, Technology and Design." 1 ed. P.393

### CASO DE ESTUDIO

Aunque son híbridos en diseño (aplicación + negocios) los siguientes servicios de ABC sigue un modelo centrado en tareas:

- Servicio de presentación de factura.
- Servicio de completamiento de orden.

#### **2.4.2.2 Servicio de negocio centrado en entidades.**

Generalmente son producidos como parte de un análisis a largo plazo con el esfuerzo de alinear servicios de negocios con los modelos de negocio corporativos existentes. Su naturaleza genérica inherente hace que sean sumamente reutilizables por numerosos procesos de negocio. Aunque el servicio de negocio centrado en entidades<sup>21</sup> a menudo se construye como parte de las aplicaciones de proyectos de desarrollo centrados en un proceso de negocio en particular, se diferencian de los servicios centrado en tareas en que ellos no proporcionan una interface específica para ese proceso, la fuente de inspiración de este tipo de servicio son los modelos entidad.

Cuando es comparado servicio centrado en tareas, servicio centrado en entidades, se incrementa significativamente la agilidad con que el proceso de orientación a servicios puede ser remodelado. Este es el porqué de que el servicio centrado en tareas a menudo es construido para ayudar a automatizar un proceso de negocio, por lo tanto pueden atarse a ese proceso.

---

<sup>21</sup> ERL THOMAS. "Service-Oriented Architecture Concepts, Technology and Design." 1 ed. P.394

Cuando la lógica del proceso cambia, el contexto en el que los servicios se utilizan y componen puede cambiar también, esto puede invalidar las operaciones de servicios agrupados originalmente y se podría esperar un rediseño y una reorganización.

Los servicios centrados en entidades requieren un análisis más avanzado, incrementando el costo de cada servicio y el tiempo que se requiere para producirlo. Adicional a esto, pueden ser genéricos, por lo tanto su uso se vuelve dependiente de un controlador de negocio padre, tal como el proceso del controlador de servicios centrados en tareas.

#### CASO DE ESTUDIO

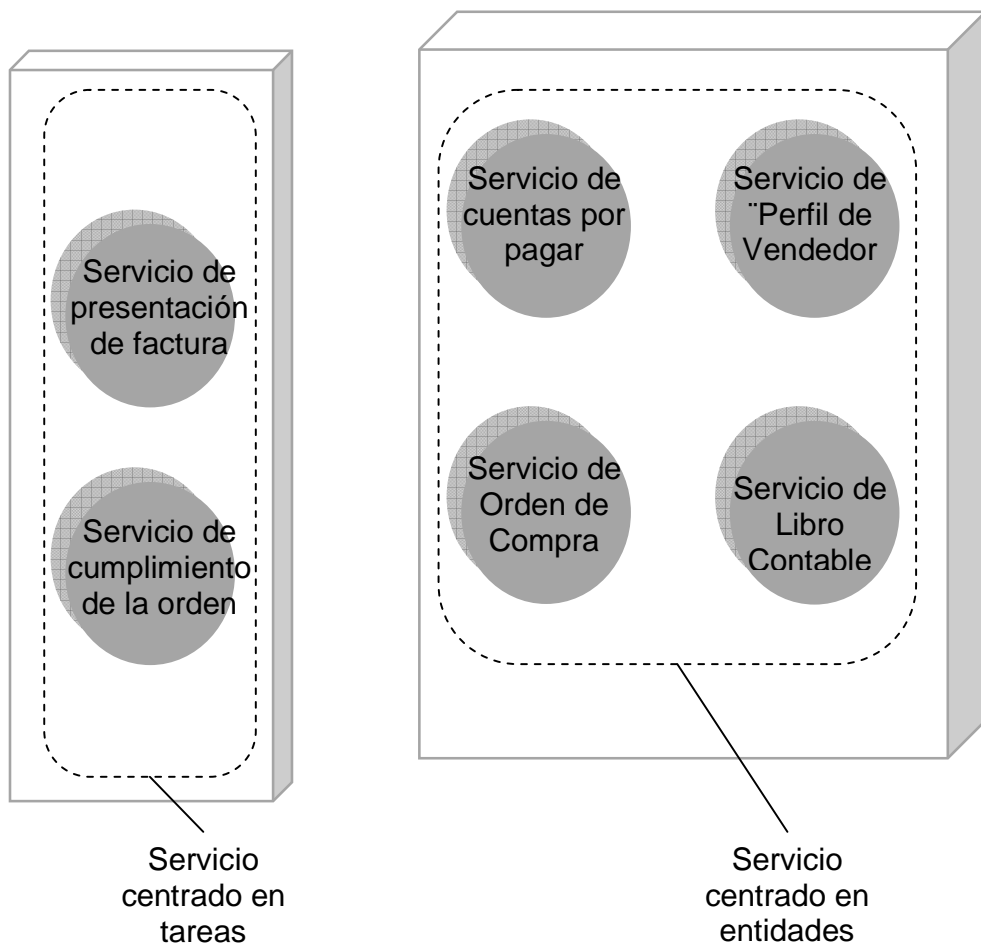
A continuación se presenta el conjunto de servicios de negocio centrados en entidades que tiene ABC:

- Servicio de Cuentas Por Pagar.
- Servicio de Orden de Compra.
- Servicio de Libro de Contabilidad.
- Servicio de Perfil de Vendedor.

Cuando estos servicios son utilizados dentro de la solución existente, realmente no necesitan ser integrados por un servicio independiente centrado en tareas o en procesos.

La razón por la que esta capa controlador padre no es requerida es porque el servicio de cuentas por pagar y el servicio de orden de compra no hacen parte de un proceso de negocio de solución específica.

Ellos simplemente cumplen con su función de procesar documentos para el sistema contable, recibiendo facturas validas y emitiendo órdenes de compra validas, por lo tanto no se requiere lógica de flujo de trabajo adicional.



### **2.4.2.3 Servicios de negocio y orquestación.**

El servicio de proceso y la implementación de orquestación<sup>22</sup>, pueden ser clasificadas como una forma de servicio de negocio, es como servicios centrados en negocios, y se encuentran en la parte más alta de la jerarquía de las capas de servicio y es responsable de la composición de los servicios de negocio de acuerdo a las reglas especificadas en el flujo de trabajo (WorkFlow) lógico de la orquestación.

Una orquestación puede estar compuesta por una combinación de servicios de negocios centrados en tareas y centrados en entidades. El modelo de negocio principal es representado por los servicios centrados en entidades, mientras que los negocios de tareas de lógica relacionada pueden ser implementados por servicios centrados en tareas, que estos son designados para suplementar el servicio de procesos.

Esencialmente, el uso de orquestación establece la siguiente estructura en las capas de servicios:

1. Las reglas de los negocios de lógica del flujo de trabajo (WorkFlow) y de procesos específicos se encuentran embebidos en una definición de procesos. La orquestación compone los servicios de negocios (y posibles servicios de aplicación) de acuerdo a la definición.

---

<sup>22</sup> Wikipedia®. "Orquestacion" [En línea]. [citado: 21-05-2009]. Enciclopedia en línea Wikipedia®. Disponible en Internet: <http://es.wikipedia.org/wiki/Animaci3n>. Ultima actualizaci3n. 20-may-2009.

2. Los servicios de negocios componen los servicios de aplicación para ejecutar una lógica de negocio.
3. Los servicios de aplicación interactúan con sistemas fundamentales para procesos de funciones requeridas.

la orquestación abstrae la lógica del flujo de trabajo (WorkFlow), la posiciona fuera de los límites de los servicios. Esto aumenta la agilidad, permitiendo que los cambios a las reglas de negocio se hagan sin afectar los servicios de negocios y los servicios de aplicación. Otro aspecto de la orquestación es como la lógica de los procesos de negocio están sujetos a muchos factores que resultan de los cambios, esto incluye la intervención humana, que realizan cambios en las políticas corporativas y las reglas de negocio y

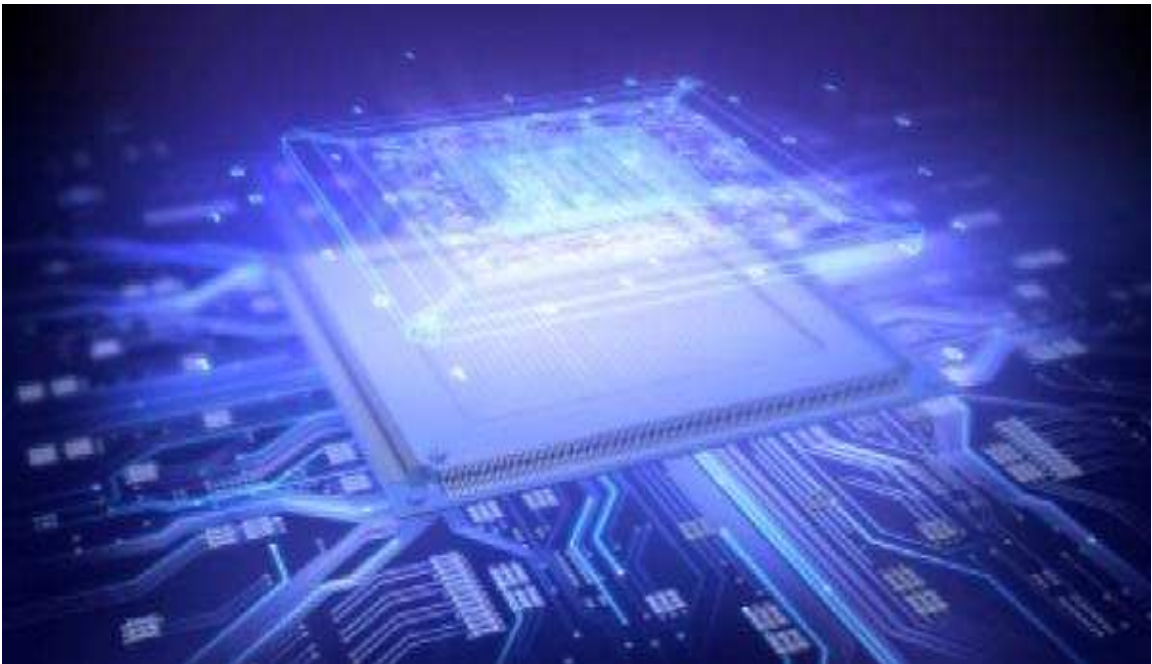
#### CASO DE ESTUDIO

ABC ha decidido que como parte del plan de ampliar la capacidad existente de SOA, introducirán una capa de servicio de orquestación. Esto inicia introduciendo un servicio de proceso que abstraiga las reglas de negocio y la lógica de flujo de trabajo (WorkFlow) de los servicios de negocios que se encuentran en uso.



## CAPITULO 3

# ***MODELADO ORIENTADO A SERVICIOS***



Ilustracion tomada de Referencia 3.

### **3.1. MODELADO DE SERVICIOS**

Un proceso de modelado de servicios<sup>23</sup> es en esencia una organización de la información que se recolecta en los pasos iniciales del análisis orientado a servicios. Las fuentes de la información requerida pueden ser diversas, desde diversos documentos del modelo de negocio existentes hasta entrevistas con personal clave que podría tener el conocimiento requerido de un área del negocio relevante. El proceso de modelado de servicios no arroja como resultado final servicios y operaciones de servicio definitivas, en su lugar produce *candidatos a servicios* y *candidatos a operaciones de servicio*. La diferenciación de ambos es muy importante ya que una vez que los candidatos sean sometidos al proceso de diseño, estarán sujetos a las realidades de la arquitectura técnica en la cual se espera que estos residan. Una vez que las restricciones, requerimientos, y limitaciones específicas al entorno de implementación sean construidas, el diseño final de un servicio podría ser significativamente diferente del candidato original correspondiente al mismo. En conclusión, el proceso de modelado de servicios tiene como finalidad definir los candidatos a servicios y a operaciones de servicio que posteriormente serán utilizados en la fase de diseño de servicios.

#### **CASO DE ESTUDIO**

El entorno de ABC es repasado con el fin de reemplazar su conjunto actual de servicios con capas de servicio de negocio y aplicación bien definidas. A medida que se avance en el proceso de modelado de este capítulo, estos nuevos servicios tomarán forma mientras ensamblamos una serie de servicios candidatos que establecerán capas de servicio preliminares.

---

<sup>23</sup> ERL THOMAS, SOA Principles of Service Design (Prentice Hall Service-Oriented Computing Series). P 52

### 3.2. PASOS PARA MODELAR SERVICIOS

Generalmente se siguen doce pasos a la hora de modelar servicios. Específicamente los pasos que se muestran a continuación (Ilustración 18.) sirven para el modelado de una Arquitectura orientada a servicios que contenga las capas de servicio de aplicación, negocio y orquestación:

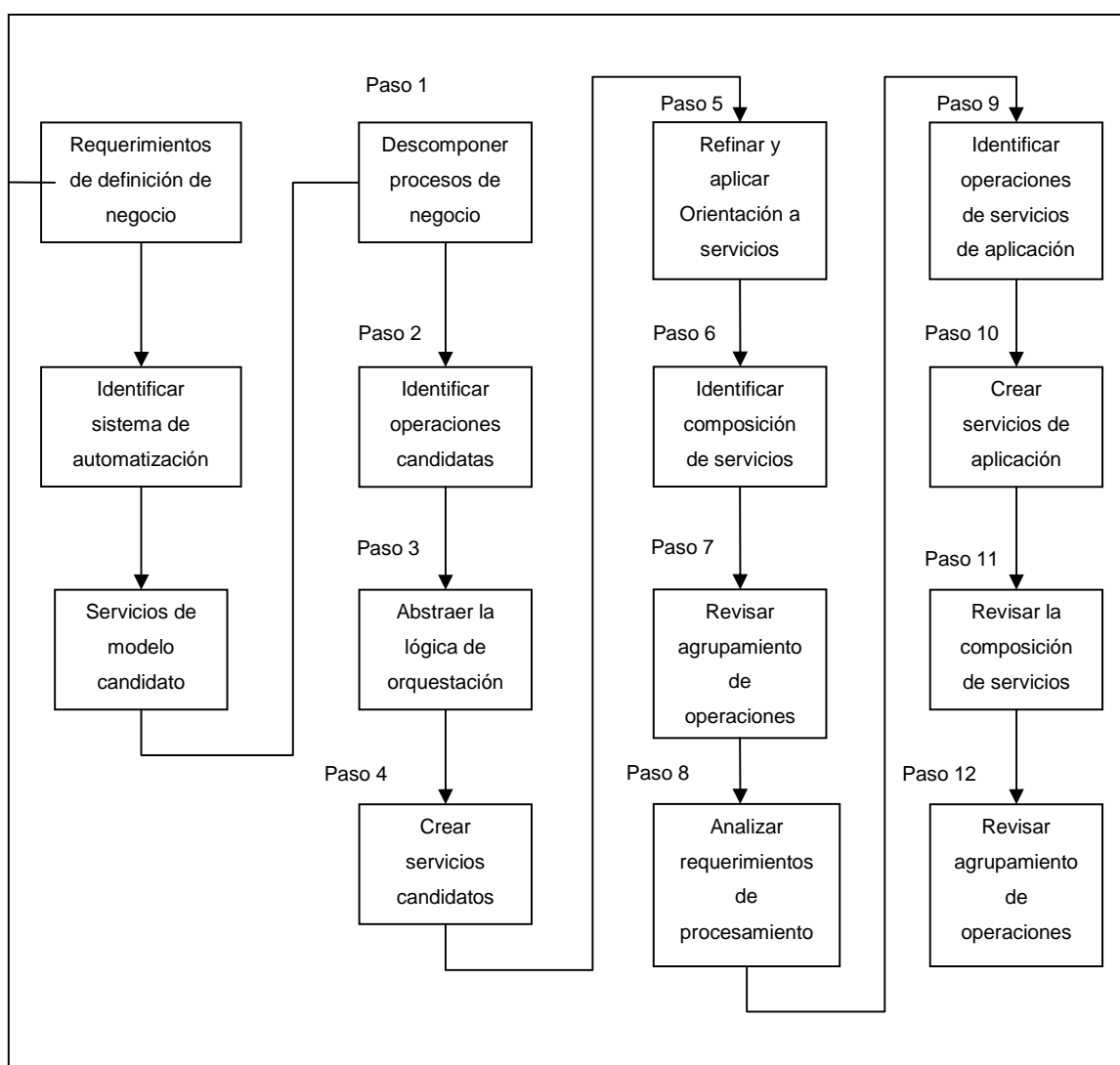


Ilustración 18 Pasos para realizar un modelado orientado a servicios

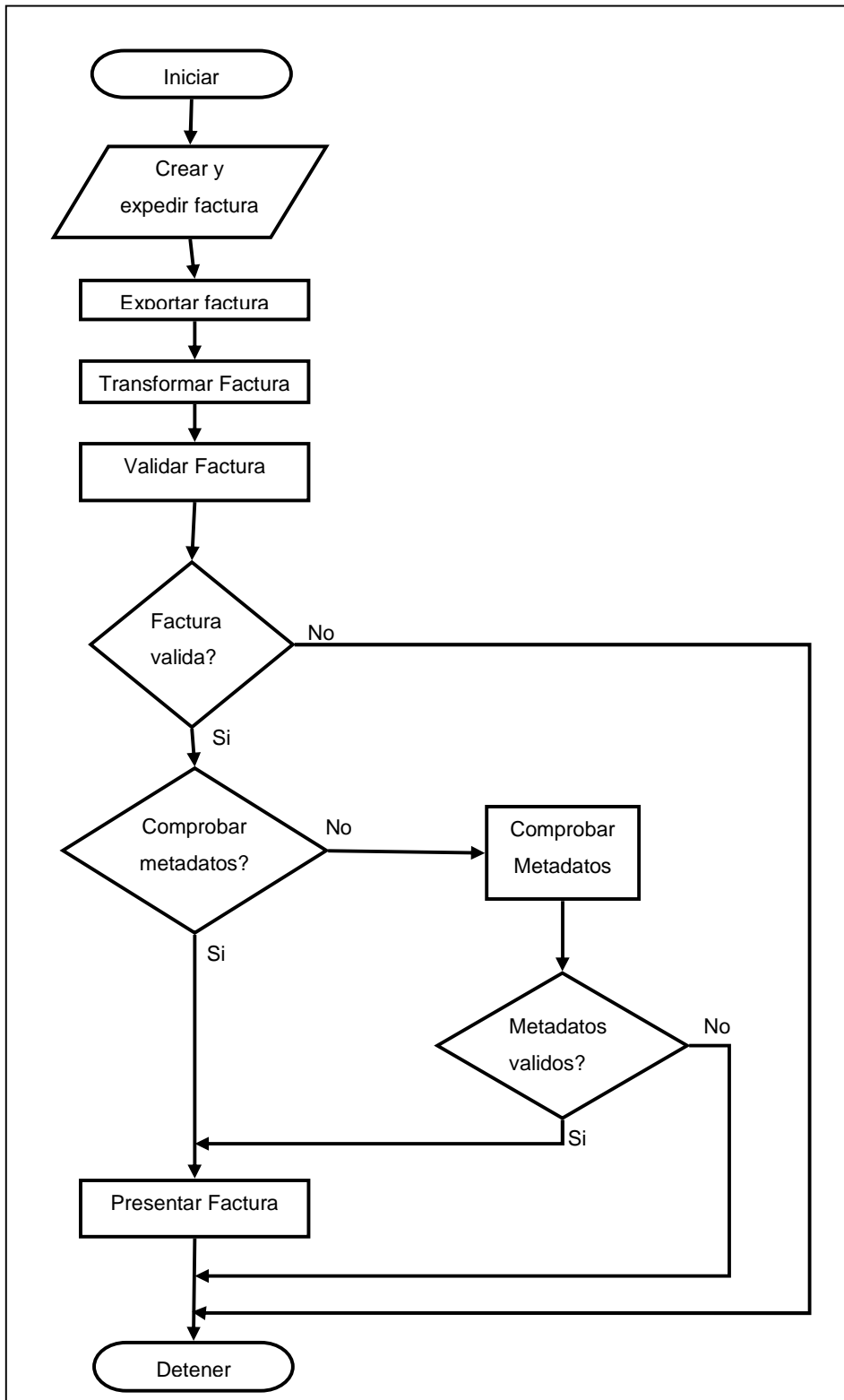
- *Descomponer el proceso de negocio:* se toma el proceso de negocio documentado y se divide en una serie de pasos de proceso granulares. Es importante que la lógica de un flujo de trabajo de proceso sea descompuesta en la mayor representación granular de los pasos de proceso, lo cual puede diferir del nivel de granularidad que tenían los pasos de proceso cuando este fue originalmente documentado. Véase el caso de estudio a continuación donde se realiza la descomposición de dos procesos de negocio.

#### CASO DE ESTUDIO

El alcance del análisis orientado a servicios de ABC incluye los procesos de completamiento de orden y de presentación de factura descritos en el capítulo anterior.

Se descompone el Proceso de Presentación de Factura (Ilustración 19.) en una serie de pasos granulares:

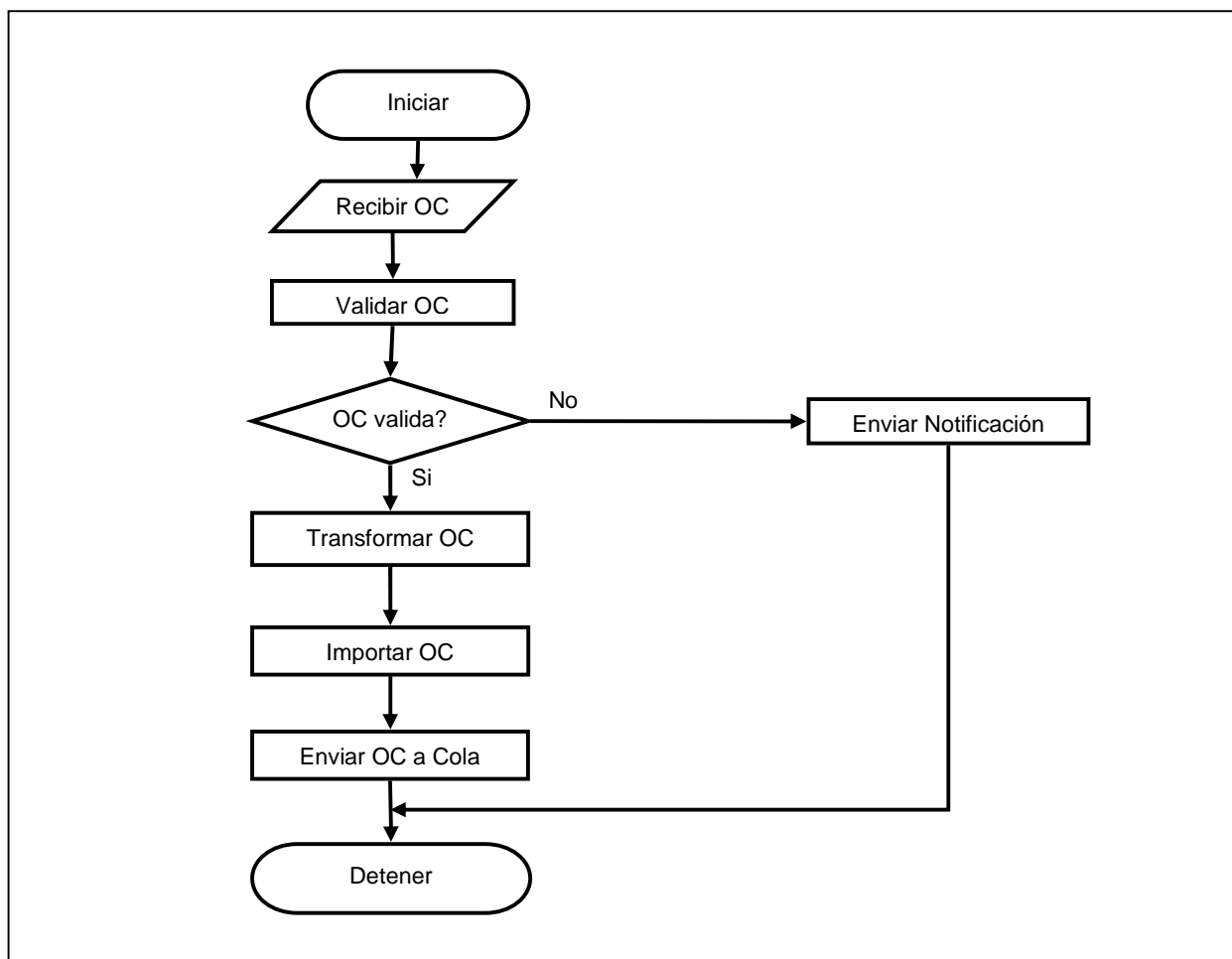
- Crear factura electrónica.
- Expedir factura electrónica.
- Exportar factura electrónica a directorio de red.
- Revisar directorio de red.
- Recuperar factura electrónica.
- Transformar factura electrónica a documento XML.
- Comprobar la validez del documento de factura. Si es inválida, terminar proceso.
- Comprobar si es momento de revisar los metadatos.
- Si se requiere, realizar comprobación de metadatos. Si la comprobación de metadatos falla, terminar proceso.



**Ilustración 19. Proceso de Presentación de Factura**

Pasos del Proceso de Completamiento de Orden (Ilustración 20.):

- Recibir documento de orden de compra.
- Validar documento de orden de compra.
- Si el documento de orden de compra es invalido, enviar notificación de rechazo y terminar proceso.
- Transformar documento XML de la orden de compra al formato nativo de orden de compra electrónica.
- Importar orden de compra electrónica dentro del sistema de contabilidad.
- Enviar orden de compra a la cola de trabajo del empleado de contabilidad.



20. Proceso de Completamiento de Orden

- *Identificar candidatos a operación de servicio de negocio:* algunos pasos dentro de un proceso de negocio pueden ser fácilmente identificados como no pertenecientes a la lógica potencial que debería ser encapsulada por un candidato a servicio. Como ejemplos pueden citarse a los pasos de proceso manuales que no pueden o no deberían ser automatizados y pasos de proceso que son realizados por lógica implementada en sistemas anteriores (sistemas legado) para la cual la encapsulación en un servicio candidato no es una opción.

#### CASO DE ESTUDIO

Después de revisar cada uno de los pasos de proceso previamente identificados, es necesario remover aquellos que no puedan o no queramos que sean parte de nuestra solución orientada a servicios.

A continuación presentaremos nuevamente los pasos de los procesos Presentación de Factura (Tabla 1.) y Completamiento de la Orden (Tabla 2.) detallados en el paso anterior, solo que esta vez identificaremos los pasos que no formaran parte de la solución y mediante la ayuda de las tablas se mostrara una descripción que contendrá la razón por la que serán excluidos (los pasos) o que se tiene planeado hacer con ellos.

**Tabla 1. Pasos removidos del Proceso de Presentación de Factura**

No	Paso	Removido	Descripción
1	Crear factura electronica	Si	Un paso manual realizado por el empleado de contabilidad.
2	Expedir factura electrónica	Si	Un paso manual realizado por el empleado de contabilidad.

3	Exportar factura electrónica a directorio de red	No	Actualmente es un extensión de un sistema heredado desarrollada a la medida. Podría ser parte de un candidato a servicio genérico.
4	Revisar directorio de red	No	Actualmente es realizado por un componente desarrollado a la medida. Podría ser parte de un candidato a servicio.
5	Recuperar factura electronica	No	Igual que el anterior.
6	Transformar factura electrónica a documento XML	No	Igual que el anterior.
7	Comprobar la validez del documento de factura. Si es invalida, terminar proceso.	No	Actualmente esta siendo ejecutado como parte de la rutina de comprobación del Servicio de Presentacion de Factura. No se prevee cambiar este comportamiento.
8	Comprobar si es momento de revisar los metadatos.	No	Actualmente esta siendo ejecutado como parte de la rutina de comprobación del Servicio de Presentacion de Factura. Parece que podría ser un candidato a operación potencialmente reusable. Podría ser movido a un candidato a servicio separado.
9	Si se requiere, realizar comprobación de metadatos. Si la comprobación de metadatos falla, terminar proceso.	No	Igual que el anterior.



**Tabla 2. Pasos removidos del Proceso de Completamiento de Orden**

<b>No</b>	<b>Paso</b>	<b>Removido</b>	<b>Descripción</b>
1	Recibir documento de orden de compra.	No	Actualmente es realizado por el Servicio de Completamiento de Orden. No se prevé la necesidad de cambiar este comportamiento.
2	Validar documento de orden de compra.	No	Igual que el anterior.
3	Si el documento de orden de compra es invalido, enviar notificación de rechazo y terminar proceso.	No	Igual que el anterior.
4	Transformar documento XML de la orden de compra al formato nativo de orden de compra electrónica.	No	Actualmente es realizado por un componente desarrollado a la medida. Podría ser parte de un candidato a servicio.
5	Importar orden de compra electrónica dentro del sistema de contabilidad.	No	Actualmente es un extensión de un sistema heredado desarrollada a la medida. Podría ser parte de un candidato a servicio genérico.
6	Enviar orden de compra a la cola de trabajo del empleado de contabilidad.	No	Igual que el anterior.

- *Abstraer lógica de orquestación:* si se ha decidido construir una capa de orquestación como parte de la SOA, entonces se deberían identificar las partes de la lógica de procesamiento que esta capa abstraería potencialmente. Los tipos de lógica que potencialmente se adaptan a esta capa son: reglas de negocio, lógica condicional, lógica de excepción y lógica de secuencia. Como en el siguiente caso de estudio que se identifican las partes de la lógica de orquestación.

#### CASO DE ESTUDIO

Con base en las descripciones de los dos procesos en estudio, la lógica de flujo de trabajo (Workflow) representada por un proceso separado candidato a servicio incluiría (pero no se limitaría):

- Si el documento de factura es válido, proceder con el paso de comprobación de metadatos.
- Si el documento de factura es invalido, terminar proceso.
- Si el intervalo para realizar una comprobación de metadatos se ha completado, proceder a la ejecución del paso de comprobación de metadatos.
- Si el intervalo no se ha completado, omitir el paso de comprobación de metadatos.
- Si el documento de Orden de Compra es valido, proceder con el paso de transformación de documento de Orden de Compra.
- Si el documento de Orden de Compra es invalido, terminar proceso.

Nótese que la lógica de flujo de la orquestación también incluiría la secuencia en la cual son ejecutados los pasos de procesamiento individuales.

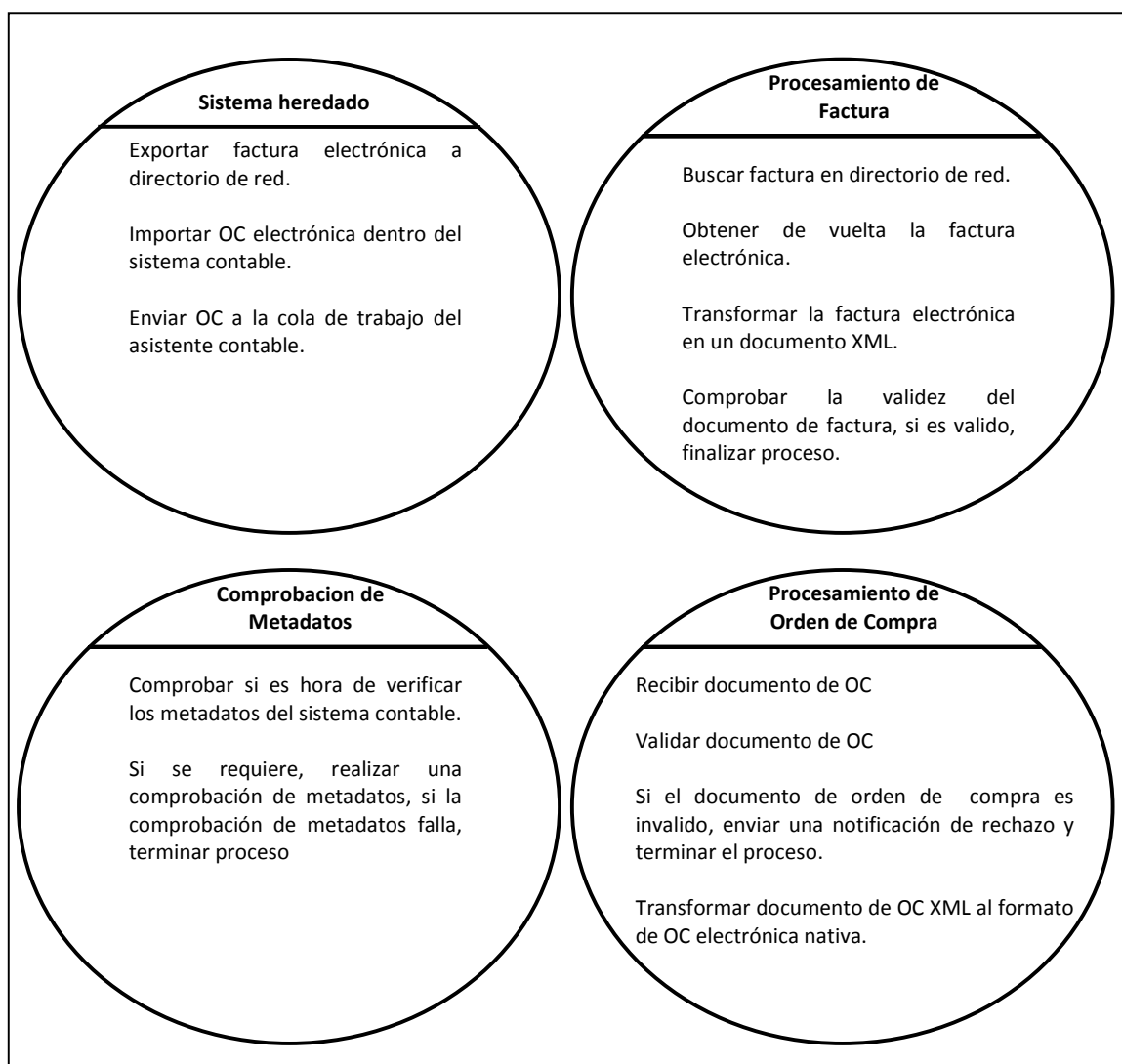
- *Crear candidatos a servicio de negocio:* revisar los pasos de procesamiento que queden y determinar uno o más contextos lógicos con los cuales estos pasos puedan ser agrupados. Cada contexto representa un candidato a servicio. Los contextos con los que se finalice dependerán de los tipos de servicio de negocio que hayan sido escogidos para construirse.

### CASO DE ESTUDIO

A continuación se presenta como podrían ser agrupados los pasos de los procesos descritos anteriormente que son candidato a servicio (Ilustración 21.):

- Servicio de Sistema Heredado
  - Exportar factura electrónica a directorio de red.
  - Importar Orden de Compra electrónica en el sistema contable.
  - Enviar Orden de Compra a la cola de trabajo del empleado de contabilidad.
- Servicio de Procesamiento de Factura
  - Revisar directorio de red por facturas.
  - Recuperar factura electrónica.
  - Transformar factura electrónica a documento XML.
  - Comprobar validez del documento de factura. Si es invalido, terminar proceso.
- Servicio de Procesamiento de Orden de Compra
  - Recibir documento de Orden de Compra.
  - Validar documento de orden de Compra.
  - Si el documento de Orden de Compra es invalido, enviar notificación de rechazo y terminar proceso.

- Transformar documento de Orden de Compra XML a formato nativo de Orden de Compra.
- Servicio de Comprobacion de Metadatos
  - Comprobar si es momento de verificar los metadatos del sistema contable.
  - Si se requiere, realizar comprobación de metadatos. Si la comprobación de metadatos falla, terminar proceso.



**Ilustración 21. Candidatos a servicio**

- *Refinar y aplicar principios de la orientación a servicios:* en este paso esencialmente se requiere verificar que cada candidato a operación de servicio sea potencialmente reusable y tan autónomo como sea posible, como se hace en el siguiente caso de uso.

#### CASO DE ESTUDIO

Revisando las operaciones candidatas dentro de los servicios candidatos, se determina que es necesario realizar una serie de ajustes como se muestran en la Ilustración 22.

Dentro del Servicio de Sistema Heredado, la acción “Enviar OC a la cola de trabajo del asistente contable” puede ser realizada solamente en la recepción de un documento. Por lo tanto, esta operación candidata depende directamente del paso “Importar OC electrónica dentro del sistema contable”, y se decide combinar estos dos pasos en uno.

La acción “Exportar factura electrónica a directorio de red” es realizada automáticamente por una macro agregada al sistema contable heredado. Por lo tanto no es requerido como parte del servicio candidato. El resultado de lo anterior es una única operación candidata que sería más reusable permitiendo manejar diferentes tipos de documentos.

El Servicio de Sistema Heredado revisado contiene los siguientes pasos:

- Exportar documento a directorio de red.
- Importar y redirigir documento a la cola de trabajo.

Al revisar el Servicio de Procesamiento de Factura, se determina que la acción “Buscar factura en directorio de red” puede ser hecha más genérica convirtiéndola en una operación candidata que simplemente busque diferentes tipos de documentos en un directorio dado. También se decide que esta acción

debería ser parte de un servicio candidato capaz de notificar a los subscriptores de la llegada de nuevos documentos.

A continuación, se decide combinar las operaciones candidatas “Obtener de vuelta la factura electrónica”, “Transformar la factura electrónica en un documento XML” y “Comprobar la validez del documento de factura” en una sola operación candidata llamada “Obtener de vuelta y transformar documento de factura”. No se menciona el aspecto de validación de esta acción porque al documento XML se le asigna automáticamente su esquema correspondiente. La validación del documento es por lo tanto, una parte intrínseca del proceso de transformación.

El resultado del análisis es un nuevo contexto (un nuevo servicio candidato), establecido para representar acciones de notificación genéricas, tal como se presenta a continuación:

Servicio de Notificación de Búsqueda:

- Buscar nuevos documentos en el directorio.
- Si llegan documentos para los cuales hay subscriptores, enviar notificaciones.

El Servicio de Procesamiento de Factura queda con un paso solamente:

- Obtener de vuelta y transformar la factura electrónica.

A continuación, se atacan las operaciones candidatas del Servicio de Orden de Compra. La acción “Recibir documento de OC” no es un servicio candidato apropiado como tal, ya que recibir un mensaje es una parte natural de las operaciones de un servicio ( y por lo tanto no se debería tener en cuenta como uno). Como conclusión se remueve esta acción de la lista.

Continuando con el proceso de revisión, una dependencia directa es detectada entre las acciones “Validar documento de OC” y “Si el documento de orden de compra es invalido, enviar una notificación de rechazo y terminar el proceso”. Como resultado, se decide combinar estas en una operación candidata única llamada “Validar documento de OC y enviar notificación de rechazo, si es requerido”.

Ahora el foco se concentra en descubrir factores comunes entre la acción “Transformar documento de OC XML al formato de OC electrónica nativa” del Servicio de Procesamiento de Orden de Compra y la acción “Obtener de vuelta y transformar la factura electrónica” del Servicio de Procesamiento de Factura. Ambas operaciones candidatas transforman documentos contables. Por lo tanto se decide crear un nuevo servicio que provea transformación genérica. El resultado es una nueva categoría de agrupamiento:

Servicio de Transformación de Documentos Contables:

- Transformar documentos XML a formato nativo.
- Transformar documentos nativos a XML.

Después de la revisión, el Servicio de Procesamiento de Orden de Compra queda con un solo paso:

- Validar documento de OC y enviar notificación de rechazo, si es requerido.

Finalmente, el último grupo de operaciones candidatas es revisado. Los candidatos por si mismos son todavía relativamente apropiados. Sin embargo, como se abstrajeron en un servicio candidato genérico, es necesario revisar la redacción para que refleje mejor esto.

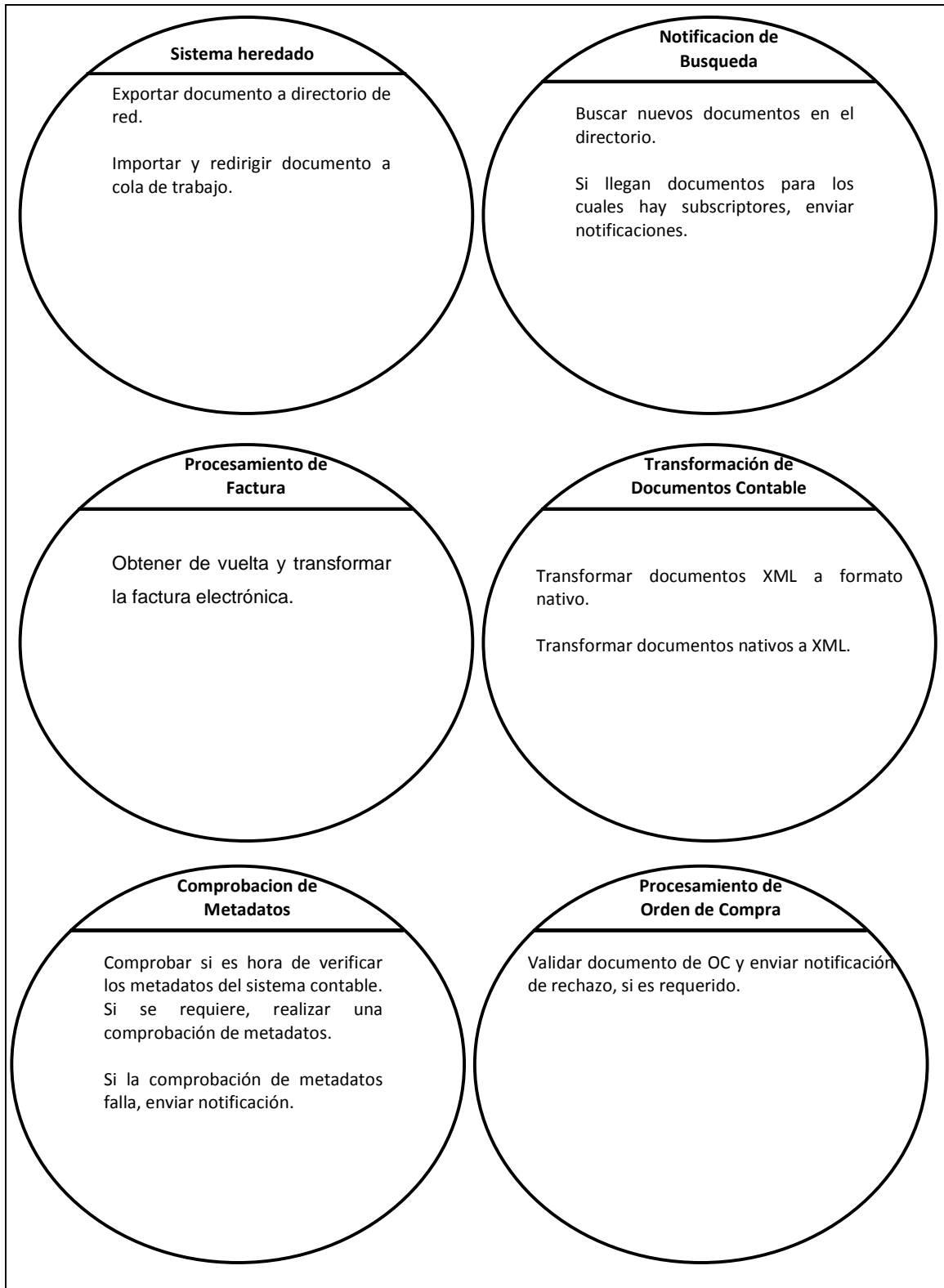
Específicamente, se agrega una característica de notificación al Servicio de Comprobación de Metadatos.

Después de la revisión, el Servicio de Comprobación de Metadatos contiene los siguientes pasos:

- Comprobar si es hora de verificar los metadatos del sistema contable. Si se requiere, realizar una comprobación de metadatos.
- Si la comprobación de metadatos falla, enviar notificación.

En este paso se realizaron algunas revisiones importantes a la lógica de negocio original. El resultado es la creación de servicios candidatos adicionales que abstraen la lógica siguiendo los principios claves de la orientación a servicios.





**Ilustración 22. Candidatos a Servicios (Revisados)**

- *Identificar los elementos que compondrán a un candidato a servicio:* Identificar un grupo de los escenarios más comunes que se pueden dar dentro de los límites de un proceso de negocio. Para cada escenario se siguen los pasos de procesamiento requeridos.

Este paso permite:

- Tener una buena idea de cuan apropiado es el agrupamiento de los pasos de proceso.
  - Demostrar la relación potencial entre la orquestación y las capas de servicio de negocio.
  - Identificar potenciales composiciones de servicios.
  - Identificar cualquier lógica de flujo de trabajo o pasos de procesamiento que hayan sido pasados por alto.
- 
- *Revisar el agrupamiento de las operaciones de servicios de negocios:* basado en los resultados de las composiciones realizadas en el paso anterior, se deben revisar los pasos de los procesos de negocio, así como también la organización de los candidatos a operaciones de servicio. No es poco usual que sea necesario consolidar o crear nuevos grupos en este punto.
- 
- *Analizar los requerimientos de procesamiento de aplicación:* este paso consiste en realizar un pequeño análisis sobre cada paso de procesamiento; este análisis es necesario para determinar:
    - La lógica de aplicación subyacente que necesita ser ejecutada para procesar la acción descrita por el candidato a operación.
    - Si la lógica de aplicación requerida ya existe o si necesita ser desarrollada desde cero.
    - Si la lógica de aplicación requerida abarca los límites de la aplicación. En otras palabras, se requiere de más de un sistema para completar esta acción?

- *Identificar candidatos a operación de servicio de aplicación:* dividir cada requerimiento de procesamiento de lógica de aplicación en una serie de pasos. Es necesario ser explícito acerca de cómo se denominan estos pasos de tal manera que referencien la función que desempeñan.
  
- *Crear candidatos a servicios de aplicación:* Agrupar estos pasos de procesamiento acorde a un contexto predefinido. Para los candidatos a aplicación de servicio, el contexto primario es una relación lógica entre candidatos a operación. Esta relación puede estar basadas en cualquier número de factores, entre los cuales se encuentran incluidos:
  - Asociación con un sistema heredado específico.
  - Asociación con uno o más componentes de la solución.
  - Agrupamiento lógico de acuerdo al tipo de función.
  
- *Revisar la composición de los candidatos a servicio:* Mirar nuevamente los escenarios identificados en el paso 6. Solo que esta vez también se deben incorporar los nuevos candidatos a servicio de aplicación. Esto dará como resultado un modelo de actividades coordinadas que darán vida a composiciones de servicio extendidas.
  
- *Revisar el agrupamiento de operaciones de servicio de aplicación:* en este punto se debe revisar cualquier omisión que se haya cometido en los pasos de procesamiento a nivel de aplicación, esto dará como resultado la adición de nuevos candidatos a operación de servicio y tal vez incluso a nuevos candidatos a servicio.

### **3.3. CLASIFICACION DE LA LOGICA DEL MODELO DE SERVICIO**

Hasta ahora la lógica de modelado se ha clasificado como candidatos<sup>24</sup> a servicio o candidatos a operaciones de servicio, sin embargo, referirse a una parte de la lógica de negocio como un candidato no da ningún indicio de cuál es la naturaleza de la lógica que está siendo representada.

La encapsulación orientada a servicios permite que una sencilla operación exprese potencialmente una gran cantidad de lógica.

Cuando se está modelando la lógica de negocio, es muy útil el ser capaz de entender el alcance de la lógica representada por un candidato a operación, servicio o proceso. Para esto, es necesario contar con un sistema de clasificación de unidades de lógica de negocio.

Existe un sistema de clasificación en donde a cada nivel de clasificación se le refiere como un *bloque de construcción*.

Los bloques de construcción (también conocidos como *unidades de modelado de servicio*) son simples etiquetas aplicadas a unidades de lógica de negocio que ayudan en la composición o descomposición de una empresa orientada a servicios. Estas etiquetas pueden ser usadas para identificar tipos específicos de lógica, las cuales se distinguen principalmente por el alcance de las mismas.

---

<sup>24</sup> ERL THOMAS SOA Design Patterns (The Prentice Hall Service-Oriented Computing Series) p. 44

### 3.3.1. El modelo SOE

En la siguiente figura ( Ilustración 23.) se puede apreciar el modelo aplicado a una empresa orientada a servicios, este modelo se conoce como el modelo SOE<sup>25</sup> (Service Oriented Enterprise, por sus siglas en inglés).

Dentro de este modelo se establecen los bloques de construcción, cada uno incrementando (de izquierda a derecha) su alcance.

Los bloques de construcción permiten categorizar distintas unidades de lógica para propósitos de modelado y diseño.

La primera capa de esta vista establece el modelo de negocio empresarial, una serie de bloques de construcción usados para representar solamente lógica de negocio modelada. La tercera capa provee los bloques de construcción del modelo de tecnología empresarial. La capa que se encuentra entre las dos anteriores representa el núcleo del modelo SOE<sup>26</sup>, el cual está definido en conjunto por las capas abstractas (negocio) y concreta (tecnología).

---

<sup>25</sup> Wikipedia®. "SOE" [En línea]. [citado: 21-05-2009]. Enciclopedia en línea Wikipedia®. Disponible en Internet: <http://es.wikipedia.org/wiki/Animaci3n>. Última actualización 20-may-2009.

<sup>26</sup> JAMES P. LAWLER, H. HOWELL-BARBER Service-Oriented Architecture: SOA Strategy, Methodology, and Technology p. 157

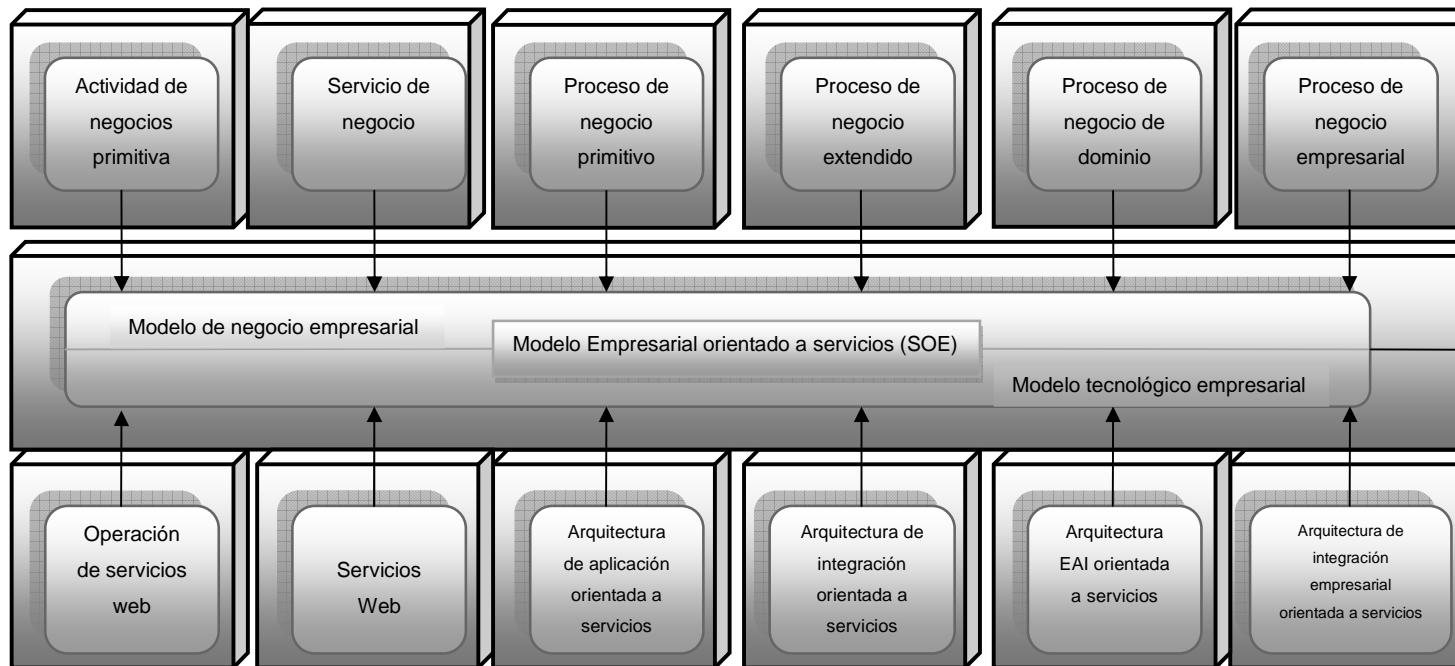


Ilustración 23 El modelo SOE

### **3.3.2. El modelo de negocio y tecnológico empresarial**

Los bloques de construcción en esta primera capa clasifican la lógica encapsulada por candidatos a servicio de negocio únicamente. Proveen una representación abstracta de la inteligencia de negocio de una empresa orientada a servicios, independientemente de la plataforma tecnológica subyacente sobre la cual esta implementada.

En el momento en que se quiera cambiar la plataforma tecnológica, no se perderá la perspectiva abstracta y orientada a servicios de la lógica de negocio empresarial. Estos bloques de construcción ayudan a etiquetar y categorizar la lógica que reside en las capas de orquestación y negocio.

### **3.3.3. Bloques de construcción para el modelado básico**

A continuación se explicaran a detalle los bloques de construcción que se utilizan en el modelado básico de una SOA.

#### *Actividades de negocio primitivas*

Una actividad de negocio primitiva representa la parte más pequeña de lógica de negocio (definible y ejecutable) dentro de un entorno orientado a servicios. Típicamente esto significa que al momento de dividir en partes un proceso de negocio, las actividades de negocio primitivas son las más pequeñas. Por lo tanto se asume que su lógica (del proceso de negocio) no puede, ni requerirá en un futuro ser descompuesta.

### *Servicios de negocio*

El servicio de negocio representa al ya mencionado anteriormente candidato a servicio de negocio. Dentro del contexto de este sistema de clasificación, cada servicio de negocio está compuesto de una o más actividades de negocio primitivas. Estas actividades pueden residir atómicamente dentro del servicio, o pueden estar interrelacionadas. En este último caso, las actividades de negocio primitivas podrían formar un algoritmo lógico que puede establecer lógica de flujo de trabajo independiente y reglas de negocio asociadas.

Cuando se implementa físicamente, la lógica que un candidato a servicio de negocio representa, típicamente existe como un servicio web.

### *Servicios de negocio primitivos*

Un servicio de negocio primitivo (o candidato a servicio de negocio primitivo) es un tipo de servicio de negocio que posee una funcionalidad limitada a una simple tarea o función de negocio. En otras palabras este bloque de construcción representa el tipo de servicio más granular dentro de las soluciones orientadas a servicios. Usualmente se expresa como un servicio Web granular.

### *Procesos de negocio primitivos*

Un proceso de negocio primitivo representa un cuerpo de lógica de flujo de trabajo compuesta de un grupo de actividades de proceso relacionadas. Un proceso de negocio primitivo es definido por una limitante funcional distintiva, la cual está típicamente relacionada a una tarea específica de negocio. Puede ser representada a través de un servicio de proceso o un servicio de negocio centrado en tareas.



**CAPITULO 4**  
***DISEÑO ORIENTADO A SERVICIOS***



**Ilustración tomada de Referencia 4.**

## **4.1. INTRODUCCION AL DISEÑO ORIENTADO A SERVICIOS**

El diseño orientado a servicios<sup>27</sup> es el proceso mediante el cual los diseños concretos de los servicios son construidos a partir de los candidatos a servicio lógicos, para luego ser ensamblados en composiciones abstractas que implementan un proceso de negocio.

Los objetivos perseguidos por el diseño orientado a servicios son:

- Determinar el grupo principal de extensiones arquitectónicas.
- Definir los límites de la arquitectura.
- Identificar los estándares de diseño requeridos.
- Definir diseños de interfaces de servicio abstractas.
- Identificar composiciones de servicio potenciales.
- Estimar la aplicación de los principios de la orientación a servicios.
- Explorar el soporte de las características de la SOA contemporánea.

El proceso de diseño orientado a servicios comprende cinco pasos:

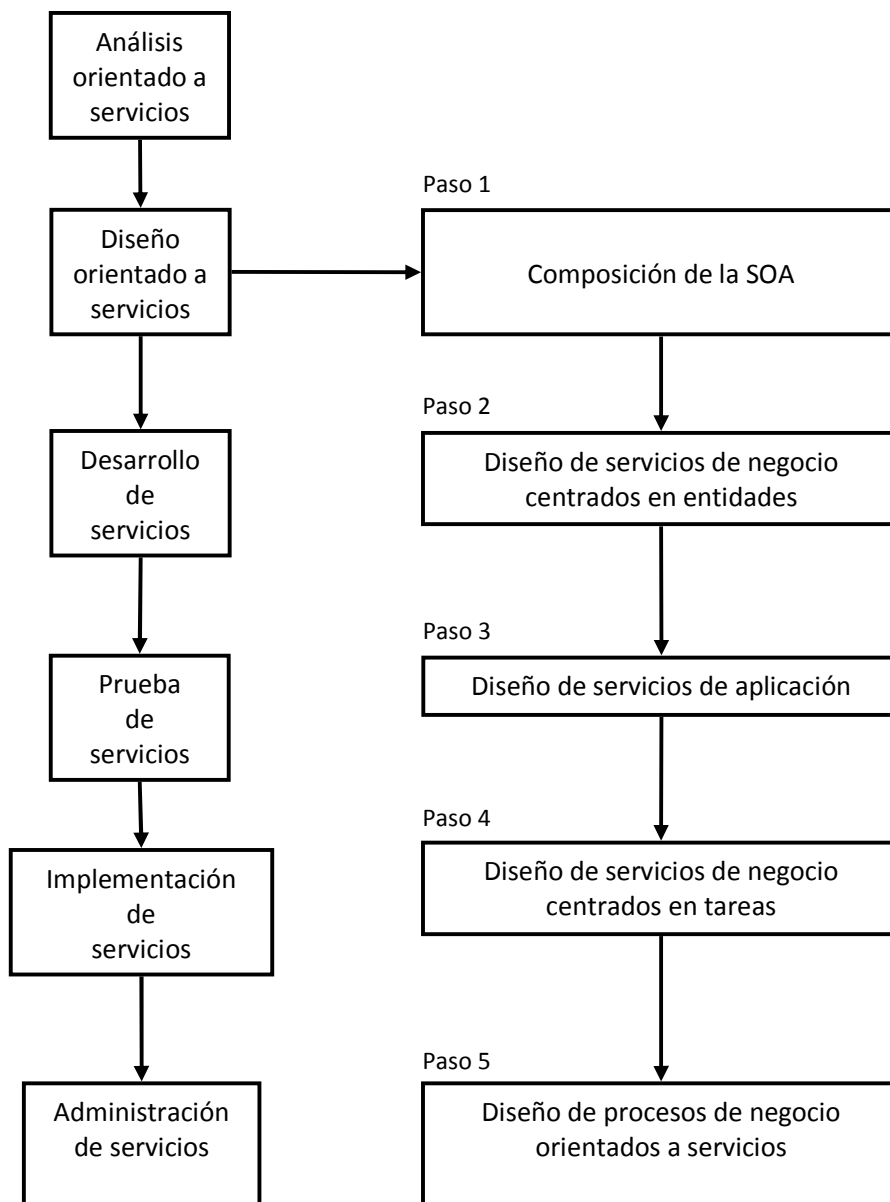
- 1) Composición de una SOA.
- 2) Diseño de servicios de negocio centrados en entidades.
- 3) Diseño de servicios de aplicación.
- 4) Diseño de servicios de negocio centrados en tareas.

---

<sup>27</sup> ERL THOMAS SOA Principles of Service Design (Prentice Hall Service-Oriented Computing Series) p. 53

5) Diseño de procesos de negocio orientados a servicios.

En la siguiente figura (Ilustración 24.) se puede apreciar el orden en el que se ejecutan los pasos anteriormente descritos así como también la posición del proceso de diseño dentro del ciclo de desarrollo de una SOA:



**Ilustración 24 Pasos para realizar un diseño orientado a servicios**

## **4.2. COMPOSICION DE UNA SOA**

En este paso se espera determinar:

- Los tipos de servicio que deberían ser construidos, y como deberían ser organizados dentro de las capas de servicio.
- El posicionamiento de los estándares de primera generación para soportar de mejor manera la Arquitectura.
- Las funcionalidades ofrecidas por extensiones, las cuales son requeridas por la Arquitectura.

Se deben llevar a cabo una serie de sub-pasos para concluir la composición<sup>28</sup>:

### *Escoger las capas de servicio*

Componer una Arquitectura orientada a servicios requiere que se decida sobre que configuración de diseño se trabajaran las capas de servicio que estandarizaran las representaciones lógicas dentro de la arquitectura. Este paso es completado estudiando los candidatos a capas de servicio producidos durante la fase de análisis orientado a servicios y explorando las capas de servicios básicas (aplicación, negocio y orquestación).

### *Posicionar estándares principales*

Es necesario tener en cuenta cuales estándares principales deberían comprender la arquitectura mencionada anteriormente y como deberían ser implementados

---

<sup>28</sup> MICHAEL ROSEN, BORIS LUBLINSKY, KEVIN T. SMITH, MARC J. BALCER Applied SOA: Service-Oriented Architecture and Design Strategies p. 78

para soportar de la mejor forma las características y requerimientos de la solución orientada a servicios.

#### *Escoger extensiones SOA*

Este paso requiere que se determine cuales características contemporáneas se quieren soportar en la solución. Esto ayudara a decidir cuál de las especificaciones WS-\* deberían convertirse en parte del entorno orientado a servicios.

## **4.2. DISEÑO DE SERVICIOS DE NEGOCIO CENTRADO EN ENTIDADES**

Los servicios de negocio centrados en entidades representan la capa de servicio<sup>29</sup> que es menos influenciada por otras (capas de servicio). Su propósito es representar de la manera más precisa los datos correspondientes a las entidades definidas dentro de los modelos de negocio de una organización. Estos servicios son construidos para ser re-usados por cualquier aplicación que necesite acceder o administrar información asociada con una entidad en particular.

A causa de que estos servicios existen preferiblemente de forma atómica en relación a otras capas de servicio, es recomendable diseñarlos antes que otros tipos de servicios. Esto establece una capa de servicio abstracta alrededor de la cual los procesos y la lógica de aplicación subyacente pueden ser posicionados.

A continuación se explicaran los pasos que generalmente se siguen a la hora de diseñar un servicio de negocio centrado en entidades:

---

<sup>29</sup> MICHAEL ROSEN, BORIS LUBLINSKY, KEVIN T. SMITH, MARC J. BALCER Applied SOA: Service-Oriented Architecture and Design Strategies p. 256

### *Paso 1: Revisar servicios existentes*

Idealmente, cuando se estén creando servicios centrados en entidad, el esfuerzo de modelado que tiene como resultado los servicios candidatos, tendrá que tener en cuenta cualquier servicio existente. Sin embargo, a causa de que los candidatos a servicio tienden a estar conformados por candidatos a operaciones relevantes a los requerimientos de negocio que formaron las bases del análisis orientado a servicios, siempre vale la pena asegurarse que alguna o toda la funcionalidad de procesamiento representada por los candidatos a operaciones no existan ya en otros servicios.

Por lo tanto, el primer paso para diseñar un nuevo servicio es confirmar si este es realmente necesario. Podrían existir otros servicios que provean alguna o toda la funcionalidad identificada en los candidatos a operación, o que ya hayan establecido un contexto adaptable en el cual estos nuevos candidatos a operación puedan ser implementados (como nuevas operaciones al servicio existente).

### *Paso 2: Definir los tipos del esquema de mensajes*

Es muy útil comenzar un diseño de interface de servicio con una definición formal de los mensajes que son necesarios que el servicio procese. Para completar esta tarea, es necesario formalizar las estructuras de mensaje que están definidas dentro del área de tipos WSDL.

Los mensajes SOAP contienen datos útiles dentro de de la sección *body*. Estos datos necesitan ser organizados y tipados. Para esto se cuenta con los esquemas XSD. Un esquema independiente puede ser incrustado en el constructor *types*, donde se pueden definir cada uno de los elementos usados para representar los datos dentro del cuerpo SOAP.

Es especialmente benéfico que el esquema XSD usado, represente de la manera más precisa la información asociada con la entidad del servicio.

*Paso 3: Derivar una interface de servicio abstracta*

En este paso se analiza el candidato a operación de servicio y se siguen los siguientes pasos para definir una interface de servicio inicial:

- 1) Confirmar que cada candidato a operación es adaptable a un nivel general y reusable, asegurando que la granularidad de la lógica encapsulada es apropiada. Se estudian las estructuras de datos definidas en el paso dos y se establece un conjunto de nombres de operación.
- 2) Crear el área *portType* dentro del documento WSDL y llenarlo con los constructores *operation* que correspondan a los candidatos a operación.
- 3) Formalizar la lista de valores de entrada y salida requeridos para acomodar el procesamiento de cada lógica de la operación.

*Paso 4: Aplicar los principios de la orientación a servicios*

Reusabilidad y autonomía, los dos principios que ya se explicaron anteriormente, son de alguna manera una parte natural del modelo de diseño centrado en entidades en el que se intenta que las operaciones expuestas por servicios de negocio centrados en entidades sean implícitamente reusables y genéricas.

A causa de que los servicios centrados en entidades necesitan por lo general ser compuestos por una capa de servicio padre y además necesitan de la capa de servicio de aplicación para que lleven su lógica de negocio, su autonomía inmediata está bien definida generalmente.

Es por las razones que se acaban de mencionar, que él no manejar estado es también relativamente manejable. Generalmente, los servicios centrados en entidades no procesan gran cantidad de lógica de flujo de trabajo y para esos casos en los cuales múltiples aplicaciones o servicios de negocio necesitan ser invocados para llevar a cabo una operación, es preferible que la administración del estado del servicio sea aplazada tanto como sea posible.

La exhibición es una parte importante del diseño de servicios centrados en entidades, así como también de su utilización después de ser implementados. Un mecanismo de exhibición haría más fácil determinar que un diseño de servicio no implemente lógica existente.

#### *Paso 5: Estandarizar y refinar la interface de servicio*

Dependiendo de los requerimientos, este puede ser un paso multifacético que involucra una serie de tareas de diseño. La siguiente es una lista de las acciones que se recomiendan tomar para alcanzar un estandarizado y estilizado diseño de servicio:

- Revisar estándares y guías de diseño existentes y aplicar cualquiera que sea apropiada.
- Revisar el soporte dado a algunas de las características contemporáneas de la arquitectura.

#### *Paso 6: Extender el diseño de servicio*

En este paso se realiza un análisis especulativo de que otro tipo de características debería ofrecer este servicio (dentro de su contexto funcional predefinido).

Existen dos maneras comunes de implementar nueva funcionalidad:

- Agregar nuevas operaciones
- Agregar nuevos parámetros a operaciones existentes



Mientras que la última opción podría estilizar las interfaces de servicio, también puede ser contra-intuitivo en el sentido que demasiados parámetros asociados con una operación podrían requerir que los solicitantes del servicio necesiten saber demasiado acerca del servicio para utilizarlo efectivamente.

Agregar operaciones es la manera más eficiente de proveer funcionalidades que están evidentemente relacionadas con la entidad. El conjunto clásico de operaciones para un servicio centrado en entidades es:

- Obtener Algo
- Actualizar Algo
- Eliminar Algo
- Adicionar Algo

Establecer estas operaciones estándar construye un consistente nivel de interoperabilidad dentro de la capa de servicio de negocio, facilitando la reusabilidad y composición ad-hoc.

Si nuevas tareas son definidas en su totalidad, entonces pueden ser incorporadas por nuevas operaciones que sigan los mismos estándares de diseño que las existentes. Si son identificados nuevos requerimientos funcionales que están relacionados a operaciones existentes, entonces un método común de extender estas operaciones es agregar valores de entrada y salida. Esto también permite que una operación transmita un rango de combinaciones de mensajes. Pero se debe tener cuidado de no complicar las operaciones en pro de la reusabilidad potencial. Por lo general es aconsejable someter a un proceso de análisis independiente cualquier nueva funcionalidad propuesta.

Por otro lado, aunque es recomendable producir servicios centrados en entidades que sean completamente auto-suficientes en la administración de los datos

asociados con el dominio de la entidad correspondiente, hay una práctica consideración clave que debería tenerse en cuenta: para cada nueva operación que se agregue, la manera por la cual esa operación completa su procesamiento también necesita ser diseñada e implementada. Esto conduce a él muy probable requerimiento de tener que agregar o extender servicios de aplicación. Siempre y cuando la sobrecarga para cada nueva operación sea calculada y aceptable, este paso es aconsejable.

Es importante tener en cuenta que la identificación de nuevas operaciones requiere que se repitan los pasos del uno al cinco para estandarizar y darle una forma apropiada a las extensiones añadidas.

#### *Paso 7: Identificar el procesamiento requerido*

En la etapa de modelado se han podido identificar algunos servicios de aplicación claves, pero puede que no se hayan podido definir todos.

Ahora que se tiene un diseño actual para este servicio de negocio, se pueden estudiar los requerimientos de procesamiento de cada una de sus operaciones de una manera más detallada. En este estudio, se debería ser capaz de determinar si nuevos servicios de aplicación son requeridos para realizar cada parte de la funcionalidad expuesta; si lo anterior es afirmativo, entonces se tendrá que determinar si los servicios de aplicación existen, o si necesitan ser agregados a la lista de servicios que serán entregados como parte de esta solución.

### **4.3. DISEÑO DE SERVICIOS DE APLICACIÓN**

Los servicios de aplicación<sup>30</sup> representan la sub-capa más profunda de la capa de servicios, es responsable de realizar cualquiera de las demandas de procesamiento dictadas por las capas de orquestación y de negocio.

Contrario a los servicios en las capas de centradas en negocio, el diseño de servicios de aplicación no requiere experiencia en el análisis de negocios. La capa de servicios de aplicación es una abstracción pura orientada a servicios de los entornos técnicos de una organización, la cual es mejor definida por aquellos que entienden mejor estos entornos.

El proceso de diseño de servicios de aplicación comprende siete pasos que se explican a continuación:

#### *Paso 1: Revisar servicios existentes*

En este paso se requiere determinar si existe redundancia de funcionalidad entre los servicios de aplicación, este requerimiento es un poco más difícil de cumplir que con otros servicios reusables.

Como estos servicios proveen funcionalidad genérica, vale la pena investigar si las características requeridas pueden ser compradas a arrendadas a terceros. Debido a que los servicios de aplicación deberían ser designados para máxima reusabilidad, puede tener sentido consumir servicios Web de terceros (que típicamente son construidos para ser reusables), siempre y cuando estos cumplan con los estándares de calidad requeridos.

---

<sup>30</sup> ERL THOMAS, SOA Principles of Service Design (Prentice Hall Service-Oriented Computing Series) p. 140

### *Paso 2: Confirmar el contexto*

Cuando se realiza un análisis orientado a servicios es natural estar enfocados en requerimientos de negocio inmediatos. Como resultado, los candidatos a servicio de aplicación producidos por esta fase no tomarán (frecuentemente) en cuenta los contextos establecidos por servicios de aplicación existentes.

Por lo tanto, es importante que el agrupamiento de candidatos a operación propuestos por los candidatos a servicio sea revaluado y comparado contra diseños de servicios de aplicación existentes. En este proceso se podría encontrar que una o más operaciones realmente pertenecen a otros servicios de aplicación.

### *Paso 3: Derivar una interface de servicio inicial*

Analizar los candidatos a operación de servicio y seguir los siguientes pasos para definir la primera parte de la interface de servicio:

- 1) Usando el candidato a servicio de aplicación como entrada principal, asegurarse que la granularidad de las particiones lógicas representada por los candidatos a operación son apropiadamente genéricas y reusables.
- 2) Documentar los valores de entrada y salida requeridos para el procesamiento de cada candidato a operación y definir estructuras de mensaje usando constructores de esquema XSD.
- 3) Completar la definición abstracta del servicio agregando el área *portType* y los constructores *message* necesarios que contengan los elementos *part* que referencien los tipos de esquema apropiados.

Es importante tener en cuenta que como unidades genéricas de lógica de procesamiento, los servicios de aplicación serán usados por diferentes tipos de servicios de negocio. Cada servicio de negocio estará procesando un tipo

diferente de documento de negocio (factura, orden de compra, recibo, etc.). Por lo tanto, los servicios de aplicación necesitan ser diseñados de tal manera que puedan procesar múltiples tipos de documentos. Dependiendo de la naturaleza de la información que está siendo procesada, existen diversas opciones de diseño.

*Paso 4: Aplicar los principios de la orientación a servicios*

Este paso resalta los cuatro principios de la orientación a servicios que fueron listados en el capítulo uno, como aquellos que no son ofrecidos intrínsecamente por la plataforma de servicios Web (reusabilidad, autonomía, servicios no tienen estado, y exhibición de servicios).

La reusabilidad ya fue vista en la fase de modelado, pero los candidatos a operación existentes deberían ser revisados para asegurar que están designados para ser genéricos y reusables.

La autonomía es un asunto primario en el diseño de servicios de aplicación. Se debe asegurar que la lógica de aplicación subyacente responsable de ejecutar las operaciones de servicio no imponga dependencias en el servicio, o sobre sí misma. Esta es la parte donde la información que se reunió en el paso dos del análisis orientado a servicios provee un punto de partida para investigar la naturaleza de la lógica de aplicación que cada servicio de operación necesita invocar.

Lograr que el servicio no tenga estado también podría ser más difícil con los servicios de aplicación. Como requieren hacer interface con una variedad de plataformas de aplicación diferentes, estos servicios están sujetos a entornos de implementación altamente impredecibles. Tarde o temprano, los servicios de aplicación encontrarán retos que imponen requerimientos de desempeño irracionales o inconsistentes (los sistemas heredados obsoletos son bien conocidos por esta razón). Por lo tanto, la mejor manera de promover un diseño

de servicio de aplicación que no tenga estado es realizar tanto análisis del front-end como sea posible. Sabiendo con anterioridad cuales serán las demandas de rendimiento, permitirá que se investiguen alternativas antes de confirmar un diseño en particular.

Como con servicios centrados en entidades, la exhibición puede ser una parte importante en la evolución de la capa de servicios de aplicación. El mecanismo de exhibición sirve más que todo para garantizar que este diseño no entre en conflicto con la lógica ya provista por otros servicios de aplicación.

*Paso 5: Estandarizar y refinar la interface de servicio*

Aunque el rol y propósito de los servicios de aplicación difieren de otros tipos de servicio, es importante que sean diseñados de la misma manera fundamental. Esto se logra asegurándose que la definición WSDL del servicio de aplicación este basada en los mismos estándares y convenciones usados por otros.

*Paso 6: Equipar al candidato a servicio con características especulativas*

Para los servicios de aplicación las extensiones especulativas giran en torno al tipo de procesamiento que cae dentro del contexto del servicio.

Es importante tener en cuenta que antes de agregar extensiones especulativas al servicio de aplicación se debería repetir el paso 1 para confirmar que ninguna parte de estas nuevas operaciones ya existan dentro de otros servicios.

*Paso 7: Identificar restricciones técnicas*

Ahora es necesario estudiar y documentar las demandas de procesamiento de cada operación de servicio de una manera más detallada. Primero, para cada operación, escribir una lista de las funciones de procesamiento requeridas para que la operación lleve a cabo su procesamiento. Entonces, para cada elemento de esta lista, encontrar exactamente como el procesamiento de la función necesitara ser ejecutado en el entorno tecnológico existente.

## RECOMENDACIONES

Es importante que los profesionales TI enfocados en el desarrollo de sistemas de gestión empresariales estén al día con respecto a los avances que se realizan en el campo de la arquitectura orientada a servicios, ya que les permitirá crear soluciones más dinámicas que ayuden a organizar, automatizar, regular y compartir de una manera efectiva los procesos de la compañía; reduciendo así costos de la misma en el mediano y largo plazo.

Recomendamos a los lectores ampliar este marco investigativo que encierra el tema de la arquitectura orientada a servicios, ya que este es muy amplio y requiere de una investigación constante y persistente para ir de la mano con los cambios que se van presentando, que este proyecto de inicio a una cadena de investigación y profundización en Tecnologías base de SOA: WEB SERVICES, SOAP, XML, XSLT, BPEL.

Se recomienda indagar en el tema de implementación de SOA en plataformas JAVA y MICROSOFT.

Es de gran importancia ampliar la información de herramientas que se puedan utilizar para el desarrollo de procesos de negocios como: Modelos BPM y Modelos Entidad.

Invitamos a explorar los temas de creación de servicios Web, integración y orquestación de procesos, consumo de servicios Web y gestionar en el ciclo de vida de los servicios.

Recomendamos a los lectores interesados en adoptar SOA a nivel empresarial seguir los siguientes aspectos:

Prepare a su equipo de trabajo (departamento de desarrollo) en: Servicios Web y Herramientas SOA seleccionadas: análisis, modelamiento, diseño, desarrollo, calidad, administración.

Defina una ruta de evolución en el tiempo de 2 o 3 años, respecto a como implementar SOA acorde a su empresa, acorde con sus necesidades actuales y futuras de negocio y la situación actual en tecnología. Defina objetivos y metas, hitos, prerequisites. Valore como está con respecto a; Proyectos, Tecnología, Gobierno, Arquitectura, Capital Humano. Recuerde revisar semestralmente su avance.

Defina una arquitectura de referencia empresarial, con estándares y herramientas de productividad. Esto permite la estandarización y el control sobre los principios, tecnologías y mejores prácticas utilizadas en todos sus proyectos internos.

Establezca un equipo de arquitectura conformado por personal de la gerencia y dirección de su empresa, expertos de negocios, personal técnico de desarrollo de proyectos, personal de seguridad informática, encargado de la implementación de la ruta de evolución de la Arquitectura en su empresa.

Inicie con un proyecto piloto. Encuentre un primer caso de negocios de alto impacto, alcance corto y bajo riesgo en su organización, con el cual pueda iniciar un proyecto piloto con esta Arquitectura. Prepare su equipo de trabajo, la metodología de trabajo, defina sus objetivos y metas. Identifique y mitigue los riesgos. Ejecute el proyecto y valore los resultados. Prepare su Infraestructura en producción. Recuerde que la puesta en producción de su proyecto piloto requiere al menos lo siguiente:

1. Adquisición, instalación y configuración de su infraestructura SOA de servidores de software y hardware.



2. Realización de pruebas de rendimiento y afinamiento de su infraestructura SOA, para asegurar la calidad de servicio requerida por la solución implementada.
3. Preparación de su equipo de administración y soporte técnico.
4. Administración del cambio en su organización.
5. Capacitación de los usuarios finales.

## CONCLUSIONES

La Arquitectura orientada a servicios garantiza la capacidad de intercambiar información entre sistemas, rapidez para producir y empezar a funcionar, capacidad de añadir, cambiar, remover o adaptar cualquier característica o funcionalidad y habilidad para aislar algún módulo de la aplicación, a fin de realizar cambios sobre el mismo o el resto de los módulos; además, este tipo de arquitectura se aplica a todo tipo de aplicaciones.

Como en la mayoría de los casos, cuando se trata de un cambio tecnológico, los obstáculos son de tipo organizacional y no técnico. Por ejemplo, para interconectar de manera eficiente aplicaciones o procesos que sean utilizados por varias áreas de la organización, es necesario compartir información y administrarla, lo cual requiere tiempo y recursos que no siempre estas áreas están dispuestas a invertir.

Dicha Arquitectura permite organizar los sistemas TI de una compañía al lograr una unión de datos entre ellos. Al implementar una arquitectura orientada al servicio, los procesos de negocio se visualizan más claros y eficientes. Es importante mencionar que al implementarla no debemos olvidar que todo el negocio debe estar implicado. Aporta ventajas importantes como:

1. Escalabilidad
2. Robustez
3. Homogeneidad
4. Facilidad en la adaptación de nuevos servicios
5. Facilidad en la reestructuración de sistemas
6. Aplicar lógica en el middleware pudiendo implementar procesos de negocio.
7. Recoger información y procesarla para obtener resultados más útiles.
8. Ahorro en tiempos de implantación

9. Ahorro en tiempos de mantenimiento y operación.

10. Ahorro en los costos de creación.

Si bien esta Arquitectura es aceptada ampliamente como la forma adecuada de construir aplicaciones modernas, la adopción de esta en grandes empresas a menudo es inhibida por la inercia cultural. A pesar de esto, muchas compañías están experimentando un éxito importante y ahorros sustanciales, incluso en las primeras etapas de la madurez de esta. Las empresas que instituyen SOA con éxito han trabajado para cambiar su mentalidad.

## BIBLIOGRAFÍA

- Erl Thomas. "Service-Oriented Architecture Concepts, Technology and Design." 1 ed. [United States]: Fundador de SOA Systems Inc.
- James A. Senn. Information Technology: Principles, Practices, and Opportunities (3rd Edition) (Hardcover - Dec 1, 2003)
- Michael Rosen, Boris Lublinsky, Kevin T. Smith, and Marc J. Balcer. Applied SOA: Service-Oriented Architecture and Design Strategies (Paperback - Jun 16, 2008).
- Erl Thomas. SOA Principles of Service Design (Prentice Hall Service-Oriented Computing Series)
- Eric A. Marks and Michael Bell. Service-Oriented Architecture (SOA): A Planning and Implementation Guide for Business and Technology (Hardcover - April 28, 2006).
- Dan Woods and Thomas Mattern. Enterprise SOA: Designing IT for Business Innovation (Paperback - April 28, 2006).
- James P. Lawler and H. Howell-Barber. Service-Oriented Architecture: SOA Strategy, Methodology, and Technology (Hardcover - Dec 3, 2007).
- Referencia 1: publicación en línea de bdsconsultores Disponible en Internet: [http:// www.bdsconsultores.com.ar](http://www.bdsconsultores.com.ar) Ultima actualización 29-may-2009.
- Referencia 2: publicación en línea Disponible en Internet: [http:// maohoy.blogspot.com](http://maohoy.blogspot.com) Ultima actualización 29-may-2009.
- Referencia 3: publicación en línea Disponible en Internet: [http:// images.quebarato.com.br](http://images.quebarato.com.br) Ultima actualización 29-may-2009.
- Referencia 4: publicación en línea Disponible en Internet:[http:// alhadradigital.files.wordpress.com](http://alhadradigital.files.wordpress.com) Ultima actualización 29-may-2009.

