Universidad de las Ciencias Informáticas

Trabajo para optar por el Título de Máster en Gestión de Proyectos Informáticos

Título: Proceso de diseño de servicios para proyectos SOA.

Autora: Ing. Yusleidy Guelmes León

Tutor: Dr. Pedro Piñero Pérez

Ciudad de la Habana, julio 2010

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo _____ soy la autora principal de la tesis de maestría _____

_____

desarrollada como parte de la Maestría en Gestión de Proyectos Informáticos y que autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio así como los derechos patrimoniales de la misma con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

_____ _____

Nombre del autor Firma

A mis padres por su apoyo incondicional,

su preocupación constante y por la educación

que me han dado.

A Yuri, quien me ha enseñado tantas cosas

valiosas, sobre todo el amor.

A mis compañeros de trabajo que han asumido parte de mis tareas,

A Yidier y Marbys por su apoyo en el trabajo y su preocupación por mí,

A Yoisy por sus sugerencias y por darme ánimos,

A los profes de la maestría, por sus enseñanzas y por el tiempo valioso que nos han dedicado para compartir sus conocimientos y experiencias.

A mis suegros por su apoyo inestimable en las tareas domésticas.

A mis padres, que nunca dejan de estar pendientes de mí,

A mi esposo Yuri, por todo su amor y por darme la confianza para seguir adelante.

Abstract

The Service Oriented Architecture (SOA) has emerged in this century as a new paradigm of software development. One of the key activities for the development of this kind of architecture is service design, which should take into account a set of best practices and principles to get the advantages of flexibility, low coupling and others that SOA proclaims. After analyzing several service-oriented methodologies, it was found that many of them are not accessible to our country and the rest are poor-defined or incomplete in matter of services design activities. This paper proposes a process for designing services focused on obtaining an adequate quality of design based on service-oriented principles. The proposed process includes a set of tasks, steps, tools, guidelines, roles definitions and artifacts for guiding designers and architects in the realization of SOA service design incorporating best practices recognized by the industrial and academic centers. The proposal was validated in practice through its application in a real project. The results of this application were used to analyze the service design accordance with services-oriented principles as well as the understandability of the design specifications.

Keywords: services design process, service-oriented architecture, SOA, service oriented principles.

Resumen

La Arquitectura Orientada a Servicios (SOA, del Inglés Services Oriented Architecture), ha emergido en este siglo como un nuevo paradigma de desarrollo de software. Una de las actividades claves para el desarrollo de este tipo de arquitecturas lo constituye el diseño de servicios en la cual deben tenerse en cuenta una serie de buenas prácticas y principios para lograr obtener las ventajas de flexibilidad, bajo acoplamiento, entre otras que promulga SOA. En un análisis realizado a varias de las metodologías existentes para desarrollar arquitecturas orientadas a servicios, se detectó que muchas no son accesibles para nuestro país y el resto resulta pobre o incompleta en cuanto a las actividades relacionadas con el diseño de servicios. En este trabajo se propone un proceso para diseñar servicios cuyo objetivo se centra en obtener una adecuada calidad de diseño de acuerdo a los principios de orientación a servicios. El proceso propuesto incluye un conjunto de tareas, pasos, herramientas, guías y definición de roles y artefactos que orientan a los diseñadores y arquitectos SOA en la realización del diseño de servicios incorporando buenas prácticas reconocidas en los ámbitos industrial y académico. La propuesta se validó de manera práctica mediante su aplicación en un proyecto real. Los resultados obtenidos de esta aplicación se utilizaron para analizar el comportamiento de los principios de orientación a servicios en el diseño realizado así como la facilidad de comprensión y completitud de las especificaciones de diseño obtenidas.

Palabras claves: proceso de diseño de servicios, arquitectura orientada a servicios, SOA, principios de orientación a servicios.

Índice

Listado de Apéndices

Introducción

Las Arquitectura Orientada a Servicios (SOA, del Inglés Services Oriented Architecture), ha emergido en este nuevo siglo como un paradigma de desarrollo de software. A finales de la década de los noventa del pasado siglo la mayoría de las empresas se caracterizaban por uno o varias de las siguientes dificultades:

- Coexistencia de un número creciente y diverso de sistemas de software y tecnologías que no se integraban entre sí.
- Mercado dinámico con necesidades crecientes y enfoque en la calidad de los servicios y productos que requería cambios en los procesos de la empresa con mucha frecuencia.
- Dificultad para dar mantenimiento a los sistemas de software por su alto costo en tiempo y recursos.
- Sistemas muy grandes con muchas funcionalidades pero difíciles de adaptar a los cambios en los procesos por su alta complejidad técnica.

Con el propósito de resolver estos problemas se plantea a la industria de desarrollo de software la necesidad de tener sistemas de software más flexibles por lo cual se requería que estos fueran más desacoplados e interoperables.

Es así como surge, con el desarrollo de internet y los servicios web, lo que se denomina arquitecturas orientadas a servicios que retoma el principio no tan nuevo de "*servicios como unidad de procesamiento lógico*". Con estos servicios se pretende ofrecer a la organización porciones de funcionalidad automatizada más pequeñas de lo que antes ofrecían los grandes y complejos sistemas de software, de manera que si se tiene necesidad de sustituir o modificar un grupo de estas funcionalidades no se afecta al sistema completo. Otra de las ventajas de los servicios es que actúan como interfaces bien definidas y estandarizadas donde la tecnología utilizada para implementar dicha funcionalidad es transparente para el consumidor del servicio. Las arquitecturas orientadas a servicios poseen además las ventajas de la computación distribuida, por lo tanto los servicios pueden funcionar en equipos o dispositivos de hardware diferentes y adaptados a sus necesidades computacionales. Otras de las

ventajas que se derivan del uso de arquitecturas orientadas a servicios son enunciadas por (OASIS, 2006) y (Panian, 2005), de lo cual podemos sintetizar:

- Integración de sistemas, lo cual da lugar a obtención de nuevas funcionalidades aparte de las que ofrecían cada uno de los sistemas de manera independiente.
- La empresa puede proveer nuevos servicios al exterior y e incorporar a su infraestructura funcionalidades provistos por otras empresa en forma de servicios.
- Alineamiento de la tecnología con la estrategia y metas del negocio.
- Menor tiempo para el mercado ya que un servicio que posee grupo de funcionalidades bien concretas y delimitadas puede desarrollarse más rápido que un sistema completo.

Al inicio SOA se concebía únicamente como estilo arquitectónico. Actualmente se concibe como un paradigma, un nuevo enfoque de desarrollo de software ya que generalmente debe venir acompañada de una transformación organizacional en la empresa o instituciones donde se aplique.

SOA difiere del resto de los enfoque de desarrollo de software en que "*no se puede simplemente comprar, es necesario entenderla y vivirla. SOA es un paradigma y una manera de pensar. Es un sistema de valores para arquitectura y diseño"* (M.Josuttis, 2007)*.*

Este enfoque facilita que los sistemas se mantengan escalables y flexibles mientras crecen y permiten una mayor alineación de los sistemas de información con la lógica empresarial, los elementos fundamentales que lo componen son (M.Josuttis, 2007):

- **Servicios**, que representan una funcionalidad de negocio autocontenida que puede ser parte de uno o más procesos y puede ser implementada por cualquier tecnología en cualquier plataforma.
- Una **infraestructura** específica, llamada bus de servicios empresarial (ESB, del Inglés Enterprise Service Bus) que nos permite combinar estos servicios de una manera fácil y flexible.

- **Políticas y procesos**, lo cual tiene que ver con el hecho de que los grandes sistemas distribuidos son heterogéneos, están bajo mantenimiento y tienen diferentes propietarios.

SOA acepta que la única forma de mantener la flexibilidad en los grandes sistemas distribuidos es soportar la heterogeneidad la descentralización y la tolerancia a fallas.

La Universidad de las Ciencias Informáticas, a poco tiempo de su creación, enfrentó el reto de desarrollar sistemas muy grandes y complejos para empresas a nivel internacional igualmente grandes y complejas en las cuales no estaban ausentes los problemas antes mencionados referentes a la flexibilidad de los sistemas, negocios altamente cambiantes y diversidad de tecnologías y proveedores de software que hacen sumamente complejo la interoperabilidad e integración de los sistemas. Como resultado del I Taller de Arquitectura de Software se planteó como una intensión de la Universidad, investigar el uso de tecnologías asociadas con BPM[1] y SOA basadas en software libre (UCI, 2007), materializándose esto con la proyección en el 2008 de un centro que se dedicaría a estudiar y aplicar estas tecnologías y enfoques. En el año 2009 quedó constituido oficialmente el Centro de Consultoría Tecnológica e Integración de Sistemas de la UCI, actualmente Centro de Desarrollo de Arquitecturas Empresariales (CDAE) que contempla entre sus líneas de investigación el enfoque BPM-SOA[2].

Uno de los principales servicios que se propone brindar el CDAE es ayudar a otras empresas a asumir el enfoque BPM-SOA para ello es necesario contar con un proceso de desarrollo, una arquitectura de referencia[3] y un marco tecnológico[4]. Cada uno de estos elementos encierra una gran cantidad de contenidos y problemas por resolver. El Centro de Consultoría avanza paralelamente en la investigación de los tres temas aunque para este trabajo se decidió centrar la atención en uno de los aspectos no resueltos del todo: el

[1] Gestión de Procesos de Negocio, del Inglés Business Process Management

[2] Conjunto de tecnologías, estilos de arquitectura, buenas prácticas y patrones asociados a BPM y a SOA

[3] Conjunto de patrones, buenas practicas y modelos predefinidos para implementar una solución SOA

[4] Conjunto de herramientas que permiten ejecutar los principios de BPM y SOA

proceso de desarrollo SOA, y de este específicamente en el diseño de servicios.

Existen diversas metodologías para el desarrollo de arquitecturas orientadas a servicios. Luego de un estudio sobre cómo algunas de estas metodologías conciben el diseño de servicios se pudo identificar que solo una parte muy pequeña de las metodologías existentes para SOA cubre el diseño de servicios y, de las que lo cubren, muchas son propietarias por lo que no se tiene total disponibilidad de su contenido. El resto presenta una o varias de las siguientes deficiencias:

- No proponen artefactos ni roles.
- Omiten o no son suficientemente explícitas con aspectos importantes en el diseño de servicios como el caso de los principios de orientación a servicios, requisitos no funcionales o atributos de calidad del servicio, diseño de tipos en los mensajes para manejar las fallas o excepciones producidas en la ejecución de los servicios, entre otras.
- Están atadas a una sola tecnología para la realización de SOA como es el caso de los servicios web.

Sobre el estudio realizado acerca de la situación de SOA y del diseño de servicios en Cuba se pudo constatar que en el ámbito empresarial cubano se ha comenzado introducir los beneficios de BPM pero no de SOA. En instituciones académicas y de desarrollo de software se han reconocido y utilizado las ventajas de proveer el software como servicio sin embargo, en casi la totalidad de los casos, estos servicios se desarrollan como servicios web generados directamente a partir de código lo cual puede acarrear problemas de interoperabilidad y no es considerado un buena practica en la orientación a servicios. En ninguno de los casos se ha definido un proceso o procedimiento para desarrollar los servicios con el enfoque "*diseño primero, código después*".

Teniendo en cuenta la situación expuesta anteriormente se define el siguiente **problema de investigación**: ¿Cómo diseñar servicios en proyectos SOA garantizando una adecuada calidad del diseño?

Como **hipótesis de la investigación** se plantea lo siguiente: contar con un proceso definido para diseñar servicios en proyectos SOA que tenga en cuenta los principios de orientación a servicios contribuirá a lograr una adecuada calidad del diseño.

Para dar solución al problema planteado se ha formulado como **objetivo general de la investigación** proponer un proceso de diseño de servicios que tenga en cuenta los principios de orientación a servicios en proyectos SOA.

Por lo que se tiene como **objeto de estudio** el proceso de desarrollo de arquitecturas orientadas a servicios, especificándose como **campo de acción** el diseño de servicios para proyectos bajo el enfoque de arquitectura orientada a servicios.

Para cumplimentar el objetivo general se formularon las siguientes **tareas de investigación**:

- Elaborar el marco teórico de la investigación.
- Elaborar el estado del arte acerca de las metodologías para diseñar servicios en el contexto del desarrollo de arquitecturas orientadas a servicios.
- Analizar la situación en Cuba respecto al enfoque SOA y el diseño de servicios.
- Proponer un proceso para diseñar servicios en proyectos SOA.
- Validar el proceso propuesto mediante su aplicación en un proyecto piloto.
- Evaluar los resultados obtenidos de aplicación del proceso.

A continuación se muestran los **métodos de investigación** que fueron utilizados:

Métodos Teóricos:

- **Análisis histórico – lógico:** para investigar y analizar los orígenes y evolución de SOA con el objetivo de lograr una mejor comprensión del objeto de estudio.

- **Analítico – sintético:** en el estudio de las metodologías, técnicas y estándares existentes para el diseño de servicios.

- **Hipotético deductivo** para el análisis y la definición de la hipótesis de la investigación que será verificada o probada en función del estudio de elementos particulares o de menor nivel de generalidad.
- **Sistémico** para establecer la relación del proceso propuesto con el proceso de desarrollo SOA general.

- **Modelación:** Para crear el proceso de diseño mediante la abstracción de sus elementos fundamentales (tareas, artefactos, guías técnicas y roles) utilizando un lenguaje de modelado de procesos.

Métodos empíricos:

- **Entrevistas y revisión de documentos:** Para conocer el estado del arte del tema en Cuba y el mundo.

- **Aplicación de la propuesta en un proyecto piloto:** Para la validación práctica de la propuesta y el análisis de los resultados obtenidos.

Aporte Teórico – Práctico de la investigación

La investigación se propone contribuir al desarrollo de una parte de una metodología mucho más completa para el desarrollo de arquitecturas empresariales orientadas a servicios. Estas metodologías hoy son prácticamente inaccesibles para nuestro país por su elevado coste (varias decenas de miles de dólares en documentación y capacitación) por lo que desarrollarlas puede contribuir a un ahorro considerable en divisas para el país además de favorecer el desarrollo de conocimiento propio y la independencia tecnológica.

Para el desarrollo de la solución se tuvo en cuenta muchas de las buenas prácticas reconocidas en la industria y la academia con respecto a SOA además de los elementos más significativos de las metodologías estudiadas, lo cual contribuye a lograr un proceso de diseño de igual o mayor calidad que los existentes. El empleo del enfoque de metodologías tradicionales en cuanto a

definición de artefactos, actividades y herramientas unido al enfoque de diseño guiado por principios se revierte en efectos positivos sobre la calidad del producto, en este caso el diseño de de los servicios.

Se desarrollaron artefactos y guías detalladas que favorecen la aplicación del proceso de diseño de servicios por personas poco experimentadas, y contribuyen a un aprendizaje más rápido.

La tesis estará estructurada en tres capítulos. En el primer capítulo se establece un marco teórico conceptual para la investigación y se realiza un estudio de las principales metodologías existentes para el desarrollo de arquitecturas orientadas a servicios con el fin de determinar sus fortalezas y debilidades. Se analiza, además, la situación existente en Cuba y el mundo respecto a SOA y específicamente al diseño de servicios.

En el capítulo 2 se define la relación del proceso de diseño de servicios con otros procesos que están presentes en el desarrollo de proyectos SOA y se describe el proceso propuesto haciendo referencia a las tareas y pasos que lo integran así como a los artefactos, roles, guías y herramientas propuestos.

En el tercer capítulo se describe la validación de la propuesta mediante su aplicación en un proyecto piloto. Primeramente se realiza una evaluación de la adherencia del diseño realizado a cada uno de los principios de orientación a servicios definidos para esta investigación y, posteriormente, se analizan los resultados de las encuestas aplicadas para medir la completitud y comprensión de las especificaciones de diseño, permitiendo esto determinar cómo influye el proceso propuesto en la calidad del diseño de servicios.

Capítulo 1: Fundamentación Teórica del Diseño de Servicios

En este capítulo se establece un marco teórico conceptual para la investigación y se realiza un estudio de los fundamentos teóricos que servirán de base para la propuesta de un proceso de de diseño de servicios. Se analizará el estado actual del tema en Cuba, cómo diferentes metodologías abordan la etapa de diseño de servicios, los principales aspectos servicios web como tecnología más difundida para la implementación de arquitecturas orientadas a servicios y aspectos a tener en cuenta para la calidad del diseño de servicios.

1.1 Arquitectura Orientada a Servicios

Las Arquitectura Orientada a Servicios (SOA, del Inglés Service-Oriented Architecture) ha emergido en este nuevo siglo como un paradigma de desarrollo de software. OASIS, en su Modelo de Referencia para SOA (OASIS, 2006) la define como un paradigma para organizar y utilizar capacidades distribuidas que pueden estar bajo el control de diferentes dominios o entidades (personas u organizaciones). Otros autores (Reynoso, y otros, 2004) la reconocen como un estilo arquitectónico en el desarrollo de software que potencia la descentralización, la interoperabilidad y la flexibilidad.

1.2 SOA en el mundo

El enfoque de orientación a servicios muestra una tendencia creciente y acelerada en el mundo. En un estudio realizado en el 2006 por IBM (Paulson, 2006) se muestra cómo el sector de los servicios juega un papel cada vez más importante en la economía los países desarrollados y emergentes mostrando una tendencia al crecimiento mucho más acelerada a partir de la segunda mitad del siglo XX.

SOA como paradigma de desarrollo de software se ha ajustado muy bien a este enfoque de economía basada en los servicios ya que provee los conceptos que permiten un enlace flexible con las tecnologías existentes. En el artículo “Gestión y Tecnología Orientada a Servicios: Perspectivas sobre la

investigación y la práctica en la década venidera" (Demirkan, y otros, 2008) se plantea lo siguiente:

- El pensamiento orientado a servicios es uno de los paradigmas de más rápido crecimiento entre las tecnologías de la información con relevancia en áreas como contabilidad, finanzas, gestión y operaciones de cadena de suministros y estrategia y mercadeo.
- De acuerdo con Forrester Research[5], las compañías que implementan una arquitectura orientada a servicios son capaces de reducir los costos de integración y mantenimiento en los proyectos en, al menos, un 30% (Wall, 2007).
- Gartner[6] predice como tendencia futura que al menos una tercera parte de los gastos en software y aplicaciones de negocio serán invertidos en "software como servicio" en lugar de licencias de productos para el 2012 y que, además, el 40% de los gastos de capital serán hechos en "infraestructura como servicio" para el 2011 (Plummer, y otros, 2008).

Actualmente se está comenzando a hablar de una nueva ciencia llamada "*Ciencia de los Servicios, Ingeniería y Gestión"*, SSME (del Inglés, Service Science, Management and Engineering).

Esta nueva ciencia se enfoca en el servicio como sistema de interacción de las partes que incluye personas, tecnología y negocio. Sustenta sus ideas en un grupo de disciplinas existentes como son: Ciencia de la Computación, Ciencias Cognitivas, Economía, Comportamiento Organizacional, Gestión de Recursos Humanos, Investigación de Operaciones e Ingeniería de Software (Brittenham, y otros, 2007).

[5] Es una compañía dedicada a la investigación en mercado y tecnología que provee a sus clientes información sobre el impacto de las tecnologías en empresas y consumidores. Tiene centros de investigación en varias ciudades estadounidenses y europeas.

[6] Compañía fundada en 1979. Su actividad fundamental radica en la investigación en tecnologías de la información, en ofrecer consultoría y realizar eventos.

Instituciones académicas, compañías y agencias de gobierno están promocionando, enseñando y adoptando la ciencia de los servicios en diversas formas (Paulson, 2006), ejemplo de ellos son:

- IBM y otras compañías, que tienen sus propios investigadores en este tema y han promocionado la creación de centros que se dediquen a este tema en asociación con universidades y otros centros de investigación.
- Universidades que estudian el tema, como la Universidad de Cambridge en el Reino Unido, Escuela de Economía de Londres y la Technische Universität Darmstadt de Alemania.

Por otra parte la mayoría de las metodologías SOA más completas y probadas son, por lo general, desarrolladas por compañías privadas para la ejecución de sus propios proyectos. Algunas de ellas venden dichas metodologías y los servicios de capacitación como es el caso de CBDI, SofwareAG, Software Associates, IBM, entre otras.

Los precios de subscripción a documentación de estas compañías pueden oscilar en los 1000.00 y 10 000.00 dólares. Los servicios de capacitación y entrenamiento más completos pueden llegar a la cifra de millones de dólares.

1.3 SOA en Cuba

Con el fin de conocer el estado del tema SOA en Cuba, específicamente en cuanto al diseño de servicios, se realizaron entrevistas a directivos y trabajadores vinculados al desarrollo de software en instituciones académicas y empresas de diversos ministerios de nuestro país que se listan a continuación:

- Ministerio de Turismo(MINTUR)
- Empresa TECNOMÁTICA del Ministerio de la Industria Básica (MINBAS)
- SOFTEL, empresa que ofrece soluciones informáticas para el sistema de salud en Cuba
- Universidad Central de las Villas(UCLV)
- Instituto Superior Politécnico José Antonio Echeverría (ISPJAE)
- Universidad de las Ciencias Informáticas (UCI)

En dichas entrevistas se les preguntó si tenían conocimiento sobre la Arquitectura Orientada a Servicios y si la institución tenía alguna experiencia relacionada con el desarrollo o adopción de este tipo de arquitectura o en la provisión de funcionalidades de software como servicios. Además se revisó un amplio número de publicaciones asociadas con el tema principalmente en los centros académicos mencionados referente a las experiencias más cercanas al tema SOA y al diseño de servicios.

En las entrevistas se pudo constatar que el enfoque SOA en la mayoría de los lugares visitados es prácticamente desconocido o se tienen vagas nociones de él.

En el Ministerio del Turismo (MINTUR) se ha adoptado la gestión por procesos como disciplina empresarial en algunas de sus entidades y ha comenzado a utilizar tecnologías como Microsoft SharePoint Server 2007, producto considerado parte de la estrategia de Microsoft en el sector (Chappell, D, 2008).

La empresa TECNOMÁTICA perteneciente al Ministerio de la Industria Básica (MINBAS), desarrolla una tecnología propia para proveer servicios a través de internet mediante el proyecto cubano, Arcoflow, que cuenta ya con algunos resultados aplicados.

SOFTEL, ha definido un grupo de lineamientos para arquitectura de los softwares que se desarrollan para el sector de la salud en los cuales se propicia la provisión de funcionalidades a través de servicios web.

Tanto en la Universidad Central de las Villas como en el ISPJAE existen grupos de investigación y especialista que estudian el tema de gestión por procesos, optimización de flujos de trabajo pero no directamente el tema SOA.

La Universidad de las Ciencias Informáticas a poco tiempo de su creación enfrentó el reto de desarrollar sistemas muy grandes y complejos para empresas a nivel internacional igualmente grandes y complejas en las cuales no estaban ausentes los problemas antes mencionados referentes a la flexibilidad de los sistemas, negocios altamente cambiantes y diversidad de

tecnologías y proveedores de software que hacen sumamente complejo la interoperabilidad e integración de los sistemas. Como resultado del I Taller de Arquitectura de Software planteó como una intensión de la universidad, investigar el uso de tecnologías BPM-SOA[7] basadas en software libre (UCI, 2007) materializándose esto con la proyección en el 2008 de un centro que estudiara y aplicara estas tecnologías y enfoques. En el año 2009 quedó constituido oficialmente el Centro de Consultoría Tecnológica e Integración de Sistemas de la UCI, actualmente Centro de Desarrollo de Arquitecturas Empresariales (CDAE) que contempla entre sus líneas de investigación el enfoque BPM-SOA.

Uno de los principales servicios que se propone brindar el CDAE es ayudar a otras empresas a asumir el enfoque BPM-SOA y para ello es necesario contar con un proceso de desarrollo, una arquitectura de referencia y un marco tecnológico, este último incluye una suite de herramientas para poder ejecutar los principios de BPM y SOA. Cada uno de estos elementos encierra una gran cantidad de contenidos y problemas por resolver entre los cuales se incluye el diseño de servicios para SOA, objetivo principal de esta investigación.

Otros de los intentos por iniciar en la UCI el estudio y aplicación práctica de los temas SOA lo constituyen los trabajos de (Sánchez, y otros, 2008), (Chaviano G., y otros, 2008) y (Rigñack, 2008). En muchos de los grandes y medianos proyectos como el ERP Cubano, Guardián del Alba, y otros para la informatización del sistema de salud cubano y venezolanos así como en proyectos para la automatización de procesos internos de la UCI de gestión académica (proyecto Akademos), y gestión de la residencia, se han provisto funcionalidades como servicios web para facilitar los mecanismos de integración con otras aplicaciones y acceso remoto.

En esencia se ha podido constatar que en el ámbito empresarial cubano se ha comenzado introducir los beneficios de BPM pero no de SOA. En instituciones académicas y de desarrollo de software se han reconocido y utilizado las

[7] Se le llama enfoque BPM-SOA a la aplicación de estos dos paradigmas combinados (BPM, como Gestión de Procesos de Negocio por su siglas del Inglés *Business Process Management* y SOA, Arquitectura Orientada a Servicios por sus siglas del Inglés *Service-Oriented Architecture*).

ventajas de proveer el software como servicio sin embargo, en casi la totalidad de los casos, estos servicios se desarrollan como servicios web generados directamente a partir de código lo cual puede acarrear problemas de interoperabilidad y no es considerado un buena práctica en la orientación a servicios. En ninguno de los casos se ha definido un proceso o procedimiento para desarrollar los servicios con el enfoque *"diseño primero, código después".*

1.4 ¿Qué es el diseño de servicios?

Siempre en el desarrollo de software el diseño ha desempeñado un papel muy importante como etapa de conceptualización previa a la implementación donde se toman decisiones técnicas importantes que contribuirán a al logro de los objetivos del sistema. Esto no es diferente en los proyectos SOA, donde el diseño juega un papel muy significativo en la previsión de defectos que puedan introducirse en etapas posteriores:

> *"Muchos de los problemas que los usuarios han descritos relativos a SOA en la práctica pueden ser asociados a un diseño de servicios inapropiado, lo cual conduce a una pobre reutilización y a modelos de procesos y aplicaciones compuestas innecesariamente complejos"* (Hailstone, 2009).

Los servicios constituyen los componentes esenciales que conforman una arquitectura orientada a servicios; no se puede hablar de diseño de servicios sin proveer una explicación más detallada cómo se estructuran estos y su relación con el resto de los componentes de una SOA.

OASIS en su *Modelo de Referencia para la Arquitectura Orientada a Servicios* (OASIS, 2006) ofrece varios mapas conceptuales en que define los elementos significativo para un servicio independientemente de la tecnología o estándares utilizados para implementar la SOA. Se seleccionó el siguiente mapa conceptual por estar centrado en el concepto *Descripción del Servicio* ya que uno de los objetivos del diseño es precisamente ofrecer descripciones suficientemente detalladas del servicio.

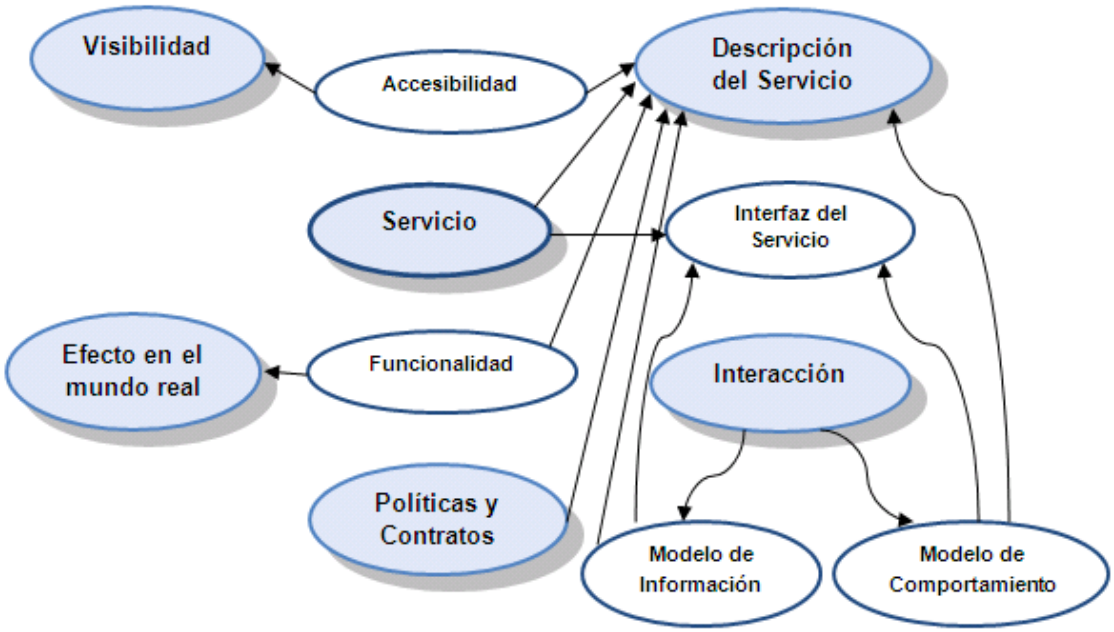

Figura 1: Elementos significativos relacionados con descripción del servicio **(OASIS, 2006)**

Servicio: es un mecanismo para posibilitar el acceso a una o más capacidades del negocio, donde el acceso se provee utilizando una interfaz prescrita la cual es ejercida consistentemente con restricciones y políticas que se especifican en la descripción del servicio. La implementación del servicio se oculta al consumidor del servicio, quien debe acceder a este a través de los modelos de información y comportamiento definidos en la interfaz.

Visibilidad: es la relación entre consumidores y proveedores del servicio, los cuales deben tener la posibilidad de "verse" para poder interactuar entre ellos. La posibilidad de interactuar en ocasiones necesita de datos adicionales como localización del servicio, protocolos, etc., estos elementos forman parte del concepto de accesibilidad.

Efecto en el mundo real: es el resultado que trata de lograr el consumidor del servicio cuando invoca al proveedor del servicio para hacer algo.

Políticas y contratos: las políticas representan condiciones o restricciones en el uso del servicio mientras que los contratos representan un acuerdo entre dos o más partes, en este caso proveedor y consumidor del servicio.

Interacción: la interacción con los servicios involucra la realización de acciones como pueden ser el intercambio de mensajes con dichos servicios o la modificación de recursos compartidos. Los modelos de información y comportamiento definen la forma de interactuar con un servicio en particular.

Descripción del servicio: representa toda la información necesaria para poder usar el servicio. Algunos elementos de la descripción son necesarios para cualquier servicio mientras que otros son necesarios dependiendo del contexto. El propósito de la descripción es facilitar la interacción y la visibilidad sobre todo cuando los participantes en la interacción (consumidor y proveedor) pertenecen a dominios y fabricantes diferentes.

Las buenas prácticas según OASIS sugieren que la descripción del servicio debe realizarse utilizando un formato estándar posible de referenciar por herramientas como los motores de descubrimiento.

Interfaz del servicio: es el medio a través del cual se interactúa con el servicio, incluye protocolos, comandos en intercambio de información mediante los cuales se inician acciones que tienen como resultado algún efecto en el mundo real. Este efecto en el mundo real se debe describir de forma no ambigua en a través de la "funcionalidad" que constituye una porción de la descripción del servicio. Tanto el modelo de información como el de comportamiento tributan a definir la interfaz del servicio.

Modelo de información: constituye una caracterización de la información que puede intercambiarse con el servicio. Define la sintaxis y la semántica de los datos.

Modelo de comportamiento: define el proceso para interactuar con el servicio, por ejemplo si las acciones definidas para el servicio deben invocarse en un orden determinado.

En este modelo OASIS considera solamente los aspectos más abstractos del servicio, o sea los que ve el consumidor, pero es importante tener en cuenta que la interfaz y el resto de los elementos de la descripción del servicio son realizados por componentes de implementación. En ese sentido, Rosen y otros (Rosen, y otros, 2008) definen a un servicio como la combinación de su interfaz (la vista pública del servicio) y su implementación (la vista privada del servicio).

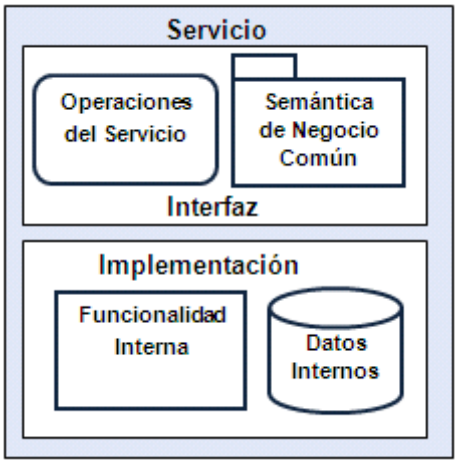

Figura 2 Anatomía de los servicios (Rosen, y otros, 2008)

No todos los autores conciben el diseño de servicios de la misma manera ni incluyen en él las mismas actividades. Algunas de las definiciones que podemos encontrar en la literatura son las siguientes:

Michael Bell, en su libro *SOA Modeling Patterns for Service-Oriented Discovery and Analysis* (Bell, 2010) plantea:

"Una vez que los servicios candidatos han sido presentados, continua la fase de diseño de servicios y arquitectura. La actividad de diseño no solo se enfoca en concretar la composición interna de los servicios sino que trata también los retos del entorno externo, tales como la integración, los mecanismos de intercambio de mensajes y la interoperabilidad de los servicios. Durante estos esfuerzos, nuevos servicios podrán ser identificados para complementar la conceptualización y proposición previa de servicios."

Y continúa:

"Estos nuevos servicios descubiertos ofrecen capacidades para hacer posible una amplio rango de soluciones tales como transformación de datos, mediaciones, seguridad, interfaz de usuario, implementación de lógica de negocio entre otros."

En el mismo libro, Bell plantea además que para documentar la fase de arquitectura y diseño, varios de los roles involucrados en esta fase aplican

patrones y realizan artefactos de modelado también conocidos como especificaciones técnicas.

Thomas Earl (Earl, 2005) concede gran importancia a la etapa de diseño de servicios:

"Las decisiones a nivel de servicios hechas en conjunto durante esta etapa determinarán la calidad y longevidad de la SOA resultante."

Define el diseño de servicios de la siguiente manera:

"El diseño orientado a servicios es el proceso mediante el cual diseños de servicios físicos concretos son derivados de los servicios lógicos candidatos y luego ensamblados en composiciones abstractas que implementan un proceso de negocio."

Además plantea:

"El diseño orientado a servicios es una fase fuertemente orientada al trabajo con estándares que incorpora convenciones de la industria y principios de orientación a servicios dentro del proceso de diseño de servicios.

Esta fase confronta a los diseñadores con decisiones claves que establecen los límites de la lógica encapsulada por los servicios".

Thomas Earl considera que esta etapa puede incluir el diseño de la capa de orquestación de servicios la cual resulta en una definición formal de los procesos de negocio a través de un lenguaje como WS-BPEL.

Una definición de la fase de diseño es ofrecido por Rosen y otros (Rosen, y otros, 2008):

"Durante el diseño se completa la identificación y especificación de servicios. Estas actividades son usualmente llevadas a cabo por un analista de sistemas que crea modelos de análisis y diseño que describen los detalle de la interfaz del servicio y el diseño de implementación, incluyendo los documentos que forman parte de sus interfaces. Un documento de diseño o documento de especificación de servicios puede ser opcionalmente producido. Las interfaces

de servicios pueden ser descritas en WSDL y los esquemas de interfaz pueden ser descritos XSDs."

Por su parte IBM (Amsden, 2007) concibe como parte esencial del diseño realizar las especificaciones de servicio:

> *"Una especificación de servicio debe describir todo lo que un potencial consumidor del servicio debe conocer para saber si está interesado en usar el servicio y saber, exactamente, cómo usarlo. También debe especificar todo lo que necesita el proveedor para implementar el servicio satisfactoriamente. Por lo tanto una especificación de servicio es un mediador o contrato entre lo que el consumidor necesita y el proveedor ofrece."*

Otro elemento al que se hace referencia como parte del diseño o especificación de servicios es a los atributos de calidad del servicio:

Un atributo de calidad es una propiedad de un sistema mediante la cual algunos interesados juzgarán su calidad. Los requerimientos de atributos de calidad tales como desempeño, seguridad, modificabilidad, confiabilidad y usabilidad tienen una influencia significativa en la arquitectura de software de un sistema (SEI, 2007).

Aunque este concepto fue pensado para sistemas de software tradicionales también se aplica a los servicios en una arquitectura orientada a servicios, con algunas diferencias ya que para el enfoque orientado a servicios ciertos atributos toman mayor relevancia porque debe invertirse mayor esfuerzo para alcanzarlos mientras que otros ya son soportados intrínsecamente por el enfoque SOA como se describe en (O'Brien, y otros, 2005), donde califican con valor "rojo" o crítico los atributos de rendimiento, seguridad, facilidad de prueba, y auditabilidad mientras que califica como "verdes" o más maduros para SOA los de interoperabilidad, extensibilidad y modificabilidad.

En la literatura consultada y las definiciones expuestas anteriormente se puede apreciar para el diseño de servicios una delimitación difusa con respecto a tareas del diseño arquitectónico e implementación. Por tal motivo se considera

útil establecer la definición del término diseño de servicios para este trabajo, teniendo en cuenta los elementos significativos de las definiciones anteriores:

> *El diseño de servicios es un proceso mediante el cual se derivan servicios concretos a partir de los servicios iniciales provenientes de las etapas de análisis y arquitectura. Se realiza la especificación de los aspectos funcionales y no funcionales de estos necesarios para ser implementados y, suficientemente entendibles, para ser consumidos. Se aplica a los servicios estándares definidos por el diseño de la arquitectura y patrones garantizando que cumplan, en lo posible con los principios de orientación a servicios.*

Como elementos indispensables a incluir en el diseño del servicio se consideran los siguientes:

- **Descripción funcional:** propósito y operaciones que provee el servicio
- Modelo de información del servicio: tipos de datos, mensajes de entrada y salida utilizadas por las operaciones del servicio
- **Requisitos de calidad del servicio:** cualidades que debe esperar el consumidor y que debe ofrecer el proveedor en términos de requisitos no funcionales, por ejemplo disponibilidad, fiabilidad, rendimiento, y seguridad.
- **Dependencia del servicio y flujos de ejecución:** dependencias que tiene el servicio de otros servicios, descripción del orden en que deben llamarse ciertas operaciones, en caso que existan restricciones de este tipo para lograr consistencia en la funcionalidad del servicio y, para los servicios compuestos, debe representarse los flujos de ejecución necesarios para llevar a cabo ciertas acciones mediante la invocación a otros servicios.
- **Restricciones de uso del servicio y políticas:** pueden ser restricciones sobre validación de los datos que sirven como entrada al servicio, políticas de seguridad en cuanto a encriptación de la información, confiabilidad en la entrega de mensajes, etc. Todas estas restricciones y políticas funcionan como reglas que se relacionan o inciden en el servicio por los cual se declaran en como parte del diseño

de este pero por lo general se implementan (en caso de ser automatizables) fuera de él.

1.5 Procesos de ingeniería de software y diseño de servicios

El SEI (SEI, 2006) define tres dimensiones fundamentales en una empresa para alcanzar una mejora y lograr los objetivos del negocio, estas dimensiones son: las personas, las herramientas y equipamiento y los procedimientos como se muestra en la Figura 3.

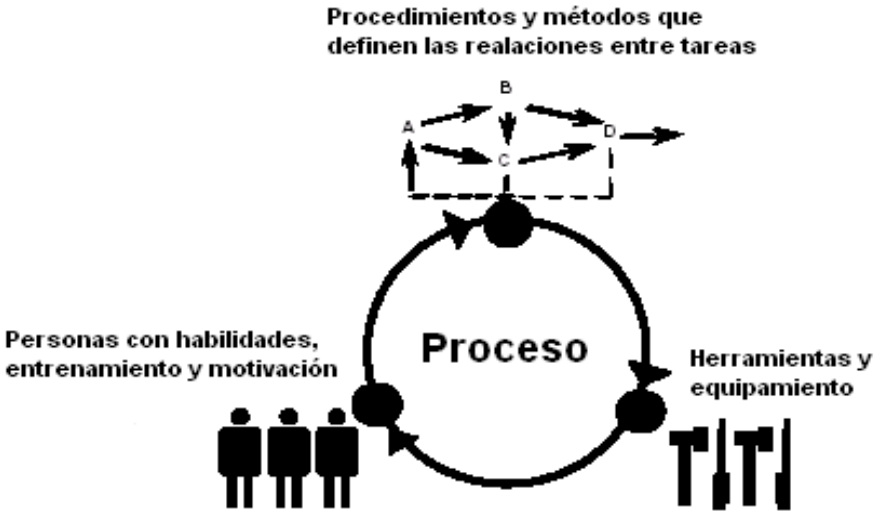

Figura 3. Dimensiones críticas para la empresa **(SEI, 2006)**.

Los procesos de la organización constituyen el elemento aglutinador y coordinador de otros elementos. Los procesos que resultan muy grandes o complejos se pueden subdividir en varios subprocesos.

Un subproceso forma parte de un proceso mucho más largo y puede, a su vez ser descompuesto en otros subprocesos y/o elementos de procesos (SEI, 2007). Esto ocurre también con los procesos ingenieriles como el que se trata en esta investigación que también son considerados por otros autores (Zhu, y otros, 2007) como micro-procesos.

Por tanto podemos concebir el diseño de servicios como un subproceso intermedio del proceso de desarrollo de una arquitectura orientada a servicios y este está fuertemente relacionado con otros subprocesos como son los procesos de análisis del negocio, de arquitectura y planificación de servicios,

gobierno SOA y de implementación y pruebas de servicios. Estos subprocesos se describen en varias metodologías (M.Josuttis, 2007), (IBM, 2007), (Earl, 2005) (CBDI, 2007), (Microsoft, 2007) aunque pueden tener nombre diferentes pero con la misma esencia en cuanto a contenido y todos ellos contribuyen a la realización de las diferentes etapas del ciclo de vida de los servicios.

1.6 Metodologías SOA y diseño de servicios

Actualmente se puede observar un gran número de metodologías provenientes tanto de la industria como de la academia para adoptar el enfoque orientado a servicios. En trabajos como (Ramollari, y otros, 2007), (Ricken, y otros, 2009), (Kohlborn, y otros, 2009) y (Klose, y otros, 2007)se hace un estudio de varias de estas metodologías teniendo en cuenta el ciclo completo del desarrollo de proyectos SOA y son analizadas de acuerdo a los siguientes parámetros:

- **Estrategia de desarrollo**: especifica si el proceso de descubrimiento y desarrollo de los servicios utilizado por la metodología es ascendente, descendente o medio (del Inglés top-down, bottom-up y meet-in-the-middle) refiriéndose al punto de partida para la identificación y desarrollo de los servicios. [8]
- **Cubrimiento del ciclo de vida de los servicios:** especifica qué fases del ciclo de vida de los servicios cubre la metodología (análisis y diseño, implementación, prueba, despliegue, etc.)
- **Grado de prescripción:** representa el nivel de detalle con que se describe la metodología y cuan restrictiva o flexible es. Las metodologías exhaustivamente detalladas o muy prescriptivas tienden a ser menos flexibles.

[8] En los métodos descendentes (top-down) los servicios se identifican a partir de los modelos y procesos de negocio existentes en la empresa. En los métodos ascendentes (bottom-up) se parte de los sistemas legados para identificar servicios y en los mixtos (meet-in-the-middle) se van identificando y desarrollando según las necesidades del negocio hallando un equilibrio entre los métodos anteriores.

- **Disponibilidad del contenido:** las metodologías que son propietarias por lo general no ofrecen abiertamente su contenido y se hace más difícil su análisis y comparación.
- **Métodos y Técnicas:** métodos y técnicas detalladas que propone o reutiliza sobre cómo efectuar diferentes actividades para alcanzar el enfoque orientado a servicios.
- **Aplicación industrial:** si ha sido aplicada en proyectos reales a nivel industrial.
- **Entre otros.**

En estos estudios encontramos información útil, como por ejemplo que algunas de estas metodologías no cubren todo el ciclo de vida de los servicios, en especial el diseño y muchas son propietarias. Sin embargo para analizar su capacidad e idoneidad para el diseño de servicios es necesario realizar un análisis de mayor profundidad en este aspecto, para lo cual hemos seleccionado un subgrupo de ellas. La selección fue hecha de teniendo en cuenta los siguientes aspectos:

- Cubrimiento de la fase de diseño de servicios.
- Disponibilidad del contenido.
- Influencia en otras publicaciones y reconocimiento internacional: aunque no se tengan referencias directas de su aplicación en la industria, consideramos prudente incluir en el análisis aquellas metodologías que expresan un nivel de detalle aceptable para su entendimiento y sus autores tienen reconocimiento internacional en las publicaciones sobre el tema.

A continuación se realiza el análisis de las metodologías seleccionadas.

RUP-SOMA de IBM (IBM, 2007): esta metodología surge de la integración entre el Proceso Unificado de Rational y la metodología SOMA (del Inglés Service Oriented Modeling and Architecture) de IBM. Cubre las áreas de análisis y diseño incluyendo diseño de la arquitectura, no así las de implementación y despliegue de los servicios. Consta de cuatro fases: análisis

de transformación empresarial, seguidas por las fases de identificación, especificación y realización que se suceden de manera iterativa.

La fase de especificación contiene, entre otros, el flujo de trabajo *Realizar Especificación de Servicios*, el cual se describe a continuación:

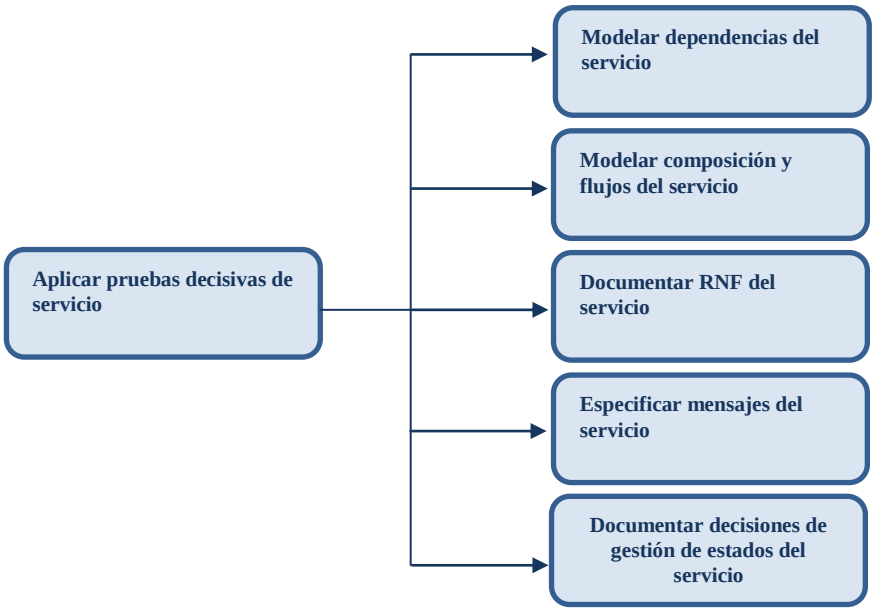

Figura 4. Flujo de trabajo *Realizar Especificación de Servicios* de RUP-SOMA.

La actividad "*aplicar pruebas decisivas de servicio"* consiste en determinar a través de una serie de comprobaciones cuales de los servicios catalogados como "candidatos" en la fase de *Identificación* son seleccionados para exponerse con una interfaz de servicio descubrible (por ejemplo WSDL).

Las actividades que siguen se pueden realizar sin un orden predefinido como se muestra en la Figura 4 desde "*modelar dependencias del servicio"* hasta "*documentar decisiones de gestión de estados del servicio".*

- **Modelar dependencias de servicios:** se modela la relación de un servicio con otros servicios a través de diagramas de estructura compuesta o de clases y luego más detalladamente.
- **Modelar composición de flujos:** se representan las secuencias de interacciones que son necesarias para realizar los servicios compuestos utilizando diagramas de interacción de actividad o secuencia. Durante esta actividad también se representan los

enlaces (codificaciones de mensajes y protocolos físicos reales, por ejemplo HTTP, SOAP, RMI, etc.) que materializan la comunicación entre servicios.

- **Documentar RNF del servicio:** se mencionan conceptos importantes a tener en cuenta para la especificación de requisitos no funcionales del servicio pero no se especifica su relación con la especificación de servicio ni se identifica todos los tipos de RNF posibles a documentar. En esta tarea se describe el "qué" se debe hacer pero no el "cómo".
- **Especificar mensajes del servicio:** constituye una de las actividades más detalladas en la que se ofrece un número de consejos o recomendaciones para modelar los mensaje de entrada y salida del servicio. Define una serie de pasos como son utilizar estándares de mensaje, reutilización del modelo de dominio, comprender los patrones de intercambio de mensajes, gestionar granularidad de mensajes, gestionar el rendimiento del intercambio de mensajes. No se hace mención a los mensajes de error que pueda proveer el servicio.
- **Documentar decisiones de gestión de estados del servicio:** mediante esta actividad y la directriz *"Gestión de Estado para Servicios"* se exploran los problemas que rodean al uso de servicios con estado y sin estado en relación con su efecto en el rendimiento y la solidez, y también cuando la interacción requiere un enfoque transaccional. Se considera que la noción de gestión de estado en un entorno de arquitectura orientada a servicios incluye tres categorías principales: estado de transacción, estado de seguridad y estado funcional.

En sentido general se puede apreciar que el proceso RUP-SOMA de IBM es un proceso bastante detallado para la realización del análisis y diseño de servicios en SOA. Define actividades y tareas y pasos que unido a las guías y directrices ayudan a la realización de múltiples tareas. También presenta roles definidos y artefactos en los cuales se refleja el resultado del diseño. Sin embargo para el

uso práctico de esta metodología se presentan algunas dificultades que se mencionan a continuación:

- Es de carácter propietario por lo que no se dispone del total del contenido ni de sus actualizaciones, esto incide también en que en algunas actividades se define el "qué" se debe hacer pero no el "cómo"
- La fase de especificación, a la cual se circunscribe en su mayoría el diseño de servicios tal como se concibe para esta investigación, consta de un amplio número de actividades y tareas sin un orden definido y no se ofrece una orientación, al menos tentativa, sobre qué es más conveniente hacer primero y qué después lo cual puede llevar a confusiones al diseñador con poca experiencia.
- No se abordan procedimientos para la elaboración de interfaces técnicas del servicio como son WSDL y XSD, entre otros estándares para servicios web, interfaces para servicios con tecnologías Java, etc. Estas interfaces técnicas son necesarias para la posterior implementación de componentes que realizan el servicio.

Por otra parte esta metodología no enfatiza en los principios de orientación a servicios, y aunque esto no es obligatorio ya que por diversas causas se puede considerar necesario relajar el cumplimiento de alguno de estos principios para satisfacer necesidades inmediatas de rendimiento, seguridad, etc., resulta importante tenerlos como meta ya que de ellos se derivan las ventajas de SOA.

Metodología de Papazoglou y otros (Papazoglou, y otros, 2006): propone una metodología para diseño y desarrollo de servicios desde el punto de vista tanto del consumidor como del proveedor del servicio. Está basada en RUP, CBD[9] y BPM. Intenta cubrir el ciclo de vida completo de SOA mediante un proceso iterativo e incremental.

[9] Desarrollo Basado en Componentes, del Inglés Component Based Development

Figura 5. *Fases de la metodología de diseño y desarrollo orientado a servicios* (Papazoglou, y otros, 2006).

Durante la fase de diseño los servicios y procesos conceptuales son transformados a un conjunto de interfaces independientes de la plataforma. Se enfatiza en características del diseño como la granularidad, reutilización, capacidad de composiciónión, cohesión y acoplamiento de los servicios y está fuertemente orientada a servicios web. No define roles ni artefactos, ni tampoco cómo llevar a cabo en detalle la fase de diseño sino que ofrece un grupo de recomendaciones para realizar la especificación de servicios a través de:

- **Especificación estructural:** define tipos, mensajes, y operaciones para el servicio.
- **Especificación de comportamiento:** define secuencias obligatorias en el comportamiento del servicio, por ejemplo para un servicio *Gestión de Pedidos* no se puede invocar la operación *Cancelar Pedido* si antes no se invocó *Crear Pedido*.
- **Especificación de políticas:** indica declaraciones y restricciones sobre el servicio para gestionar aspectos como la seguridad, entre otros.

Metodología de Thomas Earl[10] (Earl, 2005)**:** la metodología de análisis y diseño orientado a servicios documentada en el libro *SOA: Concepts, Tecnology and Design* se considera como la primera en ser publicada de manera independiente o neutral a los proveedores de tecnologías. Constituye una guía "paso a paso" que describe una serie de fases del ciclo de vida de una SOA:

- Diseño orientado a servicios
- Desarrollo de servicios
- Prueba de servicios
- Despliegue de servicios
- Administración de servicios

En la fase de análisis se identifican servicios candidatos a partir de modelos de negocio o de los sistemas legados de la empresa y luego estos servicios son realizados como servicios web en la fase de diseño. Esta fase se realiza a partir del diseño secuencial de tres tipos básicos de servicios: servicios de entidad, servicios de aplicación y servicios de tareas (Anexo 3).

El proceso descrito por Thomas Earl resulta bastante detallado en cuanto a tareas y pasos para realizar cada actividad haciendo mención además a buenas prácticas a tener en cuenta. Provee una descripción detallada de cómo crear servicios web pero queda atado solamente a este tipo de tecnología para la realización de SOA. No define explícitamente artefactos de entrada y salida al proceso de manera que dificulta la formalización de la especificación de servicios resultante así como la información necesaria de entrada al proceso de diseño.

Proceso de CBDI-SAE (Allen, 2007): el Fórum de CBDI se encuentra actualmente desarrollando una metodología como parte de su Framework de Referencia para SOA. Esta metodología cuenta con cuatro disciplinas: Consumo, Provisión, Gestión y Habilitación. Cada una de estas disciplinas se

[10]Es considerado uno de los pioneros de SOA. Se destaca por la enunciación de los principios de orientación a servicios que han constituido uno de los pilares para la definición de SOA. Ha escrito varios libros sobre arquitectura orientada a servicios, sus conceptos y desarrollo en la práctica.

divide en unidades de proceso y estas a su vez en tareas. El proceso de CBDI-SAE cubre el ciclo completo de desarrollo SOA incluyendo actividades de despliegue, monitoreo y gobierno.

Respecto al diseño de servicios, CBDI-SAE, lo ubica como una unidad de proceso en la disciplina de Provisión y los divide en dos tareas: Especificar Servicios y Certificar Servicios. Debido a que esta metodología es propietaria por lo cual no se tiene acceso al contenido de dichas tareas, sin embargo se tiene conocimiento del artefacto que propone para especificar servicios el cual es bastante detallado ya que provee una descripción exhaustiva del servicio una vez concluido el diseño de este.

En resumen se puede constatar que de las metodologías SOA presentes en la literatura consultada solo una parte muy pequeña de ellas cubre el diseño de servicios, de estas algunas son propietarias por lo que no se tiene total disponibilidad del contenido como es el caso del Proceso de CBDI-SAE y el resto presenta una o varias de las siguientes deficiencias:

- No disponibilidad de artefactos y roles.
- Omiten o no son suficientemente explícitas con aspectos importantes en el diseño de servicios como el caso de los principios de orientación a servicios, requisitos no funcionales o atributos de calidad del servicio, diseño de mensajes de error en los servicios, etc.
- Están atadas a una sola tecnología para la realización de SOA como es el caso de los servicios web.

Se tomarán como referencia fundamental para la elaboración del proceso de diseño propuesto:

RUP-SOMA de IBM: por el nivel de detalle en la descripción de algunas actividades, técnicas, recomendaciones y buenas prácticas en el diseño de servicios

Proceso de Diseño de Thomas Earl: fundamentalmente como guía para la aplicación de los principios de orientación a servicios.

Proceso CBDI-SAE: se tomará como referencia el artefacto *Especificación de Servicio* por su nivel de detalle en la descripción de los elementos relevantes para la posterior implementación del servicio.

1.7 Calidad del diseño de servicios

La calidad siempre ha sido un término difícil de definir. En el proceso de desarrollo de software este problema se acentúa debido a la naturaleza intangible del software lo cual hace más complejo la medición de sus propiedades o atributos y por tanto el establecimiento de estándares adecuados para dichos atributos.

Algunos aspectos importantes a tener en cuenta cuando se habla y se intenta medir la calidad son los siguientes (IBM, 2003):

- La calidad no es de una sola dimensión, atributo o característica.
- No se puede lograr sostenidamente si no se describe, no se mide y no parte del proceso de crear el producto.

Respecto a la calidad Pressman plantea (Pressman, 2005):

"Cuando se examina un elemento según sus características mensurables, se puede encontrar dos tipos de calidad: calidad del diseño y calidad del concordancia."

Y agrega:

"En el desarrollo de software la calidad del diseño comprende los requisitos, especificaciones y el diseño del sistema. La calidad de concordancia es un aspecto centrado principalmente en la implementación. Si la implementación sigue el diseño, y el sistema resultante cumple los objetivos de requisitos y de rendimiento, la calidad de concordancia es alta."

La calidad en el diseño de servicios se corresponde con el concepto de calidad de diseño mientras que el cumplimiento de las funcionalidades y atributos de calidad especificados en las descripciones de los servicios se corresponden con el diseño de concordancia ya que solo pueden ser verificados una vez que el servicio ha sido implementado.

El diseño de servicios guiado por una serie de principios ha sido bien documentado en la literatura (Erl, 2007), (Artus, 2006) :

- **Los servicios deben ser reutilizables:** la lógica encapsulada por el servicio debe ser suficientemente genérica como para ser empleada en numerosos escenarios de uso y por varios consumidores.
- **Los servicios deben proporcionar un contrato estandarizado:** deben expresar su propósito y capacidades a través de un contrato de servicio o interfaces técnicas mediante las cuales puedan ser accedidos y estas últimas deben responder a un estándar de definición de contrato.
- **Los servicios deben tener bajo acoplamiento:** el bajo acoplamiento se debe manifestar en el hecho de que un contrato de servicio sea independiente de la implementación del servicio y que los servicios sean independientes unos de otros.
- **Los servicios deben tener alta cohesión:** las funcionalidades provistas por el servicio deberán estar fuertemente relacionadas.
- **Los servicios deben ser abstractos:** este principio hace hincapié en la necesidad de ocultar los detalles fundamentales sobre la tecnología y la lógica interna a los consumidores del servicio tanto como sea posible.
- **Los servicios deben permitir la composición:** todo servicio debe ser construido de tal manera que pueda ser utilizado para construir servicios genéricos de más alto nivel de abstracción o de granularidad más gruesa.
- **Los servicios deben de ser autónomos:** los servicios deben ejercer tanto control como pueden sobre su entorno de ejecución y recursos. De esta manera, el servicio es totalmente independiente y se potencia su reutilización.
- **Los servicios no deben tener estado:** un servicio no debe guardar ningún tipo de información. Entre llamadas consecutivas al servicio, no debe ser responsabilidad de este mantener ninguna información.

- **Los servicios deben poder ser descubiertos:** todo servicio debe poder ser descubierto de alguna forma para que pueda ser utilizado, esto contribuye a evitar la creación accidental de servicios que

proporcionen las mismas funcionalidades. Los contratos de servicios contienen metadatos mediante los cuales el servicio puede ser descubierto y evaluado.

Debido a la ausencia de un modelo de calidad estandarizado para SOA que incluya aspectos de calidad del diseño, el cumplimiento de estos principios puede considerarse un buen indicador de calidad en el diseño de los servicios y además poseen la ventaja de poder verificarse antes de implementar los servicios.

Para medir la adherencia a algunos de estos principios, en (Sindhgatta, y otros, 2009) se han definido un conjunto de métricas las cuales han sido empíricamente validadas.

Por otra parte, para garantizar la calidad del diseño es necesario también garantizar la calidad de las especificaciones:

"El diseño deberá ser una guía legible y comprensible para aquellos que generan código y para aquellos que comprueban y consecuentemente dan soporte al software" (McGlaughlin, 1991).

Esto puede ser extendido al diseño del servicio con el fin de garantizar que las especificaciones producidas como parte del diseño sean comprensibles, y no ambiguas para los implementadores y probadores los servicios. Además el producto del diseño debe proveer todos los elementos que son necesarios para llevar a cabo dicha implementación lo cual se puede considerar cómo completitud del producto de diseño.

1.8 Tecnologías para materializar arquitecturas orientadas a servicios

Para materializar los principios de orientación a servicios definidos en el epígrafe anterior es necesario contar con tecnologías (herramientas y estándares) que lo propicien. El diseño de servicios debe realizar sus especificaciones sobre la tecnología seleccionada para crear los servicios.

Servicios Web

Los servicios web constituyen la tecnología más popular para la materialización de SOA. El World Wide Web Consortium (W3C) define un servicio web como una aplicación software identificada por un URI cuyas interfaces se pueden definir, describir y descubrir mediante documentos XML. Los servicios web permiten la interacción de sistemas distribuidos heterogéneos con independencia de las plataformas hardware y software empleados.

Puede pensarse en ellos como en una arquitectura, conceptual y tecnológica, que hace posible que distintos servicios se describan, publiquen, descubran y utilicen a través de sistemas distribuidos, empleando la infraestructura proporcionada por Internet. Este enfoque está avalado por la aparición de un gran número de recomendaciones y estándares, que han sido impulsados por los grandes actores empresariales del sector de las TIC, en los que XML forma la espina dorsal de estas tecnologías (Domínguez Jimenez, y otros, 2007).

El uso de servicios web como opción tecnológica para la implementación de SOA presenta ventajas y desventajas:

Ventajas

- Diversifican las oportunidades de negocio al facilitar que aparezcan escenarios de libre intercambio de servicios normalizados.
- Proporcionan un acoplamiento débil para la operación entre plataformas con distinto hardware, sistema operativo o que empleen distintos lenguajes de programación en sus aplicaciones.
- Los servicios son fáciles de reutilizar y reemplazar, lo que produce una reducción de costes, especialmente del de mantenimiento
- Pueden funcionar síncrona o asíncronamente, según las necesidades.

Desventajas

- Su rendimiento es, por lo general, bastante inferior al de otras arquitecturas distribuidas clásicas

- Necesitan una serie de tecnologías y especificaciones adicionales para poder operar de manera adecuada (con seguridad, fiabilidad, etc.).

Organizaciones como la W3C y OASIS se dedican a realizar y promover estándares relacionados con servicios web y mantienen una relación exhaustiva de informes técnicos, borradores, recomendaciones y estándares (World Wide Web Consortium, 2007), (OASIS, 2007).

La proliferación de estándares y especificaciones para los servicios web en los últimos años es asombrosa, se dice que se trata de más de ciento cincuenta de ellos. A este grupo de estándares se le conoce como WS-* (Weerawarana, y otros, 2005). En el Anexo 1 se puede apreciar una breve descripción de los un grupo de los estándares y especificaciones más utilizados.

Una de las características más deseables de lograr cuando se decide usar servicios web es la interoperabilidad, como capacidad de acceder a las funcionalidades de componentes de software desarrollados en diferentes plataformas, frameworks de mediación, lenguajes de programación, herramientas y sistemas operativos. Aunque la interoperabilidad ha constituido una de las principales promesas de los servicios web lamentablemente todavía existen dificultades para lograrla.

En el 2002 se creó la Organización para la Interoperabilidad de los Servicios Web, **WS-I** (del Inglés, Web Services Interoperability Organization)[11]. Hasta el momento WS-I ha producido una serie de perfiles. Un perfil consiste en una lista de especificaciones de servicios web de una versión determinada acompañado de lineamientos recomendados para el uso o exclusión de elementos definidos inadecuadamente en dichas especificaciones.

[11] Organización fundada por un grupo de las principales compañías que desarrollan y promueven la tecnología de servicios web, entre sus colaboradores se encuentran Internet Engineering Task Force (IETF), Open Applications Group (OAGi), OASIS, OMG, UDDI, W3C, entre otras. Entre los propósitos de WS-I está la creación de perfiles, herramientas de prueba, escenarios de uso y ejemplos de aplicaciones de los estándares y especificaciones de servicios web para garantizar la interoperabilidad.

Thomas Earl (Earl, 2005) recomienda entre las buenas prácticas para el diseño de servicios, usar los perfiles de WS-I aun cuando la conformidad con este no sea un requisito declarado. Argumenta que son estándares bien investigados y probados por lo que pueden ahorrar esfuerzo y tiempo cuando se desarrollen estándares propios. En el Anexo 2 se resumen los perfiles desarrollados hasta el momento por WS-I y el propósito de cada uno.

Existen algunas herramientas para el trabajo con servicios web y su conformidad con los perfiles de WS-I. Algunas de las herramientas libres más utilizadas son:

- SOAP UI: plug-in de Eclipse para validar conformidad con perfiles WS-I. Permite chequear la conformidad con varios perfiles como son Basic Profile 1.1, Simple SOAP Binding Profile 1.0 y Attachments Profile 1.0.
- WS-I Testing Tool Analyzer: permite evaluar la conformidad con el Basic Profile 1.0. Actualmente en el sitio de WS-I, están disponible para descargar versiones desarrolladas en C# y Java.

Es recomendable usar el plug-in SOAPUI de eclipse ya que tiene una interfaz más amigable y permite la validación de un número mayor de perfiles. Además si se desarrollan los servicios web con eclipse tiene la ventaja de estar integrado con el entorno de desarrollo.

REST

Otra de las variantes para la realización de SOA es REST, Transferencia de Estado Representacional (del Inglés, Representational State Transfer). Consiste en un estilo arquitectural pensado para sistemas distribuidos que operan a través de la web. Tiene como origen la tesis doctoral presentada por Roy Fielding[12] en el año 2000 (Fielding, 2000).

Un concepto básico de REST es el de recurso. Un recurso es una pieza de información que se puede identificar de manera única. Los servicios intercambian mensajes que contienen datos y metadatos. La parte que

[12] Es uno de los principales autores de la especificación del protocolo HTTP y autor del término REST (Representational State Transfer) y de los principios que guían este concepto.

corresponde a los datos en un mensaje constituye una representación particular del recurso y la de los metadatos, describe el formato y puede contener directivas de procesamiento sobre el recurso. Se puede intercambiar un recurso en múltiples representaciones. REST asume la interfaz simple y genérica que ofrece el protocolo HTTP para acceder a los recursos mediante los métodos CRUD[13] (crear, leer, actualizar y borrar) (Weerawarana, y otros, 2005).

Ventajas

- Las interfaces que ofrecen los recursos son sencillas los cual potencia la escalabilidad.
- Puede resultar beneficiosa para el rendimiento ya que los mensajes tienden a ser más ligeros que en otras tecnologías, por ejemplo los mensajes SOAP en servicios web.

Desventajas

- Es un paradigma bastante nuevo, las herramientas para implementarlo no son muchas ni poseen una madurez adecuada.
- Aún tiene un soporte rudimentario para atributos de calidad como transporte confiable de mensajes, interacciones de tipo transaccional, encriptación selectiva de partes del mensaje, etc.

En muchas publicaciones se contrapone REST y Servicios Web tratando de argumentar cual es mejor, sin embargo estos dos representan paradigmas diferentes para implementar SOA, el primero está enfocado en el acceso a recursos, el segundo se enfoca en operaciones que debe ejecutar el proveedor de servicios. REST resulta un buena opción para accede a recursos estáticos a través de internet, mientras que servicios web resulta una buena opción para la implementación de servicios a lo interno de la empresa, potenciando la seguridad, confiabilidad, entre otros atributos de calidad de los servicios (MacVittie, 2006).

[13] Del Inglés Cretaing, Reading, Updating, Deleting (CRUD).

También es posible implementar SOA con otras tecnologías para sistemas distribuidos como son CORBA, DCOM, EJB, etc. sin embargo estas tecnologías por lo general omiten o no soportan algunos de los principios de orientación a servicios como el descubrimiento de servicios, ausencia de estado en los servicios, etc., pero pueden ser una buena opción cuando se necesita potenciar otros atributos como el tiempo de respuesta.

1.9 Consideraciones del capítulo

- En este capítulo se abordaron los principales concepto asociados a la arquitectura orientada a servicios que sirven de fundamento para ofrecer una solución al problema de investigación planteado.
- Se analizó el estado actual de SOA y el diseño de servicios en Cuba y el mundo con lo cual se pudo constatar que existe un interés creciente en la investigación y desarrollo por parte de las grandes empresas del sector de la TIC y de diversas instituciones académicas debido a los beneficios que ofrece. En Cuba, sin embargo, su conocimiento es bastante reciente y no se tiene conocimiento de la aplicación completa y exitosa, de un proyecto SOA pero se comienza a manifestar la necesidad de aplicación de este paradigma al desarrollo de arquitecturas empresariales de gran magnitud.
- Se analizaron varias de la principales metodologías de desarrollo SOA desde la perspectiva del diseño de servicios concluyendo que:
    - Solo unas pocas cubren detalladamente el diseño de servicios.
    - Muchas son propietarias con altos precios para acceder a su contenido, motivo por el cual se limita en gran medida el acceso para nuestro país.
    - En las que se dispone de su contenido y cubren la fase de diseño de servicios, existen aún deficiencias y limitaciones para su aplicabilidad en proyectos con enfoque de orientación a servicios.
- Se decidió tomar como referencia fundamental para la elaboración del proceso de diseño propuesto las siguientes metodologías:
    - **RUP-SOMA de IBM:** por el nivel de detalle en la descripción de algunas actividades, técnicas, recomendaciones y buenas prácticas en el diseño de servicios.

  - **Proceso de Diseño de Thomas Earl:** fundamentalmente como guía para la aplicación de los principios de orientación a servicios.
  - **Proceso CBDI-SAE:** Se tomará como referencia el artefacto *Especificación de Servicio* por su nivel de detalle en la descripción de los elementos relevantes para la posterior implementación del servicio.

- Se reconocieron como guías para evaluar la calidad del diseño de servicios la adherencia a los principios de orientación a servicios y la legibilidad, comprensión y no ambigüedad de la especificación de diseño.
- Se estudiaron varias de las tecnologías para implementar SOA reconociendo las debilidades y ventajas de cada una, destacándose la de *servicios web* como una de las más populares y con mayores potencialidades para realizar todos los principios de orientación a servicios.

Capítulo 2: Proceso de Diseño de Servicios Propuesto

En este capítulo se describe la solución propuesta que consiste en la definición de un proceso para diseñar servicios en proyectos SOA.

2.1 Descripción general del proceso de diseño

Para la modelación del proceso se utilizó el lenguaje SPEM creado particularmente para describir procesos de software. Los elementos de lenguaje que se utilizaron fundamentalmente fueron:

- **Tareas:** constituyen unidades atómicas de trabajo que son realizadas por uno o varios roles. Por lo general requieren artefactos de entrada que contienen alguna información útil para realizar dicho trabajo y generan artefactos de salida. En su desarrollo se pueden utilizar o no herramientas y tener guías asociadas de manera opcional. Cada tarea se describe mediante un propósito y una secuencia de pasos.
- **Pasos:** constituyen una porción de trabajo a realizar por algún rol dentro de una tarea.
- **Guías:** son documentos explicativos que ayudan a describir detalladamente cómo se realiza una tarea o paso. Pueden ser de varios tipos. Los tipos de guías utilizados en la propuesta son *Directriz* y *Ayuda de Herramienta*.
- **Roles:** en este contexto representan un conjunto de habilidades, competencias y responsabilidades que debe poseer un individuo o conjunto de individuos para realizar una tarea o paso.
- **Herramientas:** representan herramientas de software que sirven de apoyo a uno o varios roles en la realización de una tarea.
- **Artefactos:** constituyen productos de trabajo tangibles por ejemplo documentos, modelos UML, archivos de código, etc.

Como resultado del análisis realizado en el capítulo anterior a las diferentes metodologías SOA y, atendiendo a los elementos más útiles encontrados en ellas, se toman como referencia fundamental para la elaboración del proceso de diseño propuesto:

- **RUP-SOMA de IBM:** por el nivel de detalle en la descripción de algunas actividades, técnicas, recomendaciones y buenas prácticas en el diseño de servicios
- **Proceso de Diseño de Thomas Earl:** Fundamentalmente como guía para la aplicación de los principios de orientación a servicios.
- **Proceso CBDI-SAE:** Se tomará como referencia el artefacto Especificación de Servicio por su nivel de detalle en la descripción de los elementos relevantes para la posterior implementación del servicio.

La adopción de un enfoque SOA en una empresa es un proceso que debe realizarse progresivamente. La planificación de servicios es la etapa en la cual se determina cuáles de los servicios identificados conformarán cada uno de los proyectos SOA y en qué orden y momento se ejecutarán dichos proyectos.

Un proyecto SOA de desarrollo de servicios tiene como meta el diseño implementación y despliegue de un grupo de servicios relacionados ya sea porque pertenezcan a un mismo dominio de la empresa, a un mismo proceso o actúen sobre las mismas entidades del negocio. En ese sentido el proceso propuesto se ocupa de efectuar el diseño de los servicios propuestos en el *Plan de Proyecto de Servicios* cuyos atributos se detallan la *Descripción Inicial de Servicios*. Estos artefactos son generados por los arquitectos SOA en la fase de arquitectura y planificación de servicios previas al diseño y constituyen una entrada al proceso propuesto. Los artefactos de entrada al proceso se describen en el Anexo 4.

La siguiente figura muestra cómo se relaciona el proceso de diseño con otros procesos en el desarrollo de proyectos SOA:

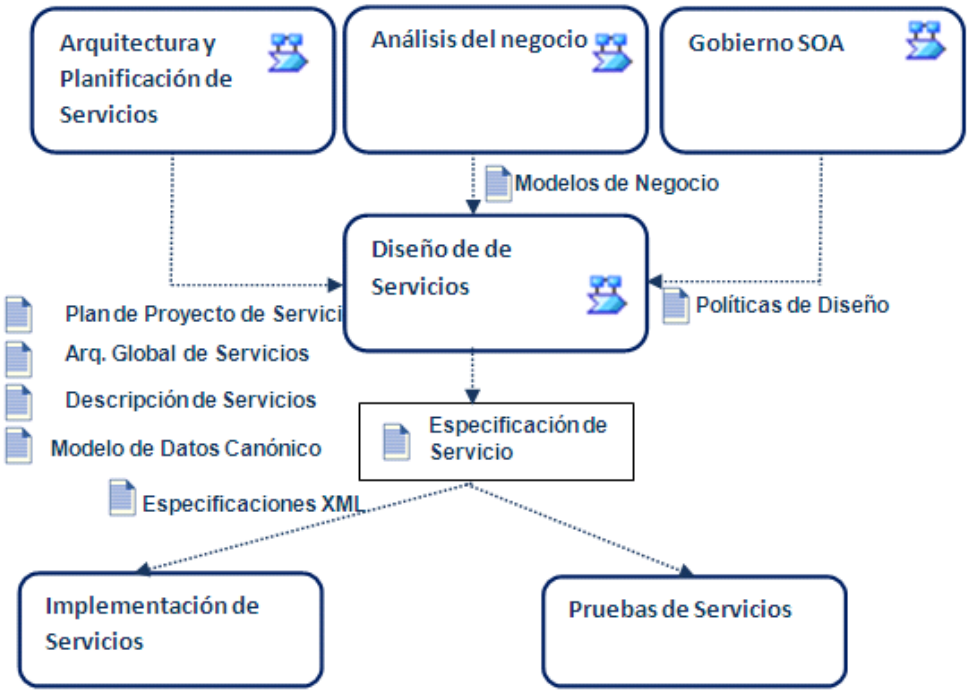

Figura 6. Relación del proceso de diseño de servicios con otros procesos.

Las herramientas propuestas en el proceso se describen en el Anexo 9 y los roles que intervienen en el proceso en el Anexo 10.

El proceso de diseño de servicios consta de ocho tareas (Figura **6**7), en la primera se analizan todos los servicios del proyecto en su conjunto para refinarlos en cuanto a aspectos como la granularidad, reusabilidad y dependencias. El resto de las tareas se realiza iterativamente donde cada iteración se corresponde con el diseño de un servicio.

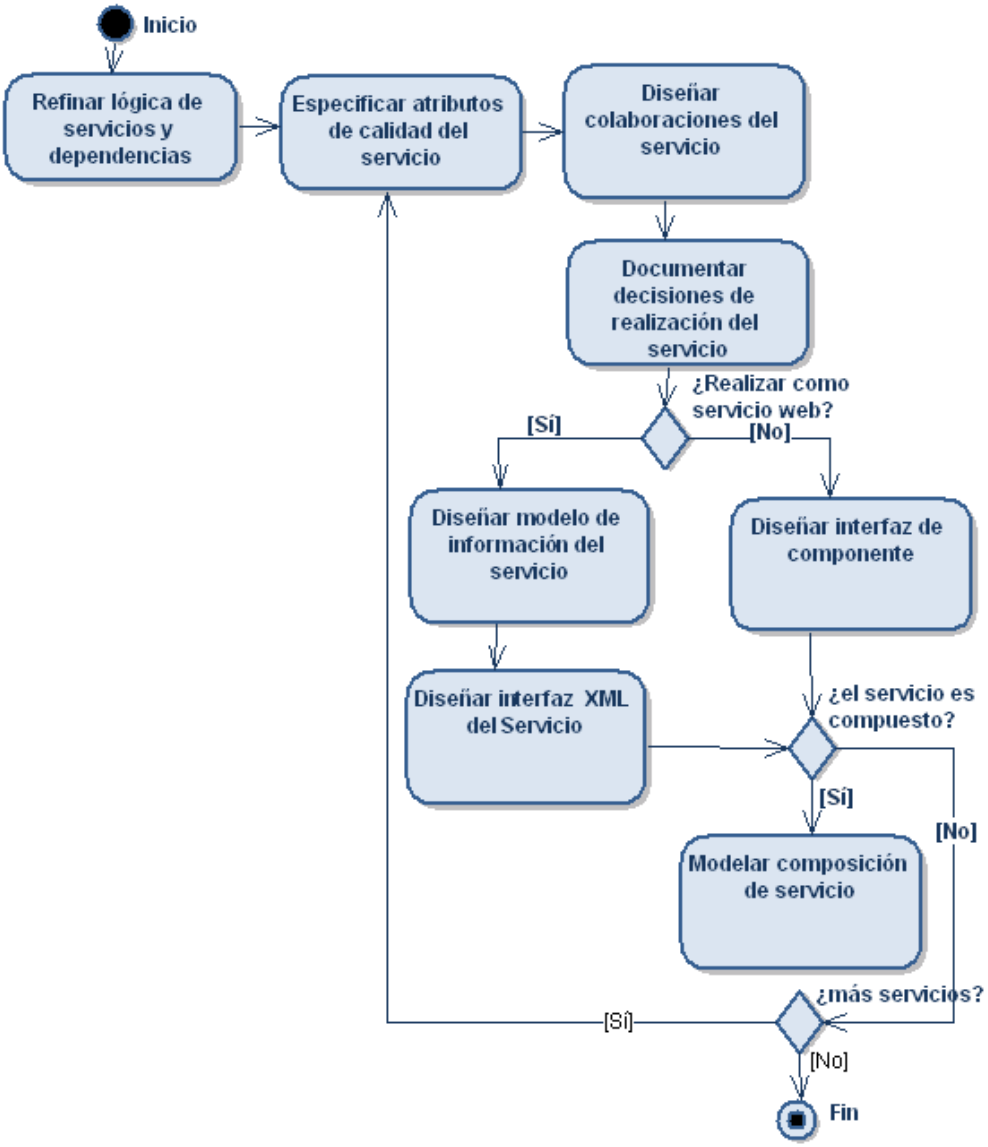

Figura 7. Proceso de Diseño de Servicios propuesto para proyectos SOA.

Es recomendable diseñar primero los servicios que presenten menos dependencias y sean más reutilizables y por último los compuestos o que contengan mayores dependencias de otros servicios, pero el orden en que se diseñan los servicios es dictado en primera instancia por el *Plan de Proyecto de Servicios*. Cada tarea del proceso ofrece recomendaciones para ir describiendo y diseñando el servicio cuyo resultado se refleja en el artefacto *Especificación de Servicio* (Anexo 5)**.**

2.2 Descripción de las tareas del proceso

A continuación se describe cada una de las tareas del proceso propuesto mediante el propósito, pasos que la componen y se muestra gráficamente su relación con las guías, roles y herramientas.

2.2.1 Tarea: Refinar lógica de servicios y dependencias

Propósito: Esta tarea tiene como propósito analizar detalladamente los servicios correspondientes al proyecto SOA sobre el que se trabaja con el fin de garantizar la granularidad, reusabilidad, autonomía, cohesión de estos, así como minimizar el acoplamiento.

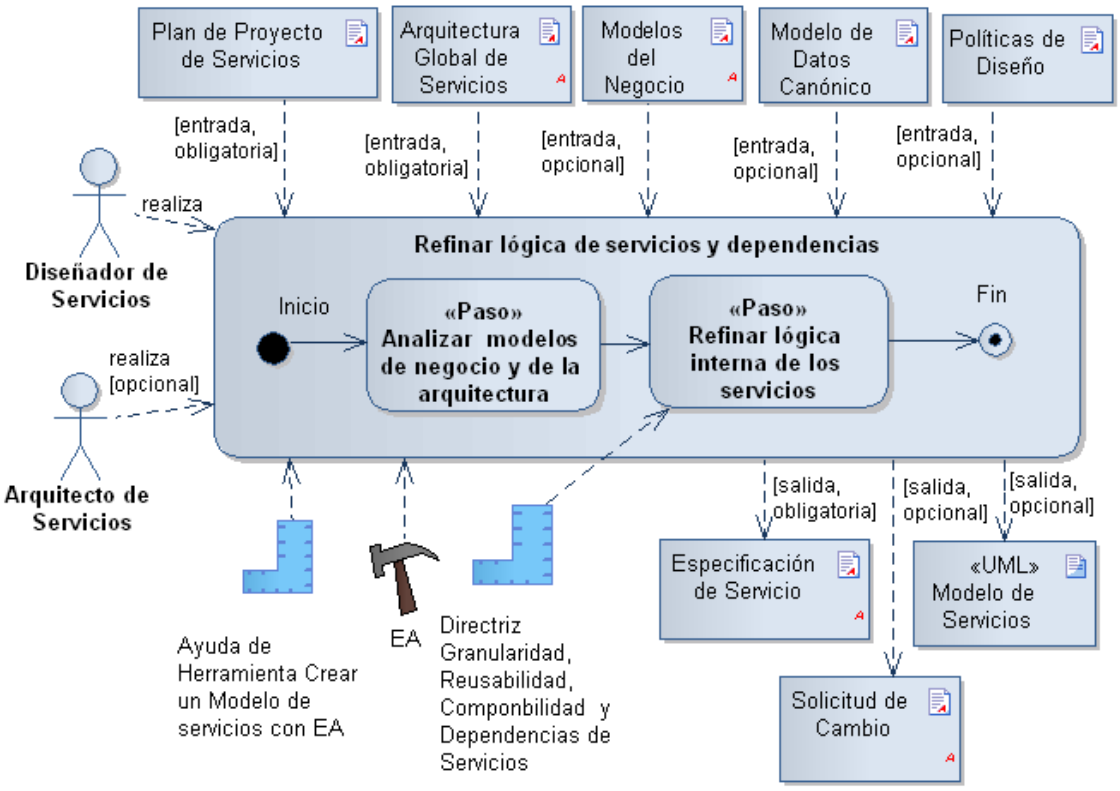

Figura 8. Tarea: Refinar lógica de servicios y dependencias. Su relación con artefactos, roles, guías y herramientas.

Pasos

1. **Analizar Modelos de negocio y de la arquitectura**

   Se analizan los modelos y artefactos provenientes del negocio para entender las necesidades de este, las actividades a automatizar y como

los servicios iniciales propuestos en la arquitectura para el proyecto SOA actual responden a esas necesidades. El *Modelo de Datos Canónico* aporta una visión general de todas las entidades que son relevantes para el negocio y que deberán reutilizare como entradas en los servicios que se diseñen. Por su parte las políticas de diseño, si ya existen en la empresa como resultado de la ejecución de proyectos anteriores o de normativas establecidas, dictarán algunas convenciones como pueden ser:

- Convenciones de nombrado de los servicios, operaciones, entidades, mensajes, etc.
- Clasificación de los servicios
- Patrones a seguir en el diseño
- etc.

La descripción de los artefactos de entrada a esta tarea se puede consultar en el Anexo 4.

Aunque la alineación de los servicios propuestos con las necesidades del negocio debe haberse realizado antes, en las etapas de análisis y diseño de la arquitectura de servicios, en este paso conviene comprobar nuevamente que cada actividad a automatizar definida en los procesos de negocio para el dominio que se diseña, esté satisfecha por alguna operación de los servicios propuestos. De no ser así se deben incorporar nuevas operaciones a alguno de los servicios o identificar nuevos servicios. También se debe evitar duplicidad de funciones, o sea funciones que son realizadas por más de un servicio ya que es una buena práctica tener una única forma consistente de hacer las cosas.

2. Refinar lógica interna de los servicios

En este paso se revisan los servicios y se reajustan en cuanto a la granularidad, reusabilidad, capacidad de composición y dependencias. Para ajustar la granularidad por ejemplo, debe tenerse en cuenta que los servicios de las capas más bajas de la arquitectura por lo general deben tener una granularidad más fina que los de las capas más altas, si

existen políticas de diseño respecto a estos elementos(granularidad, reusabilidad, capacidad de composición y dependencias) también se aplican. Como resultado de estos reajustes pueden surgir nuevos servicios o redistribuir las funcionalidades que realizan estos, en tal caso deben actualizarse la relación de los servicios con los procesos de negocio, con las entidades del *Modelo de Datos Canónico*, y con otros servicios. Estas relaciones por lo general se expresan en los modelos de negocio y de la arquitectura mediante las matrices que permiten mantener la trazabilidad de los servicios con otros elementos, estas matrices pueden ser de servicios contra procesos de negocio, servicios contra metas del negocio, servicios contra entidades, etc.

Puede que al cambiar la funcionalidad o alcance de un servicio también se modifiquen las dependencias hacia él lo cual incide en la arquitectura de servicios. En *la Directriz Granularidad, Reusabilidad, Componibilidad y Dependencias Servicios* se ofrece un conjunto de buenas prácticas y patrones para realizar estos ajustes. Los cambios que se requiera realizar en diferentes modelos (de negocio y de arquitectura) pueden gestionarse directamente entre el diseñador y los analistas y arquitectos o, para un proceso más formal, solicitarlos mediante el artefacto *Solicitud de Cambio* (Anexo 7).

Para reflejar los resultados de esta tarea, se crea para cada servicio una *Especificación de Servicio* (Anexo 5), documentando las propiedades técnicas del servicio y las propiedades del servicio de cara al negocio. El diseñador puede crear un modelo de servicios UML (ver *Ayuda de Herramienta Crear un Modelo de Servicios con EA*) que servirá de base para las posteriores tareas de modelado.

2.2.2 Tarea: Especificar atributos de calidad del servicio

Propósito: Especificar los atributos de calidad del servicio deseada por los clientes. Los atributos de calidad (seguridad, confiabilidad, rendimiento, entre otros) reflejan las necesidades de los clientes del servicio y servirá de base

para una negociación entre los clientes y proveedores cuyo acuerdo final debe quedar reflejado en el artefacto *Acuerdo a Nivel de Servicio*[14].

Por su complejidad, el proceso de negociación de de los atributos de calidad que tendrá realmente el servicio se considera externo al proceso de diseño y, cuando este se produzca, se debe actualizar la *Especificación del Servicio* colocando una referencia al documento *Acuerdo a Nivel de Servicio*. Los atributos de calidad especificados aquí, y luego negociados en el *Acuerdo a Nivel de Servicio*, influirán en decisiones arquitectónicas sobre la infraestructura de software y hardware a utilizar para soportar la SOA, y en la asignación de recursos de hardware a los componentes que realizan el servicio para lograr que se satisfagan dichos atributos de calidad. En el caso de los servicios web, los atributos especificados influirán en la decisión de utilizar o no determinado estándar para la implementación de un servicio, por ejemplo requisitos complejos de seguridad pueden conllevar a la decisión por parte de los arquitectos de seguridad de usar el WS-Security (Anexo 1) unido a lenguaje de aserciones como SSL[15].

Pasos

1. **Documentar atributos de calidad**

   En este paso se documentan los atributos de calidad del servicio (rendimiento, seguridad, disponibilidad, entre otros) en la *Especificación de Servicio*.

   La *Arquitectura Global de Servicios* debe tener definidas las expectativas de calidad para la arquitectura de servicios en general. De esta especificación, los modelo de negocio y conversaciones con clientes del servicio se pueden derivar los atributos de calidad del servicio. Estos

[14] Constituye un compromiso sobre los atributos de calidad que debe proveer el servicio, una vez negociados estos entre clientes y proveedores. El cliente tiene necesidades específicas sobre rendimiento, seguridad, disponibilidad, etc. del servicio pero el proveedor tiene posibilidades limitadas dependiendo de la infraestructura de la que dispone, tecnología, personal, conocimientos, etc.

[15] Del Inglés, Secure Socket Layer (SSL), es un protocolo criptográfico que provee seguridad para la comunicación a través de redes como Internet.

atributos se especifican y en esta sección se especifica cuáles de estas expectativas se aplican a este servicio, y si existe alguna desviación de las expectativas de calidad generales definidas en la *Arquitectura Global de Servicios.*

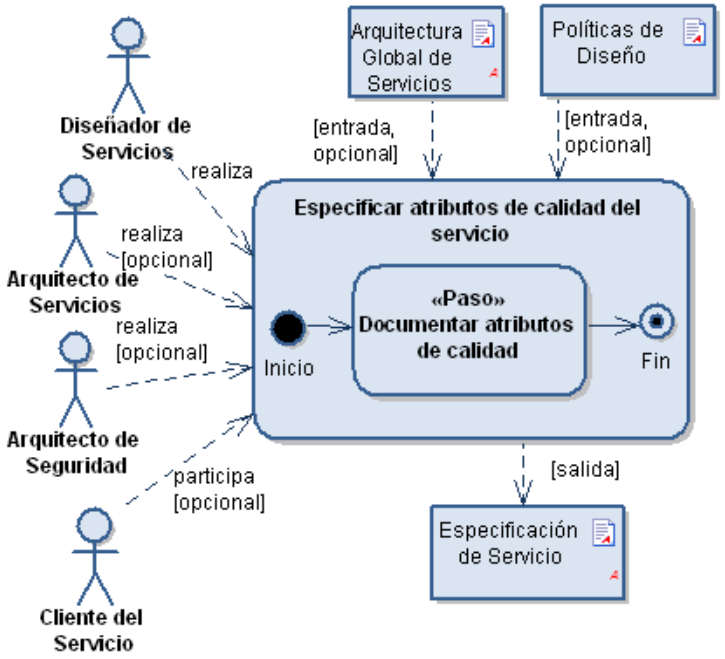

Figura 9. Tarea: Especificar atributos de calidad. Su relación con artefactos y roles.

Cada operación de servicio puede tener requisitos de calidad diferentes. Estos requisitos se pueden especificar para un grupo de operaciones que tengan en común los mismos atributos de calidad.

Como meta, cada atributo debe describirse con el mayor nivel de detalle posible garantizando que esté en un lenguaje entendible y que su cumplimiento sea medible y verificable. Es posible que no se puedan determinar desde un principio requisitos explícitos y verificables, en cuyo caso se describen solamente con un valor cualitativo (por ejemplo ”Bronce”, “Plata”, “Oro”) indicativo del nivel de la importancia que prestársele y se posterga la especificación más detallada y medible para la etapa de pruebas donde, en su interacción en tiempo de ejecución, se pueda determinar parámetros óptimos para su funcionamiento. Si el servicio es crítico para el negocio y resulta de suma importancia conocer de antemano sus parámetros de funcionamiento y no se puede obtener esta información por otra vía será

necesario crear un ambiente de simulación para el servicio y ejecutar dichas pruebas incluso antes de ser construido.

2.2.3 Tarea: Diseñar colaboraciones del servicio

Propósito: Detallar la especificación funcional del servicio y la forma de interactuar con él.

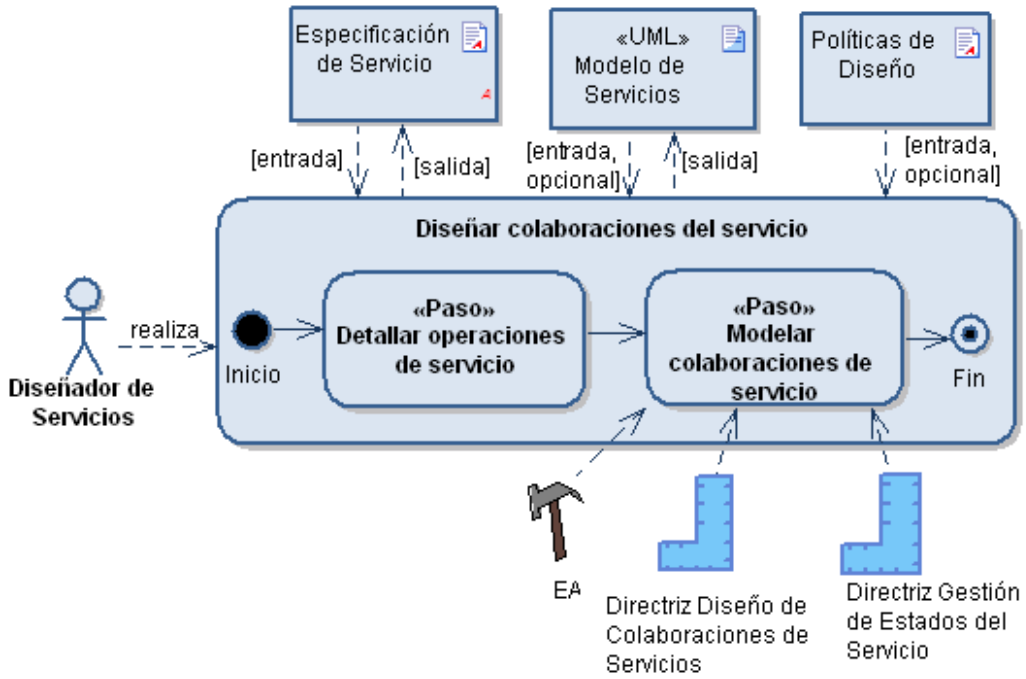

Figura 10. Tarea: Diseñar colaboraciones del servicio. Su relación con artefactos, roles, guías y herramientas.

Pasos

1. **Detallar operaciones de servicio**

   En este paso se documentan en detalle las operaciones de servicio, su propósito, precondiciones, postcondiciones y restricciones de validación que requieren los datos o mensajes de entrada al servicio y los tipos de fallos o errores que se pueden producir. Eso debe quedar reflejado en la sección *Especificación de Operaciones* del artefacto *Especificación de Servicio*. También se describe, para cada operación, si es de tipo transaccional o de actualización en cuyo caso se deberán tener en cuenta aspectos de manejo de transacciones en la próxima tarea *Documentar Decisiones de Realización del Servicio*.

2. **Modelar colaboraciones del servicio**

   En este paso se deben tener en cuenta los patrones de intercambio de mensajes:

   - Solicitud/Respuesta síncrona
   - Mensaje de una dirección
   - Notificación
   - Solicitud/respuesta asíncrona
   - Publicar/suscribir

   Se debe especificar para cada operación el que más se ajusta a ella y documentarlo en la sección *Especificación de Operaciones* del artefacto *Especificación de Servicio.* En la *Directriz Diseño de Colaboraciones del Servicio* se detallan estos patrones.

   Para aquellas operaciones que requieren ser invocadas en un orden específico para mantener la consistencia de los datos (por ejemplo solo se puede invocar a la operación *EliminarProducto* luego de haber invocado a *CrearProducto)* se modela la secuencia de operaciones mediante un diagrama de secuencia o colaboración en la herramienta CASE[16] propuesta (ver *Directriz Diseño de Colaboraciones y Directriz Gestión de Estados de Servicio).* Las colaboraciones se documentan en el artefacto *Especificación de Servicios*, en la sección *Secuencias de Operaciones Obligatorias*.

2.2.4 Tarea: Documentar decisiones de realización del servicio

Propósito: Esta tarea se centra en la especificación detallada de un grupo de decisiones que deberán tenerse en cuenta a la hora de implementar el servicio ya que dichas decisiones están asociadas al cumplimiento de los atributos de calidad del servicio. Las principales decisiones a especificar son:

- Enfoque de provisión del servicio (comprar, subcontratar, suscribir, construir, transformar o integrar)
- Si se expone como servicio web.

[16] Ingeniería de software asistida por computadora, del Inglés Computer-Aided Software Engineering.

- Si será público y/o publicado.
- Protocolos para comunicarse con el servicio (ej.: SOAP, HTTP, JMS, etc.).
- Estándares o especificaciones a usar. (ej. WSDL, WS-Security, WS-Reliability, etc.).
- Cómo se realizará la gestión de estados del servicio

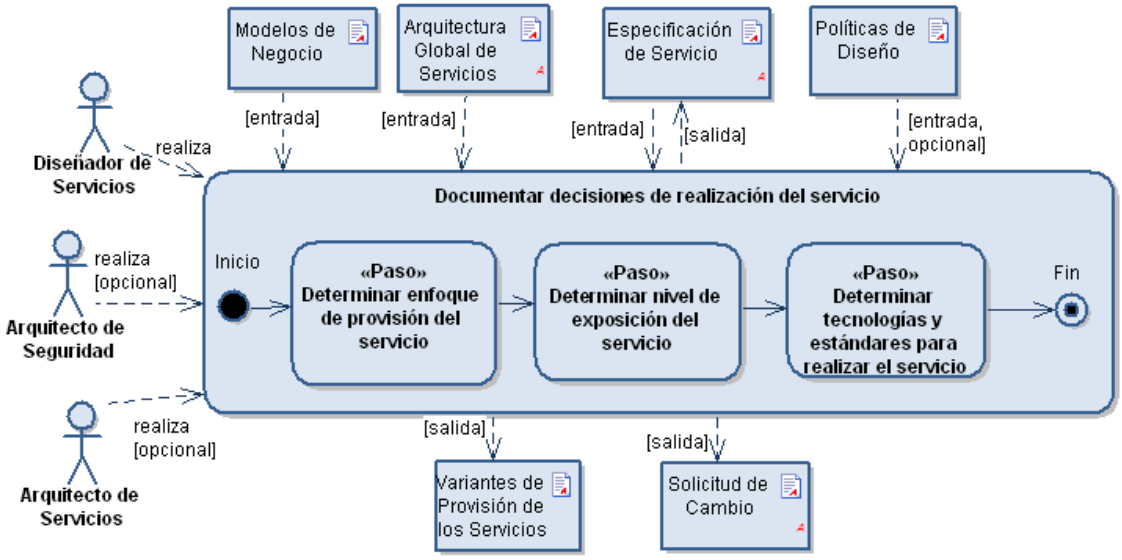

Figura 11. Tarea: Documentar decisiones de realización del servicio. Su relación con roles y artefactos.

Pasos

1- Determinar enfoque de provisión del servicio

En este paso se revisa la *Arquitectura Global de Servicios* y las *Políticas de Diseño* para determinar si se ha definido alguna política que regule el enfoque de provisión de los servicios. Si no se ha establecido nada al respecto, se analizan los *Modelos de Negocio*, específicamente aquellos artefactos que describen los activos existentes en la organización en cuanto a funciones automatizadas y sistemas legados.

Si alguna de las funcionalidades que brinda el servicio es soportado actualmente por algún sistema legado en la empresa, se debe estudiar con mayor profundidad los mecanismos de integración que proporciona dicho sistema. Algunos de los mecanismos de integración que pueden proporcionar los sistemas legados son:

- Acceso a sus bases de datos.

- Provee una interfaz de aplicación (API, del inglés, Application Interface).

- Consume y exporta datos procesados a través de ficheros.

Se debe valorar si el sistema legado ofrece potencialidades para cumplir con los atributos de calidad especificados para el servicio y si la infraestructura definida por la arquitectura, por ejemplo el ESB[17], soporta la integración con este sistema legado o si se puede realizar alguna especie de adaptador, conector o servicio envolvente[18].

Para aquellos servicios y funcionalidades que no tienen soporte automatizado en la empresa o donde la reutilización de estos sistemas sea muy costosa, se puede hace una búsqueda en el mercado para determinar si hay alguna empresa que lo provea, se dedique a desarrollarlo, subcontratarlo o brinde algún mecanismo de subscripción. El análisis de costo, riesgos, tiempo para obtener el servicio y fortalezas para cada variante se hace imprescindible.

Para un proceso riguroso a la hora de determinar el enfoque de provisión, los resultados del análisis anterior deben documentarse en el artefacto *Variantes de Provisión de los Servicios* y ser sometidos a un proceso de toma de decisiones en el cual deben participar representantes del negocio, arquitectos, diseñadores e implementadores. La decisión final debe reflejarse en el artefacto *Especificación de Servicios*, en la sección *Propiedades Técnicas del Servicio*, en la casilla *Fuentes /Organización proveedora*.

[17] Bus de Servicios Empresarial, del Inglés Enterprise Service Bus (ESB). Es un tipo de capa mediadora o middleware que tiene entre sus funciones la orquestación de servicios, enrutamiento y provee adaptadores para conectar un servicio a sistemas legados realizados en determinadas plataformas y lenguajes. La cantidad de adaptadores varía de un ESB a otro.

[18] Este término proviene del Inglés “wrapper service”. Se refiere a un servicio que actúa como mediador para posibilitar la comunicación con un sistema legado.

2- Determinar nivel de exposición del servicio

En este paso se determina cual debe ser la visibilidad del servicio teniendo en cuenta las políticas de seguridad definidas en la *Arquitectura Global de Servicios,* en caso que haya alguna que restrinja este aspecto.

Se debe determinar si el servicio será público o publicado:

- Servicio público: significa que la invocación del servicio exige previo conocimiento del consumidor.
- Servicio publicado: significa que el servicio estará disponible para ser descubierto (por ejemplo a través de una UDDI).

Luego determinar el alcance de la visibilidad:

- Solamente estará accesible dentro del dominio al que pertenece.
- Accesible a toda la empresa.
- Accesible dentro y fuera de la empresa.
- Accesible solo fuera de la empresa.

Estas decisiones deben quedar documentadas en el artefacto *Especificación de Servicios*, en la sección *Instrucciones para Despliegue*.

3- Determinar tecnologías y estándares para realizar el servicio

En este paso se documentan cuáles de las tecnologías, estándares y protocolos, de aquellas propuestas de manera general en la *Arquitectura Global de Servicios,* serán aplicadas a este servicio en particular. Algunos ejemplos de decisiones pueden ser:

- Realizar el servicio como servicio web, determinando los estándares asociados (ver Anexo 1).
- Realizar servicio con otra tecnología (por ejemplo como componentes de Java con una interfaz bien definida).

Debe tomarse en consideración la infraestructura propuesta por la arquitectura ya que esta debe soportar los estándares propuestos.

En cualquier caso se procede a documentar dichos estándares y protocolos en la sección *Cumplimiento de las Normas* de la *Especificación de Servicio;* y la decisión de realizarlo como servicio web u otra tecnología en la sección *Instrucciones para Implementación*.

2.2.5 Tarea: Diseñar el modelo de información del servicio

Propósito: Crear un modelo de datos que permita la comunicación de los consumidores con el servicio y que dicho modelo sea compatible con el modelo de datos canónicos de la empresa a fin de lograr la interoperabilidad y flexibilidad de los servicios a nivel de datos. El modelo de información del servicio define todas las entidades y mensajes asociados a las operaciones del servicio ya sean de entrada o salida.

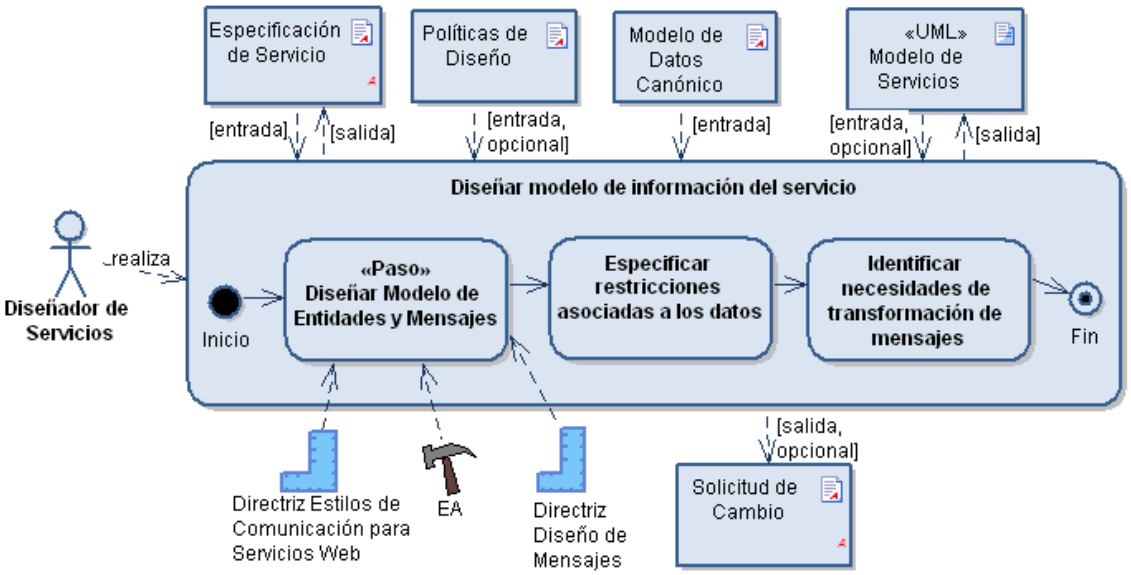

Figura 12. Tarea: Diseñar el modelo de información del servicio. Su relación con roles, artefactos, guías y herramientas.

Pasos

1. **Diseñar modelo de entidades y mensajes**

   En este paso se determina para cada operación de servicio cual será la información de entrada y salida, así como el estilo de interacción.

   Los estilos de interacción pueden ser de tipo:

-Parámetros: también conocido como RPC[19], este estilo consiste en que el servicio tiene tantos parámetros de entrada y salida como necesite.

-Documento: con este estilo todos los parámetros de entrada que necesita la operación se agrupan en un solo documento (en formato XML) y este se le pasa como entrada. Lo mismo sucede con los parámetros de salida. El estilo *Documento* tiende a ser más desacoplado y flexible que el de *Parámetros*.

Para obtener información de ayuda sobre cómo seleccionar el estilo de comunicación ver *Directriz Estilos de Comunicación para Servicios Web.*

El estilo seleccionado por cada operación se documenta en la sección *Especificación de Operaciones* de la *Especificación de Servicio.*

Para diseñar el modelo de entidades y mensajes del servicio, las entidades que contienen la información de entrada o salida se conformarán a partir de las entidades provistas en el artefacto *Modelo de Datos Canónico*. Una semántica de mensajes uniforme es uno de los requerimientos más importantes para la interoperabilidad de los servicios. Esto asegura que los consumidores y proveedores de servicio intercambian datos en una forma consistente.

Para una descripción más detallada durante el diseño de tipos y de mensajes consultar *Directriz Diseño de Mensajes.* Este modelo de tipos y mensajes puede realizarse como un diagrama de clases o entidades con la herramienta propuesta lo cual enriquecerá el *Modelo UML de Servicios* creando un modelo de datos para el servicio que se diseña.

[19] Estilo de comunicación en forma de parámetros de entrada para la operación. Este termino proviene del Inglés Remote Procedure Call (RPC) que significa llamada a procedimiento remoto y se ha utilizado en arquitecturas distribuidas precedentes (por ejemplo CORBA)como forma de invocar una funcionalidad u operación de una clase o interfaz a través de la red.

2. Especificar restricciones asociadas a los datos

Durante este paso se especifican las reglas, políticas o restricciones asociadas a las entidades y atributos que no se pueden expresar en los diagramas del modelo de información. Estas restricciones podrían haber sido especificadas como reglas por los analistas del negocio o en el *Modelo de Datos Canónico*, en este caso se pone una columna más a la tabla para hacer la referencia a dichas reglas. Es recomendable que en la *Especificación de Servicios* quede explícito en qué consiste cada una de estas restricciones independientemente de las referencias hechas. El lenguaje utilizado puede ser el común o se puede usar un lenguaje formal, por ejemplo:

- Id, nombre ≠ Ø
- 0 < edad < 18
- El número de carnet de identidad debe tener exactamente once dígitos

Tanto el diagrama de entidades como las restricciones asociadas a los datos deben quedar reflejados en la sección *Modelo de Información* de la *Especificación de Servicio.*

3. Identificar necesidades de transformación de mensajes

Si el servicio que se diseña tiene dependencias de otros servicios, o sea que requiere consumir datos de otro servicio y el formato en que ese servicio le ofrece los datos no satisface completamente sus necesidades se necesita entonces emitir una solicitud de transformación de datos mediante el artefacto *Solicitud de Cambio* al proceso de arquitectura quien decidirá si dicha transformación se materializa con un nuevo servicio de transformación de datos el cual se debe diseñar e implementar, reutilizando servicios de la infraestructura como los que ofrecen algunos ESB, a través de un motor de reglas o políticas, etc.

2.2.6 Tarea: Diseñar interfaz XML del servicio

Propósito: Crear una interfaz de servicio desacoplada de su implementación mediante el uso de estándares asociados a servicios web que permitan el acceso a las funcionalidades desde cualquier plataforma.

Pasos

1. **Transformar a estándares XML**

   En este paso se define la interfaz del servicio en la herramienta propuesta de acuerdo con las operaciones diseñadas para el servicio. La nomenclatura utilizada puede estar dictada por las políticas de diseño.

   A partir del artefacto *Modelo del Servicio (UML),* en una secuencia de pasos que se indican en la *Ayuda de Herramienta Generar WSDL y XSD con EA* se generan los respectivos ficheros XML de acuerdo al formato de los estándares especificados.

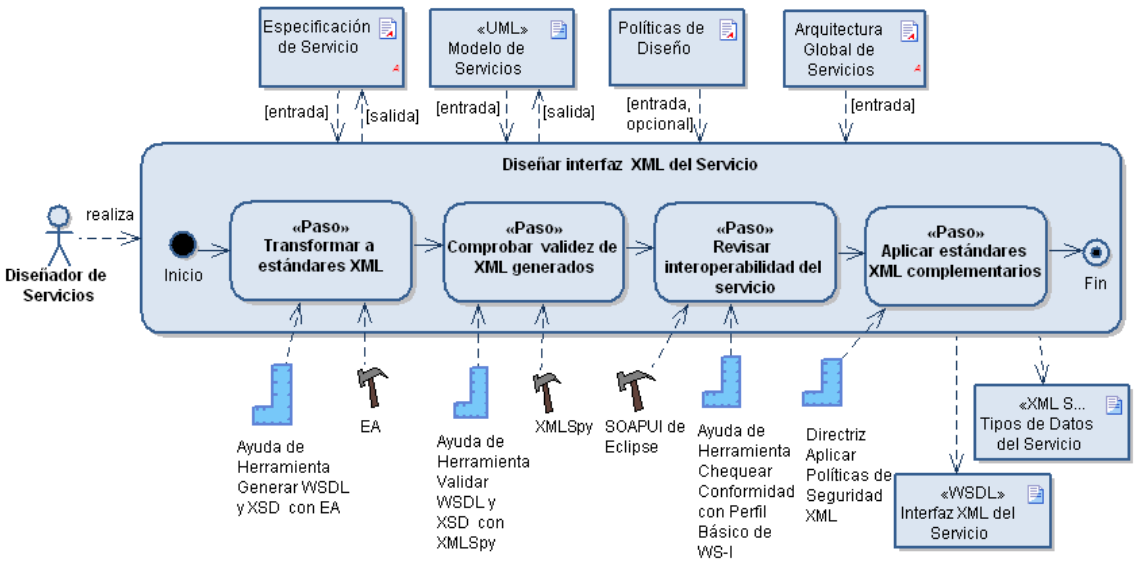

Figura 13. Tarea: Diseñar interfaz XML del servicio. Su relación con roles, artefactos, guías y herramientas.

2. **Comprobar validez de XML generados**

   Los ficheros XML generados automáticamente pueden presentar inconsistencias con el formato XML al que se supone que respondan, para eliminar estos problemas se verifica con la herramienta propuesta

XMLSpy que los XML sean válidos y bien formados (ver *Ayuda de Herramienta Validar WSDL y XSD con XMLSpy*).

3. **Revisar interoperabilidad del servicio**

   Aun cuando los ficheros XML (WSDL y XSD) de interfaz del servicio sean válidos y bien formados esto no asegura su interoperabilidad, o sea que puedan ser accedidos por aplicaciones implementadas en cualquier plataforma y lenguaje. Para minimizar los problemas de interoperabilidad resulta conveniente chequear la conformidad de estos ficheros con el Perfil Básico de WS-I, lo cual se puede realizar fácilmente con el complemento SOAPUI de la herramienta Eclipse (Anexo 9). Para realizar este chequeo de interoperabilidad, seguir los pasos definidos en *Ayuda de Herramienta Chequear Conformidad con Perfil Básico de WS-I.*

4. **Aplicar estándares XML complementarios**

   Cuando se trabaja con servicios web muchos de los atributos de calidad definidos para el servicio como seguridad, confiabilidad del mensaje entre otros requisitos especiales son tratados mediante el uso de estándares XML (ver Anexo 1) que deben ser aplicados a la interfaz del servicio. En este paso se complementa la interfaz del servicio, definida mediante el estándar WSDL, con otros estándares XML como puede ser WS-Security, WS-Reliability, etc., definidos en la sección *Cumplimiento de Normas y Estándares* (específicamente en los *Estándares Técnicos*) de la *Especificación de Servicio.* En la *Directriz Aplicar Políticas de Seguridad XML*, se ofrece un ejemplo de cómo aplicar este tipo de estándares al servicio.

Al final de esta tarea debe quedar documentado en la *Especificación de Servicio* (sección *Firma de Operaciones*) la dirección para acceder al WSDL, así como el nombre de cada una de las operaciones tal como aparecen el WSDL, los tipos de datos que utiliza, etc.

2.2.7 Tarea: Diseñar interfaz de componente

Propósito: Crear una interfaz de servicio desacoplada de su implementación. Esta tarea se aplica a aquellos servicios que no se realizarán como servicios web sino con otra tecnología que permita hacer uso de interfaces desacopladas de la implementación del servicio.

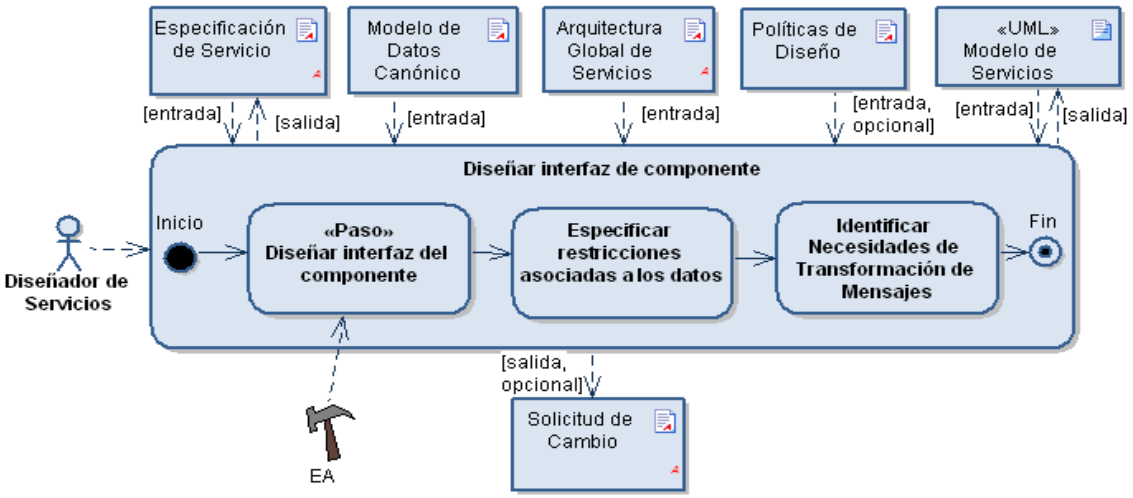

Figura 14. Tarea: Diseñar interfaz de componente. Su relación con roles, artefactos, guías y herramientas.

Pasos

1. **Diseñar interfaz del componente**

   En este paso se modela la interfaz del componente que actuará como servicio con la herramienta CASE actualizando el *Modelo de Servicio (UML)*. Las operaciones son las mismas que se definieron en la sección *Especificación de Operaciones* de la *Especificación de Servicio,* a las cuales debe aplicarse las convenciones de nombres que se definan en las *Políticas de Diseño*. Luego se le asignan los parámetros a cada operación tratando de ajustarse en lo posible a la estructura definida para las entidades en modelo canónico, aunque un parámetro también podrá ser un tipo simple por ejemplo un entero.

2. **Diseñar restricciones asociadas a los datos**

   Para este paso seguir las mismas instrucciones del paso 2: *Diseñar restricciones asociadas a los datos* de la tarea *Diseñar modelo de información del servicio* (epígrafe 2.2.5).

   Tanto las operaciones definitivas diseñadas en la interfaz como los parámetros de entrada y salida deben documentarse en la sección *Firma de Operaciones* de la *Especificación de Servicio.*

3. **Identificar necesidades de transformación de mensajes**

   Seguir las mismas instrucciones del paso 3: *Identificar necesidades de transformación de mensajes* de la tarea *Diseñar modelo de información del servicio* (epígrafe 2.2.5).

2.2.8 Tarea: Modelar composición de servicio

Propósito: Detallar la lógica de negocio que contiene el servicio y la forma en que este utiliza a otros servicios para llevar a cabo su funcionalidad.

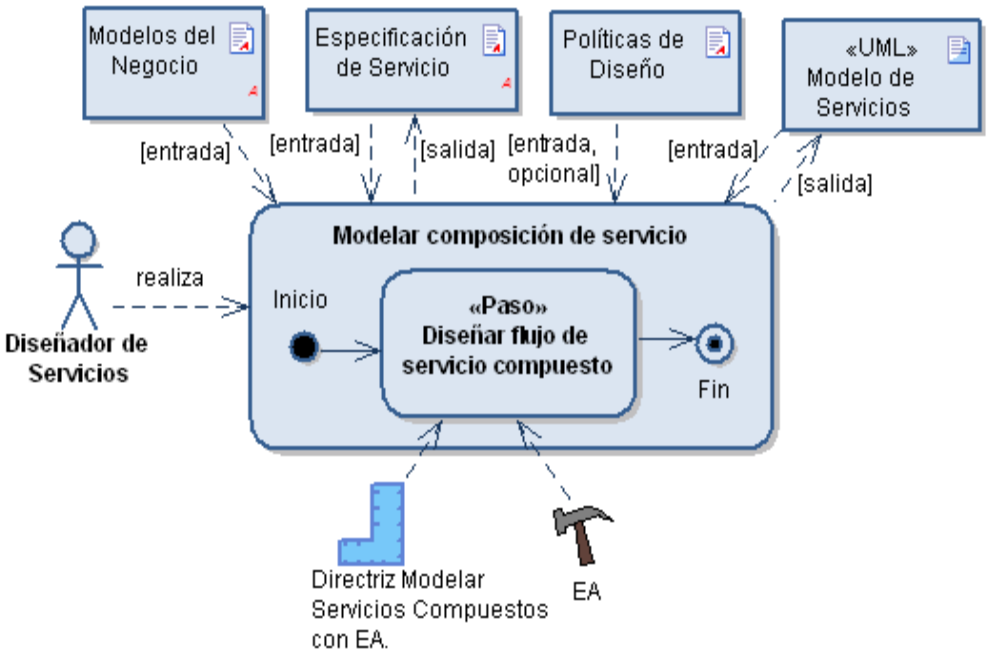

Figura 15. Tarea: Modelar composición de servicio. Su relación con roles, artefactos, guías y herramientas.

Pasos

1. Diseñar flujo de servicio compuesto

Para cada operación del servicio que requiera realizar una secuencia de acciones desacopladas, en este paso se modela, mediante un diagrama de interacción dicha secuencia de acciones. Por lo general estos son servicios que realizan algún proceso o subproceso de negocio e invocan a otros servicios. La secuencia de acciones modeladas puede ser útil a los implementadores para realizar el servicio a través de un motor de flujos de trabajo o herramienta BPM. Cada acción debe ser bien concreta y preferiblemente lo más atómica posible. Una acción concreta puede tener precondiciones y postcondiciones. La *Directriz Modelar Servicios Compuestos con EA* ofrece una guía para realizar este paso con más detalles.

Los resultados de este paso deben reflejarse en la sección *Orquestación de Operaciones* de la *Especificación de Servicio.*

2.3 Consideraciones del capítulo

- Como resultado del análisis realizado en el capítulo anterior a las diferentes metodologías SOA y, atendiendo a los elementos más útiles encontrados en ellas, se tomaron como referencia para la elaboración del proceso de diseño de servicios propuesto RUP-SOMA de IBM, Proceso de Diseño de Thomas Earl y el Proceso CBDI-SAE**.**
- Se definió la relación del proceso de diseño propuesto con otros procesos necesarios para la realización de proyectos SOA.
- Se definieron tareas, roles, herramientas y guías que, de manera detallada, ayudan a la realización del diseño de servicios teniendo en cuenta los principios de orientación a servicios.

Capítulo 3: Validación y Aplicación Práctica de la Propuesta

En este capítulo se propone evaluar la solución propuesta mediante su aplicación práctica en un proyecto piloto. El objetivo de esta aplicación es analizar si el proceso propuesto contribuyó a lograr una la calidad adecuada en diseño de los servicios mediante la adherencia a los principios de orientación a servicio y la calidad de las especificaciones del diseño.

3.1 Aplicación práctica de la propuesta mediante un proyecto piloto

Los proyectos piloto son útiles para probar soluciones con enfoque centrado tanto en el proceso como en el problema y contribuyen al perfeccionamiento de la solución para adaptarlas a la realidad lidiando con los factores inesperados que pueden incidir los resultados finales. Se puede decir entonces, que el objetivo de los pilotos es comprobar la solución en un proyecto real de la organización y capturar lecciones aprendidas y resultados para refinar la solución con vistas a su posterior instalación (McFeeley, 1996).

La secuencia de actividades para la realización de un proyecto piloto consta de ocho actividades que se muestran en la siguiente figura.

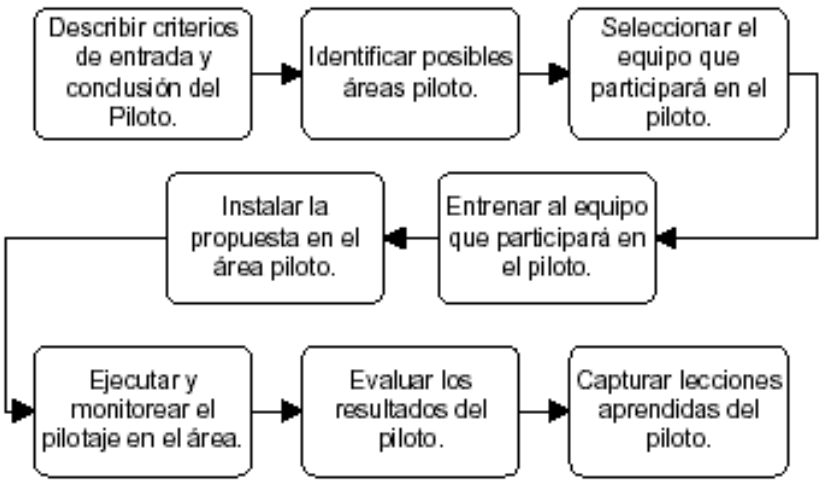

Figura 16. Actividades para la ejecución de un proyecto piloto.

3.2 Criterios de entrada y conclusión del piloto

Como criterios para la selección del proyecto piloto se tienen las siguientes:

- Debe tratarse de un proyecto real que requiera diseño de servicios según los principios de SOA.
- El proyecto debe haber transitado por las etapas de análisis del negocio, modelado de los procesos y realizado una arquitectura inicial de servicios.
- Debe existir interés y compromiso por parte de los integrantes del proyecto en la realización del proceso de diseño.

Como criterios de conclusión del piloto se deben haber obtenido los artefactos de salida que propone el proceso propuesto para los servicios diseñados:

- Especificaciones de servicios.
- Modelos UML de los servicios en formato que proporcione la herramienta CASE utilizada.
- Otros artefactos en caso que sea aplicables al proyecto como son:

  - Archivos XML y WSDL.

  - Solicitudes de cambio.

  - Variantes de provisión de los servicios.

3.3 Identificación de posibles áreas piloto

En un análisis realizado a los proyectos actuales en la Universidad de las Ciencias Informáticas se identificó como proyecto candidato que cumple con estas condiciones el proyecto de Gestión de Operaciones del CIM.

El Centro de Inmunología Molecular (CIM) tiene como misión obtener y producir nuevos biofármacos destinados al tratamiento del cáncer y otras enfermedades crónicas no trasmisibles e introducirlos en la salud pública cubana, hacer la actividad científica y productiva económicamente sostenible y hacer aportes a la economía del país. El CIM y el Centro de Desarrollo de Arquitecturas Empresariales (CDAE) de la UCI, se encuentran trabajando actualmente en un

proyecto que pretende contribuir a lograr una mayor integración y eficiencia en los procesos biotecnológicos. Para ello se ha concebido la adopción paulatina de un modelo de automatización distribuido bajo los principios de SOA.

Luego de un estudio de los diferentes modelos de negocio y objetivos estratégicos del CIM y, debido a la necesidad de probar la eficacia de un enfoque SOA, se determinó desarrollar un proyecto piloto que consiste en el desarrollo e integración de un grupo de servicios que intervienen en el proceso de Gestión de Ordenes Comerciales para la fabricación de productos.

Este proyecto comprende varias etapas y procesos comenzando por la modelación del proceso de negocio a automatizar, realización de la arquitectura de servicios, diseño de servicios, implementación, prueba y ensamblaje de los servicios en una solución de cara al cliente.

El proceso de diseño propuesto en esta investigación se aplicó en la etapa de diseño de servicios del proyecto correspondiente al proyecto piloto que se desarrolla actualmente para el CIM.

3.4 Selección del equipo a participar en el piloto

Para la ejecución del proceso propuesto y en correspondencia con los roles definidos para el proceso de diseño se seleccionó:

- Dos diseñadores de servicios.
- Un arquitecto de servicios.
- Un arquitecto de seguridad.

3.5 Entrenamiento del equipo que participa en el piloto

Al equipo que participó en el piloto se le proporcionó toda la documentación disponible y mediante varios talleres se le explicó en qué consistía el proceso. El aprendizaje se facilitó debido a que ya tenían conocimiento previo sobre SOA, los principios de orientación a servicios, XML, Servicios Web, seguridad, entre otros y su desempeño laboral estaba muy cercano a esos roles. El interés y compromiso de los participantes se logró debido a la presión por la fechas de

entrega del proyecto, aunque en ocasiones hubo retrasos debido a que se les asignó otras tareas no planificadas.

3.6 Instalación, ejecución y monitoreo de la propuesta en el proyecto

Como resultado de las primeras etapas del proyecto se obtuvieron los artefactos que sirven de entrada al proceso de diseño de servicios:

- Plan de Proyecto de Servicios.
- Arquitectura Global de Servicios.
- Modelos del Negocio:
    - Modelo de Dominios del Negocio.
    - Inventario de Procesos de la Organización.
    - Mapa de Proceso del Proyecto Piloto.
    - Matriz de Procesos vs Servicios.
- Modelo de Datos Canónico.
- Políticas de Diseño: las políticas especificadas en este caso definen la convención de nombres para servicios, operaciones, entidades de información y mensajes a utilizar en el proyecto.

El proceso a automatizar mediante servicios se describe en el Anexo 11.

La arquitectura inicial de servicios para el proyecto piloto proveniente de los procesos previos al diseño se puede observar en el Anexo 12. Esta arquitectura de inicial fue sometida a un proceso de refinamiento mientras se ejecutaba el proceso de diseño de servicios como se explica a continuación.

Durante la primera actividad *Refinar Lógica de Servicios y Dependencias*, en el primer paso *Analizar Modelos de Negocio y de la Arquitectura*, se estudiaron los modelos y artefactos de entrada para entender la relación entre los servicios propuestos y las necesidades del negocio. Fue necesario solicitar la participación de los miembros del proyecto que desarrollaron los modelos de proceso y modelo de datos canónico para entender algunos términos usados que eran muy parecidos y realizar algunos ajustes y eliminación de redundancias en las entidades que se proponían.

En lo pasos dos y tres, luego del análisis de la lógica que debía encapsular cada servicio y para lograr una mayor reutilización y cohesión se decidió unificar los servicios *ConformadoSolicitudProd* y *ChequeoSolicitudProd* en un solo servicio *GestionSolicitudProd* ya que estos dos servicios se relacionaban con las mismas entidades del negocio. Al servicio *GestionSolicitudProd* se le asignó la lógica funcional de los dos anteriores.

Para disminuir la granularidad del servicio *Materiales* ya que estaba sobrecargado al tener que gestionar varias entidades del negocio se decidió agregar un nuevo servicio: *SolicitudesMat***,** que se encargará solamente de gestionar las solicitudes y los datos propios de los materiales.

Al analizar las dependencias se identificó que existía una fuerte relación entre varios servicios de la capa intermedia y el servicio de utilidades *GestionDocs*, este es un servicio muy reutilizable que provee acceso a información y documentos proveniente de diversas fuentes como repositorios, gestores documentales, bases de datos, etc. La arquitectura de servicios para el proyecto piloto finalmente quedó conformada como se muestra en la siguiente figura.

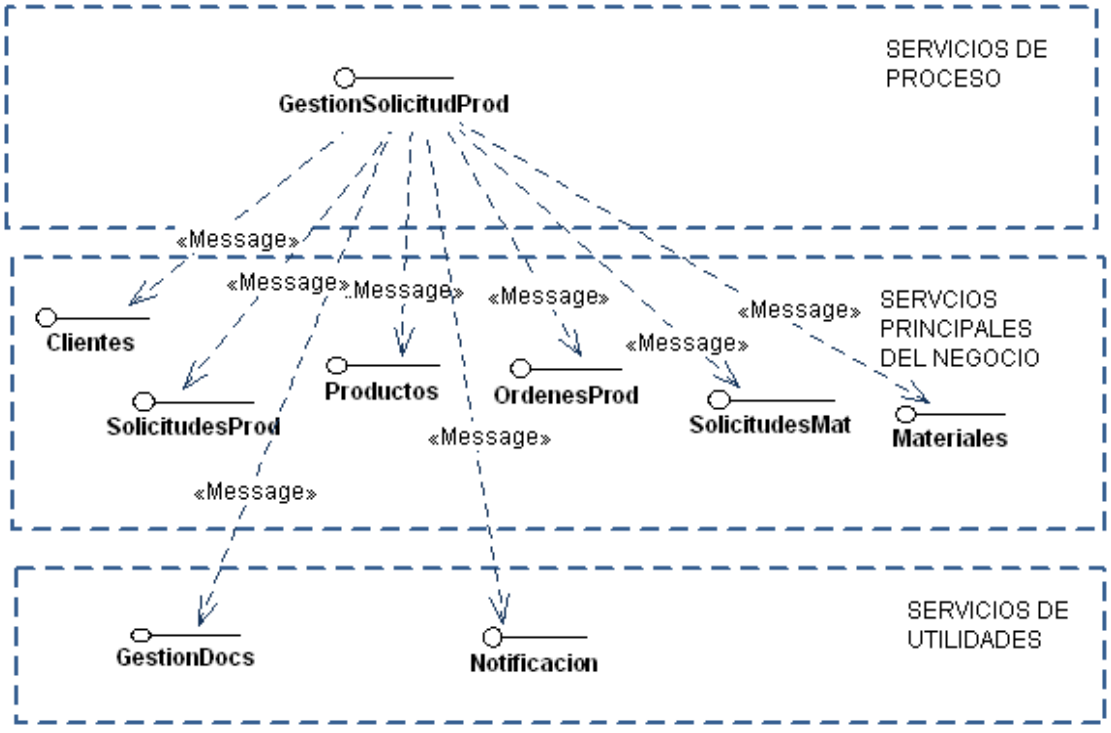

Figura 17. Arquitectura de servicios modificada para el proyecto piloto.

Posteriormente se desarrolló, iterativamente, el diseño para cada uno de los servicios individuales obteniéndose el artefacto de *Especificación de Servicios* para cada uno. En dichos documentos se registraron fundamentalmente los siguientes elementos:

- Aspectos funcionales (descripciones de operaciones y restricciones asociadas).
- Modelo de información.
- Atributos de calidad significativos para el servicio.
- Diagramas de colaboración para operaciones que requieren determinado orden de ejecución.
- Diagramas de actividad por cada operación de servicio que depende de otro servicio.
- Estándares a utilizar en el diseño del servicio.

Respecto a las decisiones de realización, se determinó realizar todos los servicios como servicio web, generando los documentos WSDL correspondientes a cada uno. Los mensajes y tipos de datos se definieron además en documentos XSD, para lo cual se aplicó el patrón "*Esquema Centralizado*" que se describe en la *Ayuda de Herramienta: Generar WSDL y XSD con EA*.

3.7 Resultados de la aplicación de la propuesta en el proyecto piloto

Los resultados obtenidos en el proyecto piloto han permitido realizar una valoración sobre el proceso propuesto para determinar si contribuyó a la calidad en diseño de servicios. La calidad en este caso es entendida, según se planteó en el epígrafe 1.7 como la adherencia a los principios de orientación a servicios y la comprensión, completitud y no ambigüedad de las especificaciones de diseño.

3.7.1 Evaluación de la adherencia a los principios de orientación a servicios

Para comprobar la adherencia a los principios de orientación a servicios fue necesario transformar estos en parámetros medibles o verificables. Para algunos de los principios se utilizaron métricas definidas por (Sindhgatta, y otros, 2009), cuyo cálculo es tomado como base para determinar en qué medida se cumple con el principio.

La notación utilizada para las métricas es la siguiente:

- El dominio del negocio es soportado por un conjunto de procesos de negocio $\boldsymbol{P = \{p_1, p_2 \dots p_p\}}$.
- Un conjunto de servicios $\boldsymbol{S} = \{\boldsymbol{s_1}, \boldsymbol{s_2} \dots \boldsymbol{s_s}\}$ son identificados y diseñado para automatizar los procesos de negocio del dominio.
- Un servicio $\boldsymbol{s} \in \boldsymbol{S}$ provee un conjunto de operaciones $\boldsymbol{O} = \{\boldsymbol{o_1}, \boldsymbol{o_2} \dots \boldsymbol{o_o}\}$ y $\boldsymbol{|O(s)| = O}$.
- Una operación $\boldsymbol{o} \in \boldsymbol{O(s)}$ tiene un conjunto de mensajes de entrada y salida que son usados como contenedores de datos entre el consumidor del servicio y el servicio propiamente. Un mensaje y los tipos de datos que lo constituyen se derivan del modelo de información del dominio. $\boldsymbol{M(o)}$ representa el conjunto de mensajes y datos para la operación $\boldsymbol{o}$. El conjunto de mensajes y sus tipos de datos contenidos para todas las operaciones de un servicio **s** se representa como $\boldsymbol{M(s)} = \cup_{\boldsymbol{o} \in \boldsymbol{O(s)}} \boldsymbol{M(o)}$
- $\boldsymbol{S_{consumidor}(s) = \{Sc_1, Sc_2 \dots Sc_n\}}$, representa al conjunto de consumidores del servicio ***s.***

En los casos en que no se pudo definir métricas por la complejidad o subjetividad del principio, la evaluación se realizó mediante listas de chequeo o de manera cualitativa describiendo las características de los servicios diseñados que soportan el cumplimiento del principio.

Evaluando el acoplamiento en el diseño de servicios

El tipo de acoplamiento sobre el cual se puede influir cuando se diseñan servicios se define como la dependencia de un servicio de otros servicios y para medirlo se propone el uso de la métrica Índice de Acoplamiento entre Servicios **ISCI** (del Inglés, Inter-Service Coupling Index) propuesta por (Sindhgatta, y otros, 2009) y se calcula para un servicio ***s*** como, el número de servicios invocados ese servicio ***s***:

$$\boldsymbol{ISCI} = |\{\boldsymbol{s}'|\exists_{\boldsymbol{o}\in\boldsymbol{s}}, \exists_{\boldsymbol{o}'\in\boldsymbol{s}'}.\boldsymbol{calls}(\boldsymbol{o}', \boldsymbol{o}) \wedge \boldsymbol{s} \neq \boldsymbol{s}'\}|$$

Para los servicios diseñados en el proyecto piloto se obtuvo los siguientes valores de ISCI:

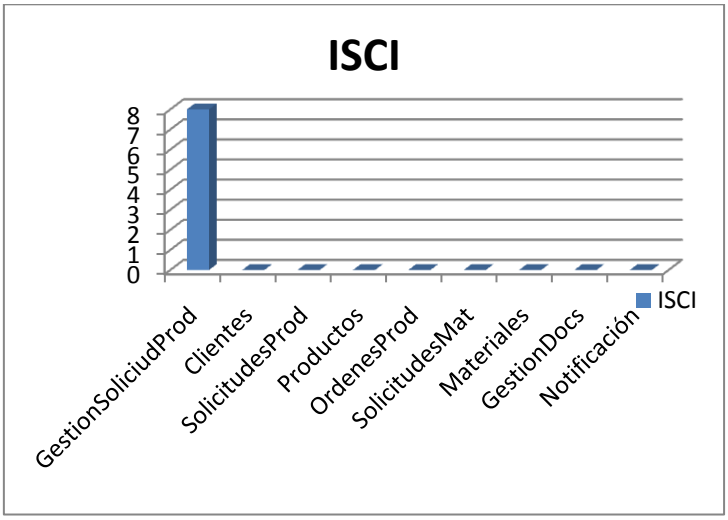

Figura 18. Índice de Acoplamiento entre Servicios en el proyecto piloto.

Como se puede observar solo el servicio *GestionSolicitudProd* tiene un alto acoplamiento ya que como servicio de proceso que necesita consumir los datos y funcionalidades que proveen los otros servicios para llevar a cabo su propósito en función del negocio, el resto de los servicios tienen acoplamiento cero ya que no dependen de ningún otro servicio. Este análisis de los datos revela que existe muy poca dependencia entre los servicios diseñados por lo cual se puede afirmar que se cumple el principio de Bajo Acoplamiento para la gran mayoría de los servicios.

Evaluando la cohesión en los servicios diseñados

Los diseños altamente cohesivos son deseables ya que son más fáciles de analizar y probar y proveen mejor estabilidad y facilidad de cambio lo cual hace a los sistemas más mantenibles (ISO/IEC, 2001).

La cohesión para cualquier sistema mide el grado en el cual los elementos de dicho sistema se relacionan para cumplir con un propósito determinado (Stevens, y otros, 1974). Este concepto es bastante genérico y puede ser aplicado a diferentes niveles de encapsulamiento como módulos, clases, componentes, servicios, etc., pero el modo en que se mide la cohesión debe ser adaptado al contexto. Se propone medir la cohesión en los servicios diseñados mediante la métrica Índice de Cohesión Funcional del Servicio, **SFCI** (del Inglés, Service Functional Cohesion Index) (Sindhgatta, y otros, 2009) y adaptada a las características de los servicios.

$$\boldsymbol{SFCI}(\boldsymbol{s}) = \frac{\mathbf{max}\,(\boldsymbol{\mu}(\boldsymbol{m}))}{|\boldsymbol{O}(\boldsymbol{s})|}$$

$\boldsymbol{\mu}(\boldsymbol{m})$ es el número de operaciones que utilizan un mensaje o tipo de dato común $\boldsymbol{m}$, donde $\boldsymbol{m} \in \boldsymbol{M}(\boldsymbol{s})$ y $|\boldsymbol{O}(\boldsymbol{s})| > \mathbf{0}$.
Esta métrica define la cohesión funcional de las operaciones del servicio basándose en el uso común por parte de estas operaciones, de los mensajes y tipos de datos claves para llevar a cabo sus respectivas funcionalidades. El valor de la métrica se encuentra siempre entre 0(no cohesivo) y 1(altamente cohesivo). Un servicio perfectamente cohesivo es aquel donde todas sus operaciones tienen al menos un mensaje o tipo de dato en común.

Los valores obtenidos de **SFCI** para los servicios diseñados en el proyecto piloto se muestran en la siguiente gráfica:

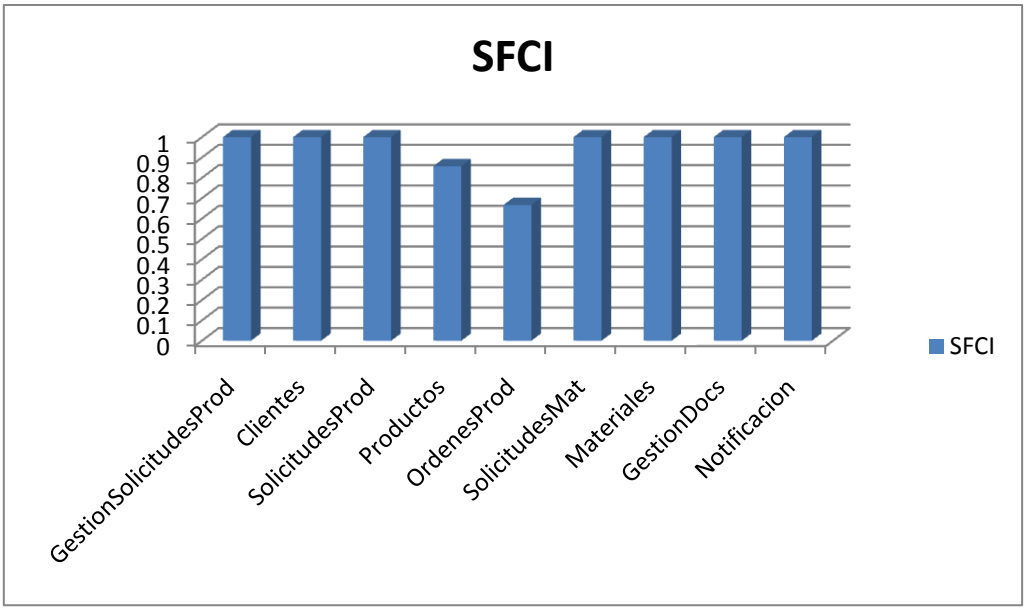

Figura 19. Índice de cohesión funcional de los servicios.

Para los cálculos de la métrica fue necesario realizar un filtrado y tener en cuenta solo los tipos de datos relevantes al negocio ya que de otra manera operaciones no relacionadas pueden aparecer como altamente cohesivas porque comparten, por ejemplo un tipo de dato primitivo (cadena, entero, etc.). Los valores obtenidos para SFCI están por encima del 0.7 (valores muy cercanos a 1) por lo cual se puede considerar un diseño altamente cohesivo.

Evaluando la reutilización en los servicios diseñados

Se propone medir la reutilización a través del Índice de Reutilización del Servicio (SRI, del Inglés Service Reuse Index) cuyo valor se calcula como el número de consumidores existentes del servicio, ya sean procesos de negocio u otros servicios. Se define la métrica como:

$\boldsymbol{SRI(s) = |S_{consumidor}(s)| = P + Q}$, donde

$\boldsymbol{P = |\{s' | \exists_{o \in s}, \exists_{o' \in s'} . calls(o', o) \wedge s \neq s'\}|}$

$\boldsymbol{Q = |\{p \in P | s \in p\}|}$

SRI intenta predecir la futura reutilización del servicio basándose en el uso actual del servicio.

Para los servicios diseñados el comportamiento de SRI se muestra en la siguiente figura:

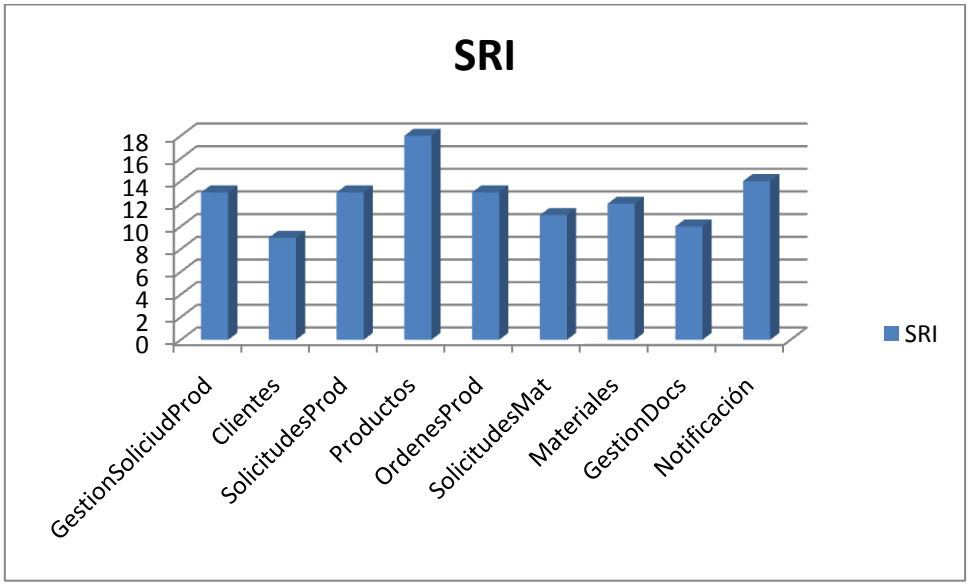

Figura 20. Índice de reutilización de los servicios.

El índice de reutilización es relativamente alto ya aproximadamente el 88% de los servicios diseñados tiene 10 o más posibles consumidores. En el caso de los servicios *Clientes* y *SolicitudesMat* el índice no es significativamente menor, su disminución se debe a que hay un menor número de procesos en la empresa hoy que requiere de la información y funcionalidades que proveen estos servicios, lo cual no limita su uso en el futuro. En el diseño en general, se trató de representar las operaciones del servicio lo más genéricas posibles para que puedan ser usadas incluso si ocurren algunos cambios en los procesos. El desacoplamiento entre la interfaz de los servicios, representada en los documentos WSDL, y los datos y mensajes representados de forma independiente en esquemas XML minimiza el impacto de los cambios si se necesita incluir nuevos atributos y tipos en los mensajes lo cual también favorece la reutilización. Por tanto se puede considerar el diseño de servicios obtenido en un nivel aceptable de reutilización.

Evaluando la abstracción de los servicios

Mientras que otros principios enfatizan en la necesidad de publicar más información sobre el servicio, el principio de abstracción de los servicios,

según Thomas Earl (Erl, 2007), se centra en la necesidad de mantener un contrato conciso y balanceado para prevenir acceso innecesario a detalles adicionales sobre el servicio. Este mismo autor, define además cuatro tipos de información sobre las cuales se debe aplicar la abstracción: tecnología, funcionalidad, lógica de programación y calidad del servicio. A continuación se describe en qué consisten estos tipos de abstracción y se argumenta cómo se aplicaron al diseño realizado para el proyecto piloto con ayuda del proceso de diseño propuesto:

- **Abstracción de la Tecnología:** ocultando la información sobre la tecnología (herramientas de desarrollo, lenguaje de programación, etc.) con la cual se implementa el servicio de manera que se puedan realizar cambios en esa tecnología sin afectar a los usuarios.

  En el proyecto piloto el uso de la tecnología de servicios web facilitó en gran medida este tipo de abstracción ya que los estándares utilizados (WSDL, XSD, WS-Security y otros) están diseñados para ser independientes de las tecnologías, lenguajes y herramientas de implementación. El diseñar primero el contrato del servicio y luego pensar en su implementación, como buena práctica reconocida por para SOA y en la cual se basa el proceso de diseño propuesto permite eliminar la posibilidad de que se definan en la interfaz del servicio aspecto propios de alguna implementación o plataforma en particular. La verificación de conformidad con el Perfil Básico de WS-I, permitió además reforzar esta abstracción eliminando problemas de sintaxis en el uso de los estándares que pudieran acarrear problemas de interoperabilidad en el futuro, facilitando que se puedan hacer cambios en la implementación sin afectar la interfaz definida para el servicio y por tanto evitar la afectación a los consumidores.
- **Abstracción de la Funcionalidad:** determinando cuales de las capacidades de los programas se hacen públicas a través de un contrato técnico (por ejemplo una API), donde las funcionalidades publicadas pueden o no ser iguales a las funcionalidades provistas por los programas.

La tecnología de servicios web, y en particular el estándar WSDL usado en el proyecto piloto, permite intrínsecamente este tipo de abstracción al definir una interfaz que hace visible sólo las funcionalidades deseadas de los componentes subyacentes.

- **Abstracción de la Lógica de Programación:** se refiere al ocultamiento de los detalles de diseño de bajo nivel como algoritmos, rutinas de autenticación y manejo de errores internos de los componentes y programas que implementan tan el servicio.

  Este tipo de abstracción también es soportado por la tecnología de servicios web aplicada en el proyecto piloto. También contribuyó a ella la definición del artefacto Perfil de Servicio (Anexo 6) en que se publica sólo la información del servicio relevante a los consumidores (funcionalidades, formas de interactuar con el servicio, etc.) ocultando los detalles de implementación y lógica interna.
- **Abstracción de la Calidad del Servicio:** establece que el consumidor del servicio solo debe conocer aquellos atributos de calidad que son relevantes para él, mientras se mantienen ocultos otros de bajo nivel como tiempo de acceso a las bases de datos, frecuencias de errores en los programas o componentes subyacentes, etc.

  En el artefacto *Perfil de Servicio*, definido en el proceso de diseño, y aplicado a cada uno de los servicios del proyecto piloto se enuncian los atributos de calidad del servicio relevantes a los consumidores (tiempo de respuesta para cada una de las funcionalidades, confiabilidad, disponibilidad, etc.) ocultando cómo deben manejar estos atributos cada uno de los componentes y sistemas que soportarán el servicio. Estos perfiles de servicios son visibles a los consumidores mientras que el artefacto *Especificación de Servicio*, que contiene elementos técnicos más detallados, es visible a los arquitectos, diseñadores e implementadores del servicio.

Por todo lo anterior se puede afirmar que los servicios diseñados poseen un nivel adecuado de abstracción.

Evaluando la autonomía de los servicios

Thomas Earl (Erl, 2007) define la autonomía de los servicios como el control que estos pueden ejercer sobre el entorno de ejecución y, como forma de medirla identifica cuatro categorías de autonomía:

- **Autonomía del Contrato del Servicio:** se refiere a que los contratos deben diseñarse alineadamente unos con otros para evitar el solapamiento de las funcionalidades expresadas.
- **Autonomía Compartida:** este tipo de autonomía está presente cuando la lógicas y recursos de la implementación subyacente del servicio es compartida con otra parte de la empresa.
- **Autonomía de la Lógica del Servicio:** cuando la lógica encapsulada en el servicio es exclusiva de este pero las fuentes de datos que se manejan en la implementación subyacente son compartidas con otras partes de la empresa.
- **Autonomía Pura:** es considerado el nivel más alto de autonomía. Se alcanza cuando lógica, datos y recursos de la implementación del servicio son dedicados a este exclusivamente.

Las tres últimas categorías de autonomía se centran en la implementación del servicio y los recursos de hardware y software subyacentes por lo tanto, a nivel de diseño, solo es posible verificar la primera categoría: Autonomía del Contrato. Para evaluar esta categoría de autonomía se utilizará una lista de chequeo (Anexo 13) basada en la propuesta realizada por Rosen y otros en su libro *Applied SOA: Service-Oriented Architecture and Design Strategies* (Rosen, y otros, 2008) .

El análisis del primer indicador arrojó que en ninguno de los servicios diseñados existía solapamiento de funciones, en el segundo indicador se comporta como se muestra en el siguiente gráfico:

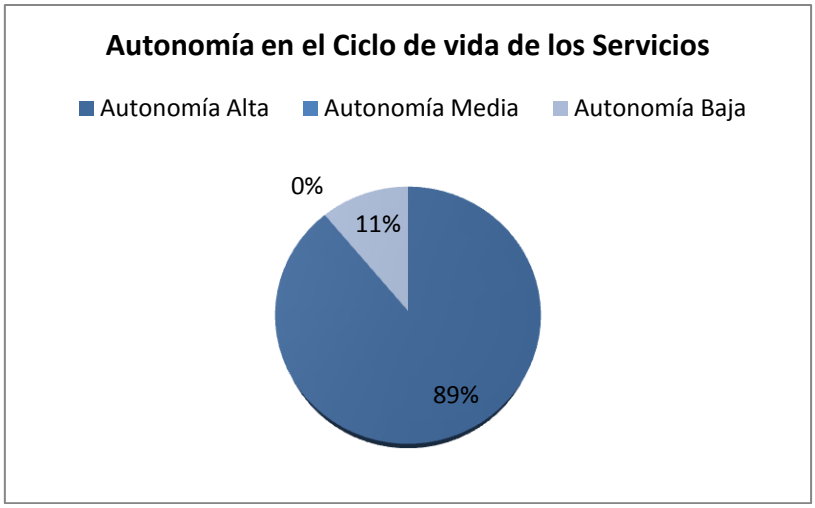

Figura 21. Autonomía del ciclo de vida de los servicios.

Un alto por ciento de servicios presenta una autonomía alta. Los valores de autonomía baja (11% de los servicios) corresponde a aquellos servicios que presentan algún tipo de dependencia respecto a otros, los cual disminuye en alguna medida su control total sobre el medio de ejecución.
De acuerdo a los datos obtenidos se puede considerar que el diseño realizado cumple en alto grado con el principio de Autonomía de los Servicios.

Evaluando la gestión de estados en los servicios

Para evaluar si un servicio mantiene o no estado es necesario hacer la pregunta: ¿guarda el servicio algún tipo de información (de sesión, información de contexto, etc.) entre llamadas a sus operaciones?

Entre los requisitos de negocio para el proyecto piloto no se identificaron necesidades críticas para la gestión de estados del tipo transaccional. Entre los aspectos que influyeron en la toma de decisiones en la gestión de estados en los servicios estuvieron los siguientes:

- Gestión de sesión (autenticación, autorización).
- Reglas de contexto para la ejecución de una operación determinada.
- Tareas complejas realizadas por un solo servicio.

Para la gestión de sesión y reglas de contexto se decidió utilizar el estándar WS-Security, con el cual permite incorporar datos de este tipo o certificados de seguridad al encabezado de los mensajes SOAP, por lo cual la autenticación y

autorización necesaria para todos los servicios viaja con los mensajes y el servicio no necesita guardar este tipo de información hasta una nueva llamada.

Respecto a las tareas complejas como las que realiza cualquiera de los procesos de la organización que requiere varias llamadas a servicios diferentes, se delega la responsabilidad en la arquitectura a la capa de orquestación de servicios y a las aplicaciones que invocan servicios, dejando a estos últimos las tareas más atómicas para responder a pedidos de información o realización de acciones puntuales. De esta manera se evitó en el diseño la incorporación de estados a los servicios.

Evaluando la estandarización del contrato

Las principales áreas en las cuales se aplica la estandarización a los servicios son (Erl, 2007):

- Expresión funcional.
- Representación de datos.
- Políticas.

La expresión funcional no es más que la forma en que el servicio expresa las acciones que puede realizar. En este caso para responder a la estandarización se utilizó la convención de nombres definida en *las Políticas de Diseño* para la nomenclatura de los servicios y operaciones. Para la descripción técnica de la interfaz del servicio se utilizó el estándar WSDL 1.1 verificando, con el uso de la herramienta plug-in SOAPUI de eclipse, su conformidad con el perfil básico de WS-I con el fin de garantizar la interoperabilidad con las diversas plataformas de software desde las cuales se puede consumir el servicio.

Para la estandarización en la representación de los datos aplicó la convención de nombres especificada en las *Políticas de Diseño*. Los datos que utilizan los mensajes de cada servicio están descritos utilizando el formato estándar XSD (del Inglés XML Schema Definition). Se aplicó además el patrón *Esquema Centralizado,* con lo cual se reutilizan las estructuras de datos definidas en el *Modelo Canónico de Datos*. Para otros datos que necesita el servicio y que no pertenecen al modelo canónico, se definieron esquemas XSD adicionales propios para el servicio.

Para las políticas de seguridad definidas se utilizaron los estándares WS-Security y SSL que hacen posible que la autenticación y autorización ya que exigen un certificado de seguridad al consumidor del servicio.

Por todo lo anterior podemos afirmar que toda la interfaz y envoltura de los servicios diseñados están basadas en estándares lo cual nos permite afirmar que se cumple el principio de estandarización del contrato del servicio.

Evaluando facilidad de descubrimiento de los servicios

En el contexto de la orientación a servicios el descubrimiento se refiere al proceso de búsqueda y localización de lógica de solución en un entorno determinado, mientras que el principio de facilidad de descubrimiento[20] establece que los servicios son provistos de metadatos comunicativos mediante los cuales pueden ser efectivamente descubiertos e interpretados (Erl, 2007).

Cuando los servicios no pueden ser descubiertos e interpretados de manera efectiva esto puede traer las siguientes consecuencias:

- Los usuarios pierden la oportunidad de reutilizar un recurso existen y disponible y terminan construyendo el suyo propio.
- Se construyen y despliegan nuevos recursos que se solapan en funcionalidad con los ya existentes no descubiertos introduciendo redundancias en la empresa.

Entre las acciones que deben aplicarse durante el diseño de servicios para lograr la facilidad de descubrimiento están (Erl, 2007):

- Los contratos de servicios son equipados con los metadatos apropiados que serán correctamente referenciados cuando se realicen búsquedas.
- Los contratos de servicios son revestidos de información que comunica claramente su propósito y capacidades a los humanos.
- Si existe un Registro de Servicios este debe ser llenado con la misma atención que se describió la meta información.

[20] Del Inglés *Discoverability*

- Si no existe un Registro de Servicios, se deben escribir documentos de perfil de servicio para complementar el contrato del servicio y formar las bases para un futuro registro.

En el proyecto piloto, para lograr comunicar el propósito del servicio y cada una de las capacidades (operaciones) que este ofrece, se utilizó el elemento *wsdl:document* definido en el estándar WSDL 1.1 (Christensen, y otros, 2001) con el fin de documentar textualmente en lenguaje natural cualquier aclaración o descripción que facilitara el consumo del servicio. Con este elemento se describió, de manera intrínseca en el WSDL, el propósito general del servicio y en qué consisten cada una de las operaciones. También se generó un perfil estandarizado para cada servicio, tal como se recomienda en la guía Diseñar Interfaz de Servicio, que contiene los elementos más significativos de la Especificación del Servicio que puedan interesar a los consumidores. Dicho perfil debe publicarse junto con el WSDL del servicio en el registro/repositorio a utilizar, y las instrucciones para esta publicación se explican en la sección *Instrucciones para el Despliegue* de cada una de las especificaciones del servicio.
Con estas medidas tomadas se puede asegurar en gran parte el descubrimiento de los servicios diseñados.

Evaluando la capacidad de composición en los servicios diseñados

El principio de capacidad de composición de los servicios diseñados se define de la siguiente manera (Erl, 2007):
"Los servicios son participantes efectivos en composiciones a pesar del tamaño y complejidad de la composición."

Una de las características de este principio es que tiene una fuerte dependencia del resto de los principios de orientación a servicios por lo que su cumplimiento debe estar en función del cumplimiento de estos. La relación de dependencia se muestra en la Figura 22 y se detalla más en el Anexo 15.

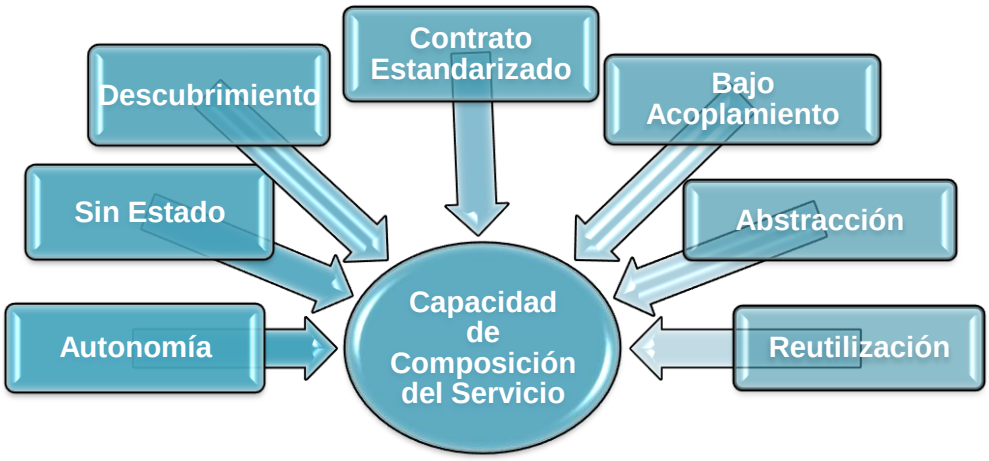

Figura 22. Relación del principio *Capacidad de Composición del Servicio* con el resto de los principios de orientación a servicios.

Para medir la aplicación de este principio utilizó la lista de chequeo definida por (Erl, 2007) y detallada el Anexo 16. En esta lista de chequeo que se verifica, para cada una de las operaciones de los servicios, el cumplimiento de un grupo de características que inciden directamente en la capacidad de composición de dicho servicio. La figura 23 muestra los resultados de aplicación de la lista de chequeo a un total de 39 operaciones de servicios definidas en el proyecto piloto.

Se puede apreciar que, en su mayoría, las operaciones de servicios cumplen con los elementos significativos de los principios que influencian la capacidad de composición de los servicios por lo que se puede afirmar que los servicios que contienen a dichas operaciones cumplen con el principio de capacidad de composición.

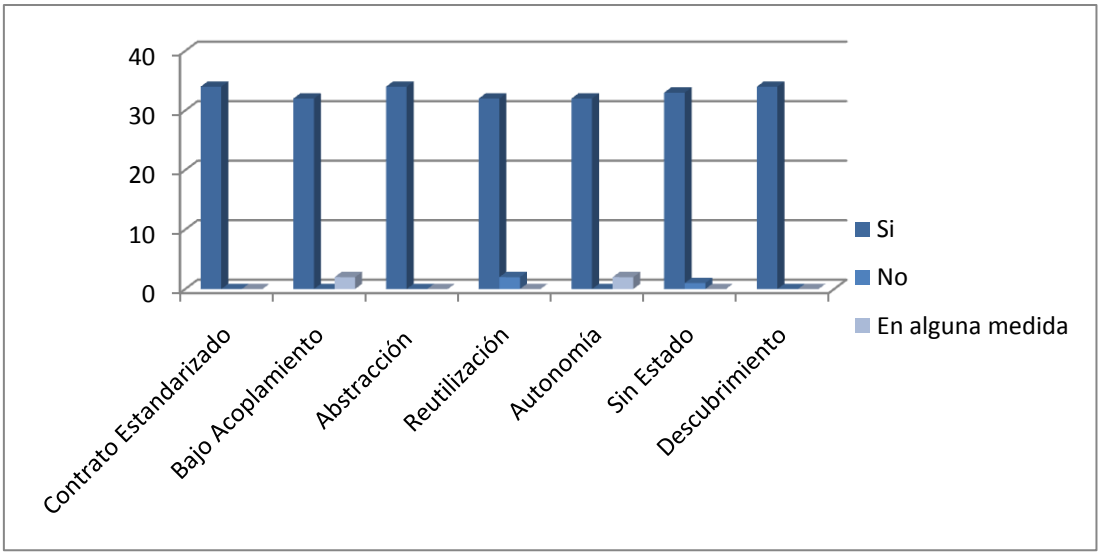

Figura 23. Cumplimiento de las características de los principios de orientación a servicios que influencian el principio de *Capacidad de Composición.*

Con los resultados obtenidos de las evaluaciones anteriores podemos concluir que la con la aplicación del proceso propuesto se logró una adecuada adherencia a los principios de orientación a servicios definidos en el epígrafe 1.7.

3.7.2 Validación de la completitud y comprensión de las especificaciones

Para validar la calidad del diseño realizado con la aplicación del proceso de diseño propuesto, como se mencionó anteriormente en el epígrafe 1.7, también debe garantizarse la completitud y comprensión de las especificaciones realizadas. Para ello se aplicó una encuesta a 10 integrantes del equipo de desarrollo del proyecto piloto (Anexo 14) donde se pudo constatar, como se muestra en la figura siguiente, que el 100% de los encuestados califican la completitud y comprensión de las especificaciones con valores entre “Alto” y “Medio”, ninguno las consideró “Baja” y, de ellos, entre el 70% y el 90% considera “Alta” la completitud y comprensión de las especificaciones de de servicios realizadas.

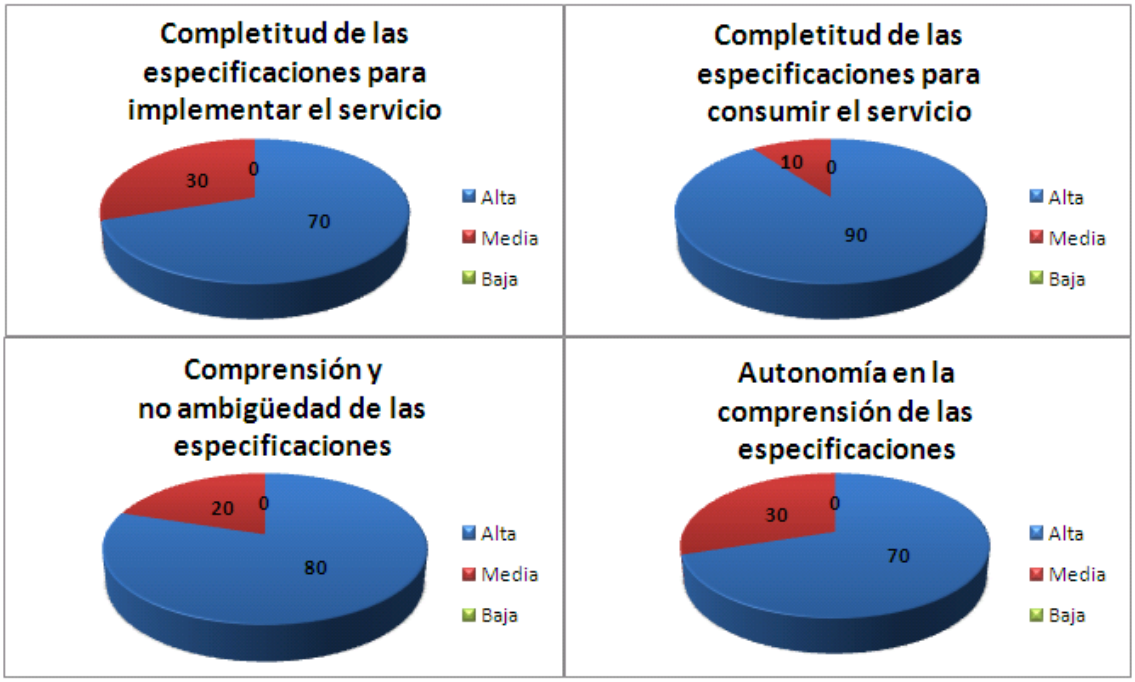

Figura 24. Resultados de las encuestas para medir completitud y comprensión de las especificaciones de servicios realizadas.

Con estos resultados obtenidos en las encuestas obtenemos evidencia acerca de que las especificaciones de servicios realizadas mediante la aplicación del proceso de diseño propuesto poseen un alto grado de completitud y facilidad de comprensión lo cual contribuye a la calidad del diseño realizado.

3.8 Lecciones aprendidas del piloto

Durante la aplicación de la propuesta se identificó una serie de aspectos a tener en cuenta en futuros proyectos SOA y que pueden ser muy útiles a la hora de diseñar una estrategia de mejora del proceso de diseño de servicios. Entre los principales aspectos y lecciones aprendidas se encuentran:

- La comunicación en el equipo juega un papel fundamental para el éxito del proceso, independientemente de lo exhaustivo que puedan ser los artefactos y especificaciones realizadas. Los diseñadores constituyen un vínculo indispensable entre los arquitectos e implementadores de los servicios por lo que deben mantenerse en contacto con ambos para lidiar con los requerimientos de cada uno.

- Es recomendable, también para los diseñadores, mantenerse en contacto con los analistas ya que, por exhaustivas que puedan ser las especificaciones de negocio, siempre pueden contener omisiones y ambigüedades.
- El conocimiento del proceso de diseño, técnicas y patrones de diseño así como el dominio de las herramientas de software empleadas juega un papel determinante en el rendimiento del proceso (tiempo que demora su ejecución completa y tiempo empleado en diseñar un servicio).
- Con realización de algunos ajustes a las herramientas empleadas en diseño, como por ejemplo definición de metadatos para los servicios en la herramienta de modelado y creación de plantillas personalizadas para la generación automática de reportes, se podría obtener un rendimiento mucho mayor en el diseño y se reducirían los riesgos de inconsistencias en documentos relacionados (por ejemplo: *Especificación de Servicio* y *Perfil de Servicio*) cuando se realizan algún cambios en alguno de ellos.

Con el análisis de los resultados presentados en este apartado se puede concluir que los objetivos definidos para la investigación fueron alcanzados, lo cual se corrobora mediante el logro de la adherencia a los principios de orientación a servicios y la completitud y no ambigüedad de las especificaciones de diseño, siendo estas las dimensiones definidas para la variable dependiente **calidad adecuada del producto del diseño de servicios**.

El cumplimiento del objetivo general mediante propuesta realizada, su aplicación en un proyecto piloto y los resultados obtenidos en dicha aplicación han permitido aportar evidencia empírica a favor de la hipótesis planteada. Sin embargo se considera que para una completa validación de dicha hipótesis sería necesario la aplicación del proceso propuesto en un número mayor de proyectos para lo cual se requiere un tiempo considerable y la existencia en dichos proyectos un grupo de condiciones mínimas necesaria para poder aplicar el proceso de diseño de servicios de las cuales actualmente no se dispone.

3.9 Consideraciones del capítulo

La evaluación realizada al diseño de servicios que se obtuvo como resultado de la aplicación de la propuesta arrojó resultados favorables acerca de la adherencia de este diseño a todos los principios de orientación a servicios definidos para esta investigación.

Mediante la encuesta realizada a los desarrolladores en el proyecto piloto se comprobó que el diseño realizado con el proceso propuesto presenta un alto grado de completitud y facilidad de comprensión.

Lo anterior permite afirmar que el proceso de diseño propuesto contribuye al logro de una adecuada calidad del diseño quedando demostrado el cumplimiento del objetivo trazado para la investigación, lo cual aporta evidencia empírica a favor de la hipótesis propuesta.

La ejecución del proceso de diseño propuesto en un proyecto real permitió valorar aspectos y dificultades reales que pueden influir en el éxito de la aplicación del proceso de diseño y permitió identificar un conjunto de “lecciones aprendidas” a tener en cuenta en futuros proyectos y en estrategias de mejora.

Conclusiones

- En la presente investigación se realizó el marco teórico de la investigación donde se abordaron los principales concepto asociados a la arquitectura orientada a servicios que sirvieron de fundamento para ofrecer una solución al problema de investigación planteado.
- Se analizó el estado actual de SOA y el diseño de servicios en Cuba con lo cual se pudo constatar que el conocimiento de este paradigma en el sector empresarial es aún muy básico. Sin embargo se está comenzando a manifestar la necesidad de su aplicación para lo cual resulta indispensable disponer de un proceso o metodología que sirva de guía en la aplicación de los principios de orientación a servicios.
- Se analizaron varias de las principales metodologías de desarrollo SOA profundizándose en el modo en que estas abordan el diseño de servicios. Dicho análisis permitió evidenciar las fortalezas y debilidades que presenta cada una de estas metodologías y seleccionar los aspectos positivos de cada una para conformar una propuesta que superara sus limitaciones en cuanto al diseño de servicios.
- Se definió un proceso de diseño que contiene tareas, roles, herramientas y guías que, de manera detallada, ayudan a la realización del diseño de servicios teniendo en cuenta los principios de orientación a servicios.
- La ejecución del proceso de diseño propuesto en un proyecto real permitió valorar aspectos y dificultades reales que pueden influir en el éxito de la aplicación del proceso de diseño y permitió identificar un conjunto de “lecciones aprendidas” a tener en cuenta en futuros proyectos y en estrategias de mejora.
- Luego de la evaluación realizada al diseño de servicios que se obtuvo producto de la aplicación del proceso propuesto y, teniendo en cuenta los resultados del análisis de las encuestas aplicadas a los desarrolladores en el proyecto piloto se puede afirmar que el proceso de diseño propuesto contribuye al logro de una adecuada calidad del diseño quedando demostrado el cumplimiento del objetivo trazado para la

investigación, lo cual aporta evidencia empírica a favor de la hipótesis propuesta.

Recomendaciones

- Aplicar el proceso de diseño propuesto en proyectos SOA con el fin de obtener estadísticas y métricas que permitan emprender una estrategia de mejora.
- Mejorar las herramientas propuestas a través de configuraciones (por ejemplo para generación automática de reportes y código XML), implementación de complementos y adaptaciones que permitan obtener una mayor eficiencia y eficacia en el proceso de diseño de servicios.
- Formar nuevos diseñadores, arquitectos de servicios y arquitectos de seguridad con los conocimientos básicos que necesitan para la aplicación de este proceso y favorecer la superación de los que actualmente ejercen estos roles.

Referencias

Allen, P. 2007. The Service Oriented Process. *CBDi Journal.* [En línea] Febrero de 2007. [Citado el: 12 de Diciembre de 2009.] http://www.cbdiforum.com/secure/interact/2007-02/service_oriented_process.php.

Amsden, Jim. 2007. Modeling SOA: Part 2. Service Specification. [En línea] 9 de Octubre de 2007. [Citado el: 12 de Diciembre de 2009.] http://www.ibm.com/developerworks/rational/library/07/1009_amsden/.

Artus, David J N. 2006. SOA realization: Service design principles. [En línea] 2006. [Citado el: 25 de Febrero de 2010.] http://www.ibm.com/developerworks/webservices/library/ws-soa-design/.

Bell, Michael. 2010. *SOA Modelling Patterns for Service-Oriented Discovery and Analysis.* Hoboken : John Wiley & Sons, Inc., 2010.

Brittenham, P., y otros. 2007. IT service management architecture and autonomic computing. *IBM Systems Journal.* Riverton : IBM Corp. , 2007. Vol. 3, 46, págs. 565-581.

CBDI. 2007. *CBDI-SAE META MODEL FOR SOA VERSION 2.0.* 2007.

Chaviano G., E y Carrascoso P., Y. 2008. Propuesta de Arquitectura Orientada a Servicios para el Módulo de Inventario del ERP Cubano. [Trabajo de Diploma]. La Habana : Universidad de las Ciencias Informáticas, 2008.

Christensen, Erik, y otros. 2001. Web Services Description Language (WSDL) 1.1. [En línea] 2001. [Citado el: 10 de Abril de 2010.] http://www.w3.org/TR/2001/NOTE-wsdl-20010315.

Demirkan, Haluk, y otros. 2008. Service-oriented technology and management: Perspectives on research and practice for the coming decade. Elsevier B.V., 2008.

Domínguez Jimenez, J. J., y otros. 2007. El reto de los servicios Web para el software libre. [En línea] 2007. [Citado el: 12 de Diciembre de 2009.] http://www.willydev.net/InsiteCreation/v1.0/WillyCrawler/2008.06.07.Articulo.El reto de los servicios Web para el software libre.pdf.

Earl, Thomas. 2005. *Service-Oriented Architecture. Concepts, Technology and Design.* Prentice Hall PTR, 2005.

Erl, Thomas. 2007. *SOA, Principles of Service Design,.* Prentice Hall, 2007.

Fielding, Roy Thomas. 2000. Architectural Styles and the Design of Network-based Software Architectures. [Tesis Doctoral]. Irvine : Universidad de California, 2000.

Grid Today.IDC: SOA. 2006. SOA offers tremendous opportunity for service providers. [En línea] 10 de Abril de 2006. [Citado el: 10 de Diciembre de 2009.] http://www.gridtoday.com/grid/617472.html.

Hailstone, Rob. 2009. Applying Normalisation to SOA Service Design. *Sitio Web del Butler Group.* [En línea] 24 de Agosto de 2009. [Citado el: 20 de Marzo de 2010.] http://www.butlergroup.com/research/KCInterPages/%7B8509CC86-E086-4BFF-8635-BF7AA9B054FB%7D.asp.

IBM. 2003. *Best Practice: Continuously Verify Quality.* [Ayuda de Rational Unified Process] Junio de 2003.

IBM. 2007. Rational Unified Process. [Software de Ayuda]. 2007.

ISO/IEC. 2001. ISO/IEC 9126-1:2001 Software Engineering Product Quality – Quality Model. Geneva : International Standards Organization, 2001.

Klose, K., Knackstedt, R. y Beverungen, D. 2007. Identification of services - A stakeholder-based approach to SOA development and its application in the area of production planning. [ed.] H. Oesterle, J. Schelp y R. Winter. *ECIS '07.* St. Gallen. 2007.

Kohlborn, Thomas, y otros. 2009. Service Analysis: A Critical Assesment of the State of the Art. [En línea] 2009. [Citado el: 22 de Enero de 2010.] http://eprints.qut.edu.au/26755/1/26755.pdf.

Linthicum, Dave. 2006. SOA service design: Developing for the future. *Infoworld Inc.* [En línea] 8 de Mayo de 2006. [Citado el: 19 de Enero de 2010.] http://www.infoworld.com/t/architecture/soa-service-design-developing-future-219.htm.

M.Josuttis, Nicolai. 2007. *Soa in Practice.* Sebastopol : O'Reilly Media, Inc., 2007.

MacVittie, Lori. 2006. Taking a REST from SOAP. *Network Computing.* 5 de Octubre de 2006. págs. 17-20.

McFeeley, B. 1996. IDEAL: A User's Guide for Software Process Improvement. Software Engineering Institute. Carnegie Mellon University, 1996.

McGlaughlin, R. 1991. Some Notes on Program Design. *Software Engineering Notes.* 1991, Vol. 16, págs. 53-54.

Microsoft. 2007. Definición de Gobierno. [En línea] Microsoft Technet. SharePoint Server TechCenter, 2007. [Citado el: 30 de marzo de 2010.] http://technet.microsoft.com/es-es/library/cc263356.aspx..

OASIS. 2006. Reference Model for Service Oriented Architecture 1.0. [En línea] Agosto de 2006. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm.

OASIS. 2007. Standards and other approved work. [En línea] 2007. http://www.oasis-open.org/specs.

O'Brien, Liam, Bass, Len y Merson, Paulo. 2005. Quality Attributes and Service-Oriented Architectures. [En línea] Septiembre de 2005. http://www.sei.cmu.edu/publications/documents/05.reports/05tn014.html..

Panian, Zeljko. 2005. *The business value propositions of service-oriented architectures.* Athens, Greece : World Scientific and Engineering Academy and Society (WSEAS), 2005. Proceedings in 9th WSEAS International Conference on Computers.

Papazoglou, Michael P. y van den Heuvel, Willem-Jan. 2006. Service-Oriented Design and Development Methodology. 2006.

Paulson, Linda D. 2006. Services Science: A New Field for Today's Economy. [ed.] Lee Garber. *Computer. Industry Trends.* Agosto de 2006.

Plummer, D. C., y otros. 2008. Gartner's top predictions for IT organizations and users, 2008 and beyond: going green and self-healing. *Gartner Inc., Stamford, CT.* [En línea] 8 de Enero de 2008. www.gartner.com/it/page.jsp?id=593207.

Pressman, Roger S. 2005. *Ingeniería del Software: Un Enfoque Práctico.* Quinta edición. La Habana : Felix Varela, 2005. págs. 132-133.

Ramollari, Ervin, Dranidis, Dimitris y Simons, Anthony J. H. 2007. A Survey of Service Oriented Development Methodologies. 2007.

Reynoso, Carlos y Kicillof, Nicolás. 2004. Estilos y Patrones en la Estrategia de Arquitectura de Microsoft (Versión 1.0). Universidad de Buenos Aires, 2004.

Ricken, Jan y Petit, Michaël. 2009. *Characterization of Methods for Process-Oriented Engineering of SOA.* [ed.] D. Ardagna et al. Berlin : Springer-Verlag, 2009.

Rigñack, Q. A. 2008. Estudio de las capacidades de modelación en las tecnologías BPM BizTalk Server y Oracle BPA Suite. La Habana : Universidad de las Ciencias Informáticas, 2008.

Rosen, Mike, y otros. 2008. *Applied SOA: Service-Oriented Architecture and Design Strategies.* Indianapolis : Wiley Publishing, Inc., 2008.

Sánchez, L. A. y Lamoth, K. E. 2008. Propuesta de guía para adoptar una arquitectura orientada a servicios en la UCI. [Trabajo de Diploma]. La Habana : Universidad de las Ciencias Informáticas, 2008.

SEI. 2006. *CMMI for Development. Version 1.2.* Software Engineering Institute. 2006.

SEI. 2007. Software Architecture Glossary. *Software Engineering Institute.* [En línea] 2007. http://www.sei.cmu.edu/architecture/glossary.html.

Sindhgatta, Renuka, Sengupta, Bikram y Ponnalagu, Karthikeyan. 2009. Measuring the Quality of Service Oriented Design. *7th International Joint Conference on Service-Oriented Computing.* Estocolmo, 2009. págs. 485-499.

Stevens, W., Myers, G. y Constantine, L. 1974. Structured Design. *IBM Systems Journal.* 1974. Vol. 13, págs. 115-139.

UCI. 2007. Decisiones e Intensiones I Taller de Arquitectura de Software. La Habana : Universidad de las Ciencias Informáticas, 2007.

Wall, Q. 2007. Rethinking SOA Governance. [En línea] 14 de Marzo de 2007. [Citado el: 10 de 12 de 2009.] http://www.oracle.com/technology/pub/articles/a2a/2007/03/soa-governance.htm.

Weerawarana, S., y otros. 2005. *Web Services Platform Architecture: SOAP, WSDL, WS-Policy, WS-Addressing, WS-BPEL, WS-Reliable Messaging, and More.* Prentice Hall PTR, 2005.

World Wide Web Consortium. 2007. Technical reports and publications. [En línea] 2007. http://www.w3.org/TR.

WS-I Committee. 2006. WS-I Frequently Asked Questions. *Sitio web de Web Service Interoperability Organization.* [En línea] 2006. [Citado el: 20 de Enero de 2010.] http://www.ws-i.org.

Zhu, Liming, y otros. 2007. Effects of Architecture and Technical Development Process on Micro-process. *Software Process Dynamics and Agility.* Springer Berlin / Heidelberg, 2007, págs. 49-60.

Anexos

Anexo 1: Algunos de los estándares asociados a servicios web más usados.

- **SOAP:** Recomendación del W3C. Especifica la estructura de los mensajes que intercambian los servicios web. Es independiente de la plataforma, flexible y fácilmente extensible.
- **WSDL:** Recomendación candidata del W3C. Su propósito es proveer una estándar de la interfaz de los servicios web. Permite definir la estructura de los mensajes SOAP que intercambiará el servicio que describe con otros servicios web.
- **UDDI:** Estándar de OASIS. Permite mantener repositorios de especificaciones WSDL simplificando el descubrimiento del servicio web y el acceso a sus especificaciones. Hace posible que una aplicación busque dinámicamente servicios que ofrezcan una serie de características, seleccione el más adecuado (por coste, calidad, etc.) e incluso localice servicios alternativos si uno falla.
- **WS-Addressing:** Recomendación del W3C. Permite incluir en un mensaje SOAP información sobre el emisor, los destinatarios, a quién se debe responder, a quién se debe informar en caso de error, etc. Con estos elementos se consigue un direccionamiento independiente de los mensajes con respecto a capas de transporte inferiores como HTTP.
- **WS-AtomicTransaction:** Borrador para revisión pública de OASIS. Suele emplearse en la coordinación de servicios web que realicen actividades de corta duración en entornos de confianza. Las acciones a realizar por cada servicio implicado se agrupan en una transacción atómica. El coordinador decide cuándo realizarla y puede abortar una transacción en curso devolviendo a los servicios web implicados a su estado previo.
- **WS-BPEL:** Especificación de OASIS. Define un lenguaje que facilita la composición de servicios web. Permite especificar la lógica de la composición de los servicios (envío de mensajes, sincronización,

iteración, tratamiento de transacciones erróneas, etc.) independientemente de su implementación.

- **WS-Coordination:** Borrador para revisión pública de OASIS. Permite crear contextos de coordinación para la sincronización de servicios web. Requiere protocolos complementarios como WS-AtomicTransaction.
- **WS-Policy:** Borrador de trabajo del W3C. Proporciona un medio de especificar las características que presentan y exigen los servicios web durante su operación. Por ejemplo, un determinado servicio puede exigir para operar que los mensajes se firmen o cifren con determinados algoritmos o que la coordinación se realice mediante un protocolo dado. Con esto se dota a los servicios web de la capacidad de negociar entre ellos las condiciones de interacción.
- **WS-Reliability:** Estándar de OASIS. Es un protocolo que permite numerar los mensajes SOAP y obtener confirmación de su recepción en el destino. Con esto se puede garantizar el orden de recepción de los mensajes, evitar duplicados, comprobar la entrega, etc.
- **WS-ReliableMessaging:** Borrador para revisión pública de OASIS. Esta especificación define un protocolo que permite el intercambio de mensajes de manera fiable en presencia de fallos en el software, la red, etc.
- **WS-Security:** Estándar OASIS. Proporciona integridad, confidencialidad y autenticación en las comunicaciones entre servicios web. Incluye varios protocolos de seguridad, como X.509 y Kerberos, de manera que los servicios web puedan utilizar distintas políticas de seguridad.
- **WS-Trust:** Borrador para revisión pública de OASIS. Su objetivo es facilitar el intercambio de series de mensajes seguros mediante la emisión, renovación y validación de *tokens* de diversos protocolos.

Anexo 2: Algunos de los perfiles de interoperabilidad que propone WS-I.

- **Basic Profile:** Establece un conjunto guías y clarificaciones para un grupo de especificaciones y estándares de servicios web como son SOAP, WSDL, UDDI, XML Schema, HTTPS) que deberían ser usados juntos para desarrollar servicios interoperables.
- **Attachments Profile**: Complementa el Perfil Básico con clarificaciones para lograr la interoperabilidad en servicios que tratan con mensajes SOAP que contienen adjuntos.
- **Simple SOAP Binding Profile**: Incorpora nuevas actualizaciones y rectificación de errores al Perfil Básico sobre la serialización de los mensajes SOAP.

Anexo 3: Proceso para diseñar servicios según Thomas Earl.

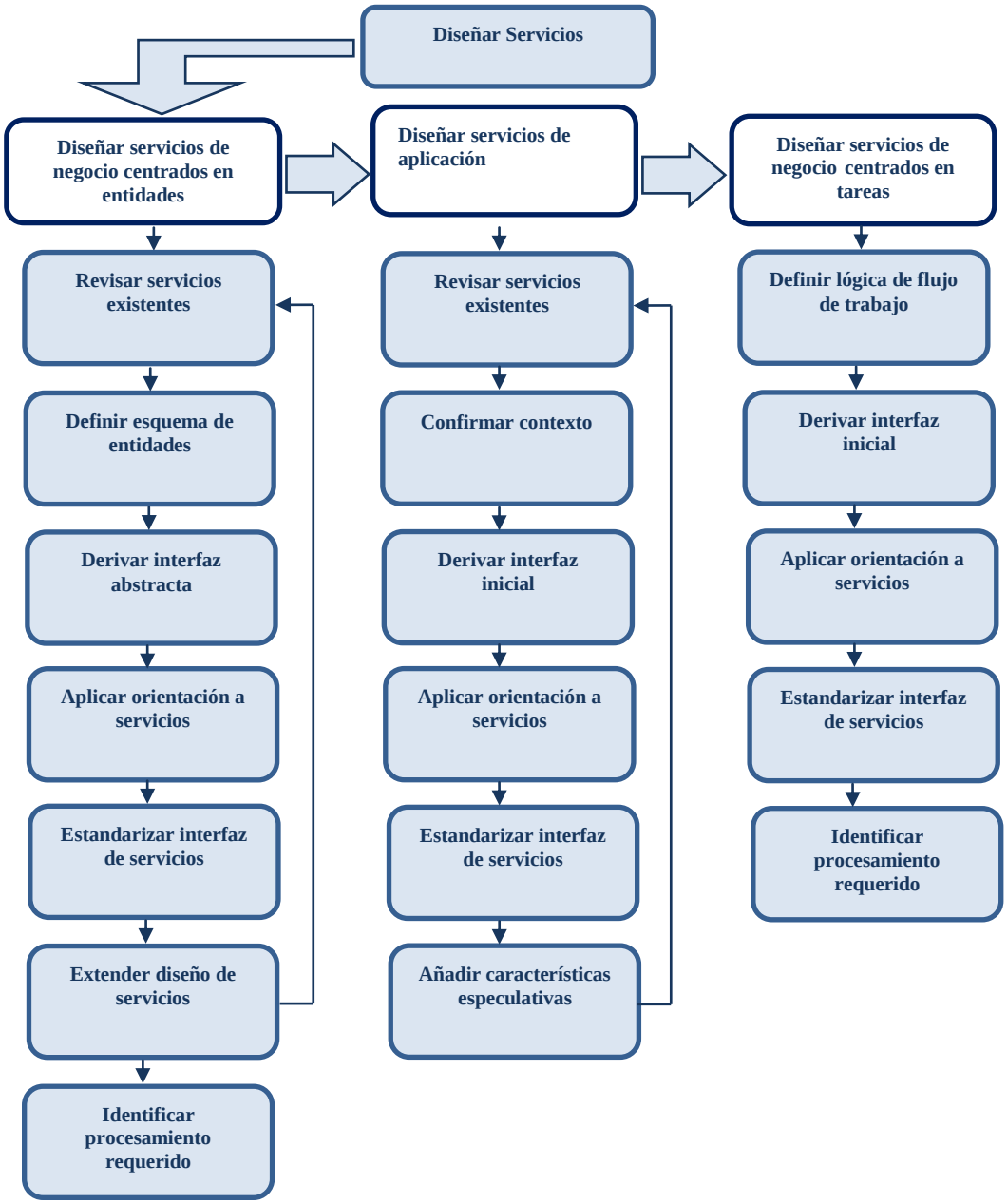

Figura 25: Proceso para diseñar servicios según Thomas Earl

Anexo 4: Descripción de los artefactos de entrada al Proceso de Diseño de Servicios.

Modelos del Negocio: Constituyen un conjunto de artefactos que describen los elementos fundamentales que sirven de base para conocer la organización y diseñar la arquitectura de servicios más apropiada para ella. Entre los modelos de negocio se puede encontrar:

- Mapas de procesos de la organización.
- Fichas de procesos.
- Inventario de sistemas legados y recursos de información como bases de datos, servidores de información, etc.
- Modelo de dominios de la organización.
- Matrices de procesos vs servicios de software identificados.
- Matrices de procesos vs entidades del modelo canónico.
- Entre otros.

Arquitectura Global de Servicios: Ofrece una visión general de la arquitectura de servicios en la organización. Contiene varias vistas de la arquitectura de servicios incluyendo:

- Aspectos del negocio relevantes para la arquitectura.
- Vista de Especificación de Servicios.
- Vista de Implementación.
- Vista de Despliegue.

Incluye además:

- Agrupaciones de los servicios por dominios y capas de la arquitectura
- Expectativas de calidad (rendimiento, seguridad, etc.) de los servicios en general o por dominio de servicios.

Define todos los servicios identificados hasta el momento en la organización, estructurados en capas de la arquitectura y que deberán realizarse paulatinamente a través de diversos proyectos.

Plan de Proyecto de Servicios: Define cuales son los servicios que forman parte del proyecto que se describe y en qué orden se realizarán. Establece además un cronograma para el desarrollo del proyecto entre otros elementos de gestión.

Descripción Inicial de Servicios: Contiene los elementos primarios de los servicios identificados: nombre, descripción, operaciones, descripción de operaciones y entidades del *Modelo de Datos Canónico* a las cuales están asociadas las operaciones, reglas de negocio que debe tener en cuenta el servicio.

Modelo de Datos Canónico: Es un modelo de datos lógico (entidades con sus atributos, detalles de estas entidades, relaciones entre ellas, cardinalidad, etc.) que representa la información relevante del negocio y la estructura con que los sistemas informáticos, por acuerdo general, deben intercambiar esta información. Puede contener además restricciones acerca de los datos.

Políticas de Diseño: Son restricciones y normativas que rigen la actividad de diseño establecidas por arquitectos, diseñadores que responden a la necesidad de lograr cierto nivel de estandarización, potenciar determinado atributos de calidad u otros requerimientos de la organización a la cual va dirigido el diseño. Pueden ser de tipo:

- Convenciones de nombres.
- Granularidad de los servicios.
- Clasificación de los servicios.
- Estándares para el diseño de servicios a usar por defecto.
- Entre muchas otras.

Anexo 5: Plantilla del Artefacto Especificación de Servicio.

Especificación de Servicio

1. Propiedades del Servicio

1.1 Propiedades del servicio de cara al negocio

Nombre del Servicio	*Nombre que la empresa usará para el servicio*
Alias	*Otros nombres para el servicio, lo que podría ser utilizada por la búsqueda de este servicio.*
Propósito del Servicio	*En palabras que describa el alcance y los beneficios del Servicio.*
Dominio del Negocio	*El Dominio de Negocio al que el Servicio pertenece. Deje en blanco si no está asignado a ningún Dominio de Negocio.*
Dueño del Negocio	*Persona que aprueba este servicio, así como los cambios que se realicen.*
Consumidores Potenciales	*Organizaciones y/o desarrolladores para los cuales el servicio está destinado.*
Apoyo al Proceso de Negocio	*Lista de Procesos de Negocio soportados por alguna versión de este servicio.* *Distinguir entre soporte planeado/espera y soporte actual alcanzado por una versión actual de este Servicio.*
Apoyo a los Objetivos del Negocio	*Lista de objetivos o estrategias de negocio formalmente definidas que el Servicio soportará.*
Estabilidad (en los próximos años)	*Necesidad de cambios previstos dentro de los próximos años y/o tiempo de vida, en caso que sea un servicio temporal.*
Factores críticos de éxito	*Identificar uno o más criterios claves que pueden ser usados para determinar si este servicio, una vez en producción, representara un éxito inversionista.*
Otra Información	*Alguna otra información de interés.*

1.2 Propiedades Técnicas

Nombre del Servicio (Técnico)	*Nombre usado en la definición tecnológica (ej. WSDL nombre para un Servicio Web).*
Dominio del Servicio	*El dominio de Servicio al cual este servicio ha sido asignado.*
Propietario Técnico	*Persona responsable de la provisión de este servicio.*
Arquitectura de Capa	*Proceso | Capacidad | Servicios Centrales | Comunicación con Aplicaciones | Utilidad |*

	Infraestructura
Responsabilidad por tipos de Negocio	*Lista de tipos o entidades del negocio para los cuales este servicio toma al responsabilidad principal. Esto es principalmente aplicado a los servicios de entidad.*
Otras responsabilidades claves	*Alguna responsabilidad no especificada en la entrada propósito del Servicio.*
Operaciones	*Una lista de operaciones sin parámetros.*
Dependencias especificadas en otros servicios (dependencia de mensajes)	*Lista de servicios "requeridos" de los cuales este servicio está obligado a depender consumiendo sus resultados o salidas.*
Otras dependencias especificadas (dependencia de integridad)	*Otras dependencias especificadas ej. Dependencias de integridad.*
Fuentes / Organización proveedora	*Como el servicio fue o será aprovisionado.* *Nombre del suministrador externo, si fue aprovisionado o desarrollado por una tercera persona.*
Otras Clasificaciones	*Clasificadores posteriores que pueden ser útiles al buscar un servicio.*
Otra Información	*Alguna otra información que el autor necesite comunicar a los desarrolladores o consumidores.*

2. Calidad del Servicio

Grupo de Operaciones:		
DESEMPEÑO	Expectativas y cualquier desviación.	Justificación de la desviación.
Disponibilidad	Oro | Plata | Bronce	
Fiabilidad	Oro | Plata | Bronce	
Rendimiento	Oro | Plata | Bronce	
Tiempo de Respuesta	Oro | Plata | Bronce	
SEGURIDAD	Expectativas y cualquier	Justificación de la

	desviación.	desviación.
Autenticación	Rojo | Ámbar | Amarillo	
Integridad del Mensaje	Rojo | Ámbar | Amarillo	
Privacidad	Rojo | Ámbar | Amarillo	
Autorización	Rojo | Ámbar | Amarillo	
No Repudio	Rojo | Ámbar | Amarillo	

Acuerdo de Nivel de Servicio	*Insertar referencia.*

3. Especificación de Operaciones

3.1 <Nombre de la Operación> del <Nombre del Servicio>

Firma	*Copiar o añadir referencia a la sección Firma de Operaciones*
Propósito:	*Copiar o añadir referencia a la sección Propiedades del Servicio de Cara al Negocio*
¿Actualización?	*Si/No (Si, si cambia el estado del Modelo de Información)*
¿Transacción Atómica?	*Si/No (Si, si es actualizado y garantiza todas las actualizaciones o ninguna de ellas)*
¿Puede participar en transacciones?	*Si/No (Si, si es actualizado y puede participar en una gran unidad de trabajo)*
¿Necesaria transacción coordinada?	*Si/No (Si, si es actualizado y es especificado para invocar muchas transacciones atómicas)*
Operación de coordinación alcanzada por:	*Solo responda esta sección si es necesaria una transacción coordinada* *Dos Fases | Tres Fases | Compensación |<nombre del estándar>*
Transmisión	*Un sentido, solicitud – respuesta, notificación, etc.*
Estilo	*RPC | Documento*

3.2 Precondiciones y Postcondiciones de la Operación

Nombre del par 1	
Precondición 1	
Postcondición 1	
Nombre del par 2	
Precondición 2	
Postcondición 2	
Excepción	Mensaje retornado

4. Secuencias de Operaciones Obligatorias

<Nombre de la Secuencia 1>

Introducción

Diagrama *(o ruta del diagrama)*

Notas Aclaratorias

5. Orquestación de Operaciones

Cuando existen dependencias entre servicios, esta relación se modela mediante un diagrama de interacción (secuencia o actividad) para la operación que es dependiente.

Diagrama *(o ruta del diagrama)*

Notas Aclaratorias

6. Cumplimiento de las normas y estándares

Esta sección define los estándares que el servicio debe seguir o son utilizados.

Cumple con los estándares definidos por defecto para:	*Empresa | <Nombre del Dominio de Servicios> | "Ninguno de estos"*		
Nombre del Estándar o Convención	Versión / Entrega	Desviación	Justificación
Ej.: convención de nombres, convención de la numeración de las versiones, procedimientos de aprobación.			

Técnicos, ej. protocolos específicos para los Servicios Web.			
Negocio ej. estándares generales para el uso de la empresa, o por el proveedor de software.			
Reguladores ej. regulados por el gobierno o por la industria.			

7. Modelo de Información

7.1 Diagrama del Modelo de información *(o ruta del diagrama)*

Notas Aclaratorias

7.2 Modelo de Información: Definición de los Tipos de Información

En la columna de mapeo escriba el nombre del tipo definido en el Modelo de Datos Canónico del cual se deriva el tipo de información, si existiese. Explique alguna diferencia.

Nombre del Tipo de Información	Ejemplos de definición de tipos de información	Mapeo desde el Modelo de Datos Canónico

Ruta del Modelo de Tipos de Negocio *(ruta del modelo o diagrama)*

7.3 Invariantes

Dueño del Tipo de Información	Atributo aplicable o Asociación final	Expresión invariante

8. Firmas de Operaciones

Localización del WSDL *(ruta)*

Nombre de la Operación +propósito	Estilo	Entradas	Salidas	Fallos

9. Definición de mensajes

Nombre corto para el esquema XML (para su uso en esta especificación)	Breve descripción del contenido del mensaje	Localización del esquema XML. Ruta y nombre de archivo

10. Instrucciones para la Implementación

11. Instrucciones para el Despliegue

Anexo 6: Plantilla del Artefacto Perfil de Servicio.

Perfil de Servicio

Nombre del Servicio:

Alias:

Propósito:

Dirección del WSDL:

Operación	Descripción	Entradas	Salidas	Protocolo de Conexión	Atributos de calidad

Secuencias de Operaciones Obligatorias

En caso de que un grupo de operaciones necesiten ser invocadas en un determinado orden se representa la secuencia mediante un diagrama y notas aclaratorias. Las operaciones pueden ser de propiedad exclusiva de este servicio o pueden trabajar en conjunto con otras operaciones definidas en otro servicio.

<Nombre de la Secuencia 1>

Diagrama

Notas Aclaratorias

Anexo 7: Plantilla del Artefacto Solicitud de Cambios.

Solicitud de Cambios

Identificador del grupo de cambios:

Fecha de emitido:

Solicitantes:

< Cambio1>

- Descripción de cambio:
- Artefactos afectados:
- Personas afectadas:
- Responsable de ejecución:
- Fecha límite:
- Notificar cumplimiento a:

< Cambio2>

- Descripción de cambio:
- Artefactos afectados:
- Personas afectadas:
- Responsable de ejecución:
- Fecha límite:
- Notificar cumplimiento a:

Aprobado por:

Anexo 8: Plantilla del Artefacto Variantes de Provisión de los Servicios.

Variantes de Provisión de los Servicios

Servicio:

Descripción:

Operaciones:

	<Operación 1>		<Operación 2>	
Parámetros	**< Variante de Provisión 1>**	**< Variante de Provisión 2>**	**< Variante de Provisión 1>**	**< Variante de Provisión 2>**
Enfoque (Comprar, subcontratar, Desarrollar)				
Proveedor				
Costo estimado				
Riesgos				
Fortalezas				
Debilidades				
Tiempo estimado para obtener el servicio				
Otras consideraciones				

Selección

	<Operación 1>		<Operación 2>	
	< Variante de Provisión 1>	**< Variante de Provisión 2>**	**< Variante de Provisión 1>**	**< Variante de Provisión 2>**
Candidata				
Seleccionada				

Anexo 9: Herramientas propuestas para el Proceso de Diseño de Servicios.

Herramienta	Justificación de la Selección
Enterprise Architect (EA)	Constituye una de las herramientas CASE más potentes • Satisface casi todas las necesidades de modelado para el todas las fases de los proyectos SOA (modelado negocio, arquitectura, diseño de servicios, componentes de implementación y despliegue). • Provee facilidades altamente configurables para generación de reportes. • Permite la generación automática de código XML (WSDL, XSD, etc.) indispensable en el diseño de servicios. • Alta flexibilidad, disponibilidad de ayuda, usabilidad, frecuente actualización, etc. • Tiene la desventaja de ser propietaria pero su licencia fue adquirida por la Universidad.
Plug-in SOAPUI de Eclipse	Con una interfaz amigable permite el chequeo de interoperabilidad de los servicios. Es altamente configurable.
XMLSpy	• Potente herramienta para el trabajo con archivos XML permite el modelado visual de esquemas XML y generación de código. • Permite chequear que los archivos XML son válidos y bien formados. • Tiene la desventaja de ser propietaria. • No existen herramientas libres comparables a su gama de funcionalidades y robustez.

Anexo 10: Roles del Proceso de Diseño de Servicios.

Rol	Descripción	Habilidades y Conocimientos
Diseñador de Servicios	Este rol dirige el diseño de un grupo de servicios, dentro de las restricciones de los requisitos de negocio, arquitectura y proceso de desarrollo para el proyecto. Identifica y define las responsabilidades y operaciones definitivas de los servicios y aplica estándares y políticas los servicios. El diseñador se asegura de que el diseño sea coherente con lo establecido por la arquitectura de servicios, y que este esté detallado hasta un punto en que pueda proceder la implementación.	**Conocimientos:** -Patrones de Diseño SOA -Principios de Orientación a servicios -Estándares de Servicios Web -Conocimiento de la tecnología y la infraestructura sobre la que se diseñará **Habilidades:** -Dominio de las herramientas y lenguaje de modelado de servicios -Conocimientos básicos de implementación de servicios.
Arquitecto de Servicios	Dirige las principales decisiones técnicas, de la arquitectura orientada a servicios. Esto incluye la identificación y la documentación de los aspectos arquitectónicamente significativos para el desarrollo de los servicios. El arquitecto también es responsable de	**Conocimientos:** -Amplia gama de infraestructura y tecnologías para desarrollar los servicios. - Principios de orientación a servicios. - Patrones de Diseño SOA. -Estándares de Servicios Web. -Conocimiento de la

	proporcionar el fundamento de estas decisiones, equilibrando las preocupaciones de los diferentes interesados, reduciendo los riesgos técnicos, y garantizando que las decisiones se comunican, y validan con eficacia, y que se acatan.	tecnología y la infraestructura sobre la que se diseñará. **Habilidades:** -Dominio de las herramientas y lenguaje de modelado de servicios. -Saber implementar completamente servicios sencillos.
Arquitecto de Seguridad	Este rol dirige el desarrollo de la seguridad de la arquitectura orientada a servicios, que incluye la selección de patrones de seguridad para el diseño del sistema, la documentación de políticas técnicas relativas a la implementación de la seguridad y el soporte de decisiones técnicas clave que limiten la implementación de seguridad global para el proyecto.	**Conocimientos:** - Conceptos medios o avanzados de seguridad. -Patrones de seguridad. -Estándares, protocolos y tecnologías asociados a seguridad. **Habilidades:** -Dominio de herramientas asociadas a seguridad.

Anexo 11 . Proceso a automatizar en el proyecto piloto.

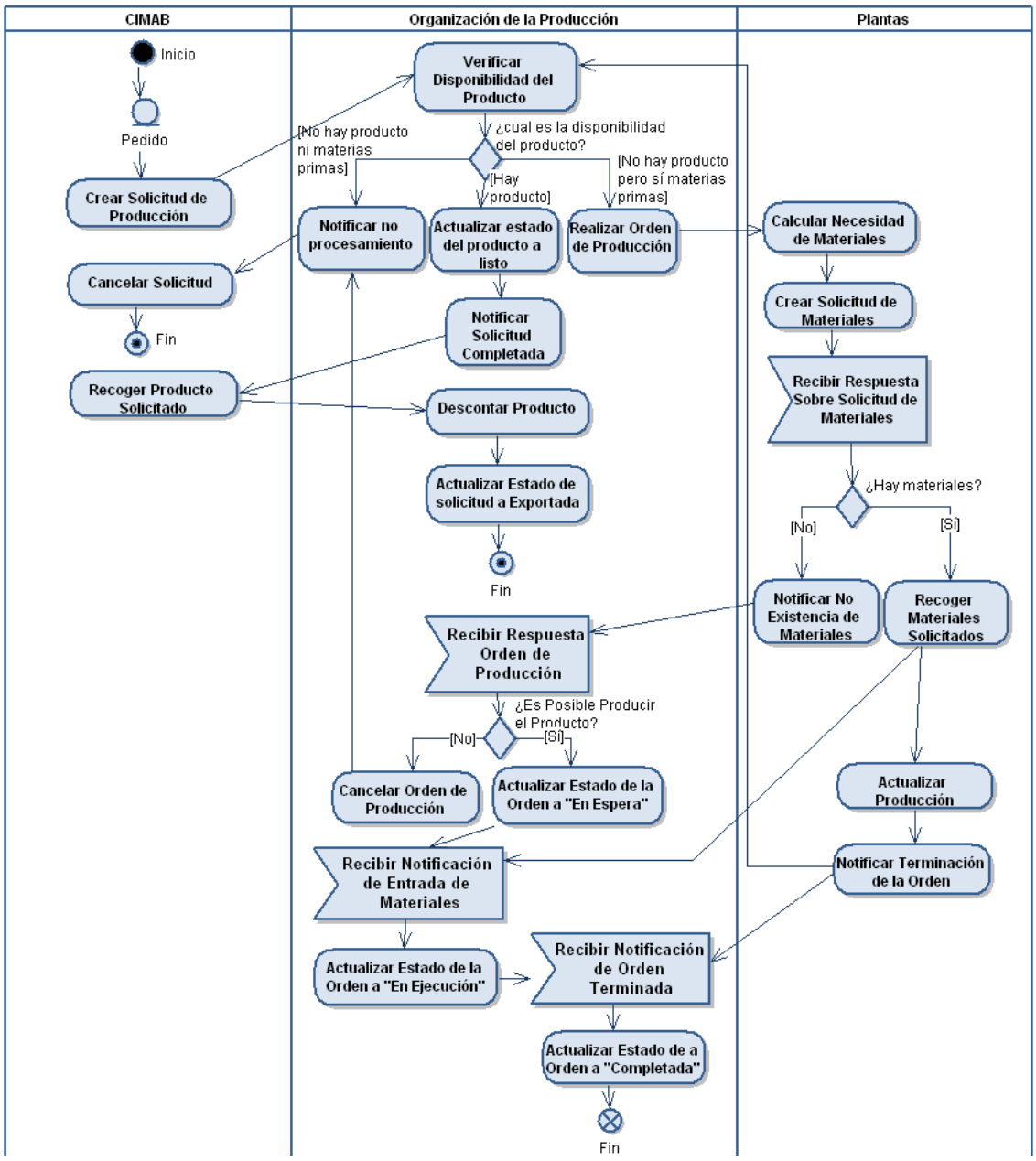

Figura 26. Proceso *Gestión de Ordenes Comerciales* del CIM a automatizar en el proyecto piloto.

Anexo 12: Arquitectura de servicios inicial para el proyecto piloto.

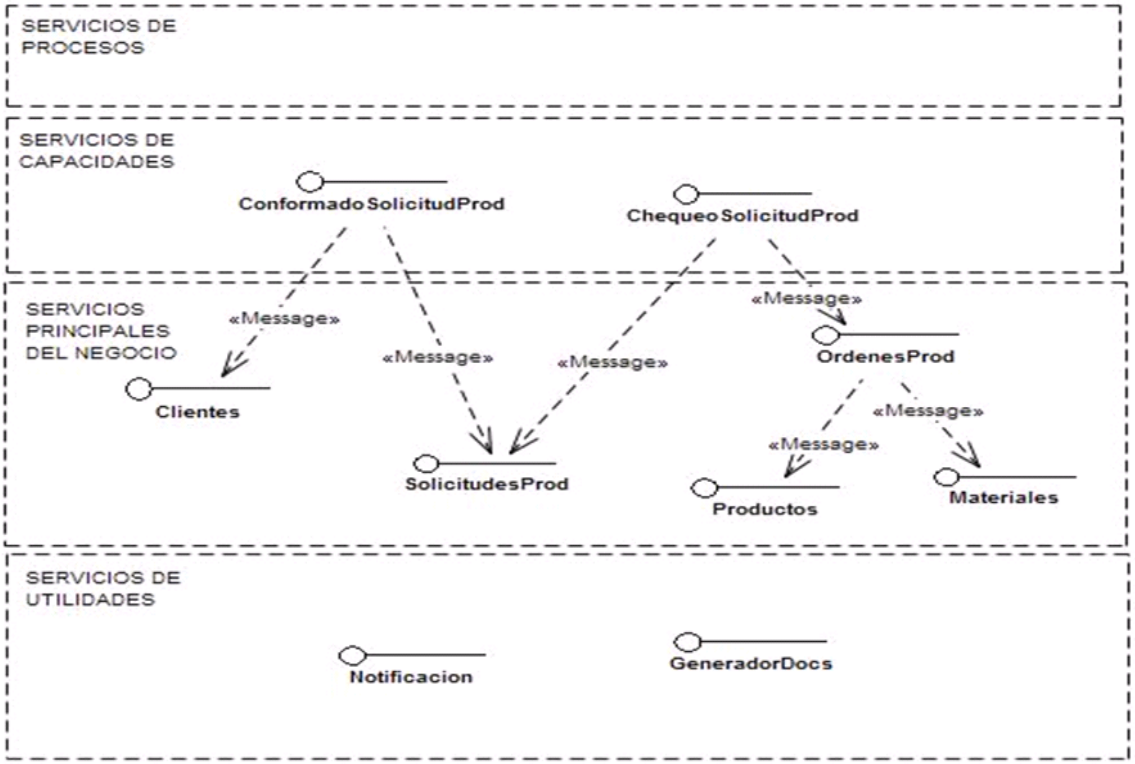

Figura 27. Arquitectura de servicios inicial para el proyecto piloto.

Anexo 13: Lista de Chequeo para medir la autonomía de los servicios diseñados.

No	Indicador	Pregunta	Nivel de Autonomía	
1	Solapamiento de funciones	¿Tiene el servicio funciones solapadas con otros servicios?	Alto	No tiene funciones solapadas con otros servicios.
			Bajo	Tiene alguna función solapada con otro servicio.
2	Autonomía del Ciclo de vida del servicio	¿Puede el servicio ser desplegado, modificado y mantenido independientemente de otros servicios?	Alto	Si, el servicio no depende de ningún otro servicio o componente para esto.
			Medio	El servicio depende de un componente o servicio para ejecutar sus funcionalidades.
			Bajo	El servicio depende de dos o más servicios o componentes, para ejecutar sus funcionalidades y para ser actualizado.

Anexo 14: Encuesta realizada al equipo de desarrollo del proyecto piloto para valorar la calidad de la especificación de servicios realizada a partir del proceso de diseño propuesto.

Encuesta para valorar la calidad de las especificaciones de servicios realizadas en el proyecto Gestión de Operaciones para el Centro de Inmunología Molecular (CIM)

De acuerdo a las especificaciones de servicios realizadas para el proyecto Gestión de Operaciones para el Centro de Inmunología Molecular (CIM) responda las siguientes interrogantes:

Completitud

1. ¿En qué medida considera que las especificaciones de servicios realizadas recogen la información necesaria para implementar dichos servicios?
   ___Alta ___Media ___Baja

2. ¿En qué medida considera que las especificaciones de servicio realizadas aportan la información necesaria para consumir dichos servicios?
   ___Alta ___Media ___Baja

Comprensión

3. ¿En qué medida considera que las especificaciones de servicios son claras y no ambiguas?
   ___Alta ___Media ___Baja
4. ¿En qué medida considera que las especificaciones de servicios realizadas son entendibles sin tener que consultar a los analistas u otra documentación?
   ___Alta ___Media ___Baja

Anexo 15: Relación del Principio de Capacidad de Composición con el resto de los principios de orientación a servicios.

Principio de orientación a servicios	Relación del Principio de Capacidad de Composición	Nivel de importancia
Contrato Estandarizado de los Servicios	Aumenta la estandarización de los servicios que participan en las composiciones reduciendo la necesidad de transformaciones de mensajes. Una creciente estandarización tiende a decrecer la complejidad de las composiciones.	Alta
Bajo Acoplamiento de los Servicios	Reducir las dependencias del servicio le permite a este formar parte de un mayor número de composiciones, lo cual provee más opciones en tiempo de diseño.	Media
Abstracción de los Servicios	Un mayor ocultamiento de información reduce el conocimiento de lo que hay por debajo del controlador de la composición, esto puede simplificar las composiciones de servicios y aumentar la confianza en los subcontroladores de dichas composiciones.	Media
Reusabilidad de los Servicios	Mientras más reusables sea la lógica que provee el servicio, más oportunidades tiene este de participar en composiciones.	Alta
Autonomía de los Servicios	Reduce el solapamiento funcional entre servicios lo cual potencia la calidad del diseño de las composiciones estableciendo la normalización entre los miembros de dichas composiciones.	Media
Ausencia de estado en los Servicios	La complejidad de las composiciones puede aumentar si algunos servicios requieren facilidades diferentes de de gestión de estados respecto a otros.	Baja

Facilidad de Descubrimiento de los Servicios	Con un mayor conocimiento de candidatos potenciales para las composiciones se podrán seleccionar los miembros más apropiados. La facilidad de descubrimiento puede beneficiar al proceso de diseño de las composiciones.	Alta

Anexo 16: Lista de chequeo para la verificar el cumplimiento del Principio de Capacidad de Composición en base al cumplimiento del resto de los principios de orientación a servicios.

		Servicios →		
		Operaciones →		
Principio	**No Preg.**	**Pregunta ↓**		
Contrato Estandarizado	1	¿Los mensajes de entrada y salida requeridos por esta operación están estandarizados con aquellos usados por otros servicios (que usan los mismos datos) dentro del mismo inventario?		
Bajo Acoplamiento	2	¿Evita esta operación altos niveles de acoplamiento a la implementación, la tecnología y a funcionalidades definidas para otros servicios?		
Abstracción	3	¿Se provee una clara expresión de los atributos de calidad relevante para este servicio en los metadatos asociados al servicio?		
Reutilización	4	¿Fue diseñada esta operación para una alta reutilización?		
Autonomía	5	¿Posee esta operación un nivel razonable de autonomía?		
Sin Estado	6	¿Esta operación utiliza gestión de estados de manera consistente con el resto de los servicios definidos en el inventario?		
Descubrimiento	7	¿Está el servicio equipado (respecto a esta operación) con todos los metadatos necesarios para ser fácilmente localizado y entendido por los diseñadores de composiciones?		

Las preguntas se responden de acuerdo a los siguientes valores cualitativos:

- **Sí:** Significa que se cumple el principio al que tributa la pregunta en cuestión, lo cual influye positivamente en el cumplimiento del principio de capacidad de composición.

- **No:** Significa que no se cumple el principio al que tributa la pregunta en cuestión, lo cual influye negativamente en el cumplimiento del principio de capacidad de composición.
- **En alguna medida:** Significa que el principio al que tributa la pregunta en cuestión se cumple parcialmente lo cual influye en el cumplimiento del principio de capacidad de composición.