



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática
Escuela Académico Profesional de Ingeniería de Sistemas

**Método propuesto para el desarrollo aplicaciones SOA
(service oriented architecture), caso Telefónica**

TESINA

Para optar el Título Profesional de Ingeniero de Sistemas

AUTOR

Javier MENDOZA SILVA

ASESOR

Carlos Alberto CÁNEPA PÉREZ

Lima, Perú

2008



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Mendoza J. (2008). *Método propuesto para el desarrollo aplicaciones SOA (service oriented architecture), caso Telefónica*. Tesina para optar el título profesional de Ingeniero de Sistemas. Escuela Académico Profesional de Ingeniería de Sistemas, Facultad de Ingeniería de Sistemas e Informática, Universidad Nacional Mayor de San Marcos, Lima, Perú.

Este trabajo está dedicado a mi padre y a mi madre Sixto y María que con su esfuerzo y apoyo han logrado que siga siempre adelante.

AGRADECIMIENTOS

A Dios, el ser supremo que siempre está conmigo.

A mis padres que siempre me han apoyado, la fortaleza que me brindan en las cosas difíciles, me aconsejan para ser mejor cada día.

A mis hermanos que siempre están pendientes de mí.

Al Licenciado Carlos Canepa, por su orientación y dedicación para que este trabajo cumpla con los objetivos trazados.

A todas aquellas personas que indirectamente me ayudaron para culminar este trabajo y que muchas veces constituyen un invalorable apoyo.

Método propuesto para el desarrollo aplicaciones SOA (Service Oriented Architecture), caso Telefónica

RESUMEN

En los últimos años los sistemas de información de una organización ya no solo deben de poder comunicarse entre ellos mismos ahora además deben de poder comunicarse con sistema de otras organizaciones, no solo por tener la información integrada sino más bien para tener al proceso de negocio de una organización funcionando como un todo, como es el caso del proceso de negocio de Comercio Exterior de Telefónica Gestión de Servicios Compartidos S.A.C.

El Sistema de Comercio Exterior – COMEX, es una propuesta como parte de este esfuerzo, el sistema dirigido por la Dirección de Logística de Telefónica Gestión de Servicios Compartidos S.A.C. (tgestiona) será el encargado de controlar y almacenar toda la información relacionada con las importaciones. El sistema debe permitir facilitar la gestión de seguimiento de los embarques de importación usando la tecnología de servicios que sigue los principios de SOA (Service-Oriented Architecture).

Service Oriented Architecture (SOA) es un estilo de Arquitectura de Software basado en la definición de servicios reutilizables, con interfaces públicas bien definidas, donde los proveedores y consumidores de servicios interactúan en forma desacoplada para realizar los procesos de negocio. Los servicios representan grupos lógicos de operaciones relacionadas con algún concepto del negocio, y los procesos del negocio se realizan

mediante secuencias definidas de invocaciones a servicios, en orquestación o coreografías de servicios.

La tesina abarca los problemas de integración y comunicación entre las empresas y el sistema COMEX. La principal contribución al trabajo es proporcionar un método para el desarrollo de una Arquitectura de Servicios basado en SOA.

El método fue integrado en el proceso base del RUP para luego ser integrado a la metodología de trabajo de la empresa Telefónica Gestión de Servicios Compartidos S.A.C. Esta extensión se denomina Extensión del proceso RUP Adaptado para desarrollo SOA, incluye la definición entre otros de Disciplinas, Actividades, Entregables y Roles, adecuados para guiar un desarrollo SOA cumpliendo con las características que define este enfoque.

El método propuesto está actualmente en prueba en el marco de estandarizar los procesos de desarrollo en la Dirección de Tecnología y Sistema de Información de tgestiona, para la construcción de una aplicación de Comercio Exterior, perteneciente a la Dirección de Logística.

Palabras Claves: Arquitectura de Software, Procesos y Metodologías de desarrollo Service Oriented Architecture (SOA), Procesos de Negocio, Servicio, Orquestación, Coreografía.

Proposed Method for developing SOA (Service Oriented Architecture) Applications - Telefonica Case

ABSTRACT

In recent years organization information systems not only must be able to communicate among themselves, they also must be able to communicate with other organizations systems, not only to have integrated information but for running organization business processes as a whole, that's the case of Foreign Trade business processes of Telefonica Gestion de Servicios Compartidos S.A.C.

Foreign Trade system – COMEX, is a proposal, part of the efforts lead by Foreign Trade Supervision of Telefonica Gestion de Servicios Compartidos S.A.C. which will be in charge of controlling and storing all information related to importation goods. The system must facilitate tracking management of import ships by implementing service oriented technologies based on SOA principles (Service-Oriented-Architecture)

Service Oriented Architecture (SOA) is a Software Architecture based in reusable services definition with very well specified public interfaces where providers and consumers interact in a decoupled way to perform business processes. Services represent logical groups of operations related to some business concept, business processes are performed through defined sequence of service invocations in orchestration or choreography of services.

Thesis comprises integration and communications problems between external companies and the COMEX system. The main contribution in this paper is to propose a method for developing a Service Oriented Architecture based on SOA.

The method has been integrated in the RUP base process to later be integrated to the working methodology of Telefonica Gestión de Servicios Compartidos S.A.C company. This extension is called RUP Extension Adapted for SOAP Development which includes the definition among others of Discipline, Activities, Deliverables and Roles, adapted to guide a SOA development fulfilling with the characteristics that defines this approach.

The method proposed is currently being tested in the context of standardize business processes in the T-Gestiona Technology and Information System Office for building the Foreign Trade application that belongs to the Logistics Office

Key Words: Software Architecture, Processes and Methodologies of Development of Service Oriented Architecture (SOA), Business Processes, Services, Orchestration, Choreography.

ÍNDICE

Lista de Figuras	xi
Lista de Tablas	xii

CAPÍTULO 1: INTRODUCCIÓN..... 1

1.1	Antecedentes.....	2
1.2	Definición del problema	4
1.3	Objetivos.....	4
1.3.1	Objetivos General.....	4
1.3.2	Objetivos Específicos	4
1.4	Justificación	5
1.5	Propuesta	5
1.6	Organización de la Tesina	6

CAPÍTULO 2: MARCO TEÓRICO CONCEPTUAL..... 7

2.1	Antecedentes de la investigación.....	7
2.2	Bases Teóricas	10
2.2.1	Procesos y metodologías de desarrollo de software	10
2.2.1.1	Ciclo de Vida del Software.....	10
2.2.1.2	Procesos de desarrollo de Software	11
2.2.2	Metodología para el desarrollo de proyectos.....	15
2.2.2.1	Marco Estratégico de Gestión Orientada al Negocio (MEGON).....	15
2.2.3	Presentación de la Arquitectura Orientada a Servicios	18
2.2.4	SOA (Service Oriented Architecture) Arquitectura Orientada a Servicios	19
2.2.4.1	Contexto de SOA.....	19
2.2.4.4	Conceptos de SOA.....	22
2.2.4.5	Elementos de SOA.....	23
2.2.4.8	Clasificación y relaciones entre servicios	27
2.2.4.15	Procesos de negocio.....	32
2.2.4.16	Orquestación de Servicios	33
2.2.4.17	Coreografía de servicios	33
2.2.4.18	Niveles de realización de SOA	34
2.2.4.19	Aspectos técnicos de la realización de SOA.....	35
2.2.5	Business Process Modeling (BPM).....	39
2.2.5.1	Presentación de BPM.....	40
2.2.5.2	Modelado de Procesos del Negocio con BPM.....	40
2.2.5.3	Business Process Modeling Notation (BPMN).....	41
2.2.5.5	Business Process Modeling Language (BPML)	42
2.2.5.7	Business Process Management Systems (BPMS).....	43
2.2.6	SOA y BPM juntos.....	44

CAPÍTULO 3: ENFOQUES PARA EL DESARROLLO DE SOA 48

3.1	El enfoque SOA basado en patrones para e-business.....	48
3.2	Enfoque SOA basado en mejores Prácticas de Krafzig Banke, Slama.....	50
3.3	Enfoque SOA de Thomas Erl.....	52
3.4	Caso de Existo de aplicaciones SOA.....	57

CAPÍTULO 4: RESOLUCIÓN DEL PROBLEMA APLICANDO LA TÉCNICA SELECCIONADA.....	61
4.1 Evaluación y selección de la técnica.....	61
4.1.1 Criterios de evaluación:.....	61
4.1.2 Selección de la técnica usada:.....	62
4.2 Adaptación o aplicación de herramienta teórica para resolver el problema	64
4.2.1 Contexto de la propuesta	64
4.2.2 Proceso base del RUP Adaptado.	66
4.2.3 Extensión del proceso RUP Adaptado para desarrollo SOA.....	68
4.2.3.3 Disciplinas para el proceso de adopción de SOA	69
CAPÍTULO 5: DESCRIPCIÓN DE LA SOLUCIÓN	93
5.1 Contexto del caso Telefónica Gestión de Servicios Compartidos S.A.C.	93
5.2 Principales Requerimientos del Sistema.....	93
5.3 Desarrollo y seguimiento de la prueba	94
5.4 Revisión de entregables	95
5.5 Resultados por fase e iteración	96
5.6 Resultados por Disciplina, actividad y entregable.....	97
CAPÍTULO 6: CONCLUSIONES Y TRABAJOS FUTUROS	104
6.1 Conclusiones sobre el trabajo	104
6.2 Trabajos Futuros	104
6.3 Recomendaciones	105
CAPÍTULO 7: REFERENCIAS BIBLIOGRÁFICAS	107
CAPÍTULO 8: ANEXOS	109
Anexo A: Plantillas de entregables de la Extensión SOA	109
8.1 Plantillas de la Disciplina Modelado del Negocio.....	109
8.2 Plantillas de la Disciplina Diseño Extensión SOA	116
Anexo B: Guía para la clasificación de servicios en SOA	121

Índice de figuras

Figura 1: Dimensiones de modelado del RUP de [RUP08]	13
Figura 2: Hitos y Entregables Megon (resumen) [Cast08]	17
Figura 3: Evolución de Arquitecturas de Software hasta SOA de [Endr04]	18
Figura 4: Dos aspectos claves de las Organizaciones [Erlt05]	20
Figura 5: Capa de servicios entre los procesos del Negocio y las tecnologías de [Erlt05]	21
Figura 6: Granularidad de servicios en SOA de [Erlt05]	25
Figura 7: Paradigma “descubrir, ligar e invocar” [Stev04]	26
Figura 8: Clasificación de servicio Technology gateways.	28
Figura 9: Clasificación de servicio Adapters.	28
Figura 10: Clasificación de servicio Facade.	29
Figura 11: Capas de servicios según la clasificación de [Kraf05]	30
Figura 12: Capas de servicios según la clasificación de [Erlt05]	32
Figura 13: Marco de referencia de implementación con Web Services de [Erlt05]	38
Figura 14: Ejemplo básico de Business Process Diagram (BPD) en BPMN de [BPMN06]	41
Figura 15: Correspondencia uno a uno entre BPMN y BPML de [Shfp03]	43
Figura 16: SOA provee la infraestructura para BPM adaptada de [Kraf05] [Endr04]	45
Figura 17: Pasos del enfoque SOA propuesto en [Endr04]	50
Figura 18: Fases del proceso de desarrollo propuesto en [Erlt05]	56
Figura 19: Bloques y fases Megon [Cast08]	65
Figura 20: Bloques y fases Megon adaptado para SOA	69
Figura 21: Diagrama de la actividad Evaluar la Organización Objetivo	71
Figura 22: Diagrama de la actividad Identificar Procesos de Negocio	72
Figura 23: Diagrama de la actividad Identificar Servicios	75
Figura 24: Diagrama de la actividad Especificación de Servicios	76
Figura 25: Diagrama de la actividad Servicio Existentes	78
Figura 26: Diagrama de la actividad Asignar servicios a componentes	79
Figura 27: Diagrama de la actividad Orquestación de Servicios	81
Figura 28: Diagrama de la actividad Implementar servicios	83
Figura 29: Diagrama de la actividad Planificar la Integración de la iteración	85
Figura 30: Diagrama de la actividad Integrar Sistema	87
Figura 31: Diagrama de la actividad Pruebas internas	88
Figura 32: Diagrama de la actividad Planificar Pruebas	90
Figura 33: Diagrama de la actividad Especificar Casos de Prueba	91
Figura 34: Diagrama de Casos de Uso del Negocio en la Prueba SOA	98

Índice de tablas

Tabla 1: Comparación entre BPM y SOA	47
Tabla 2: Cuadro comparativo de enfoques SOA	63
Tabla 3: Cuadro de roles en RUP y Megon (tgestionaria)	67
Tabla 4: Planificación de Entregables	95
Tabla 5: Duración prevista y real de las Fases e iteraciones de la prueba	96
Tabla 6: Cantidad versiones y entregables principales en cada Fase e iteración	97
Tabla 7: Ejemplo de descripción del Caso de Uso del Negocio Registro de Requerimiento	99
Tabla 8: Descripción de los principales Casos de Uso del Sistema en la prueba SOA	101
Tabla 9: Ejemplo de Identificación y categorización de servicios en la prueba SOA	102
Tabla 10: Especificación del servicio Gestión de Orden de Compra en la prueba SOA	102
Tabla 11: Contrato funcional del método AsignarResponsableOC	103

Capítulo 1: Introducción

Toda empresa siempre requiere mejorar sus procesos en la cadena de suministro de tal manera que pueda estar siempre informado de su abastecimiento, y así poder tomar decisiones claves con anticipación.

Por ende siempre se cuenta con personal responsable de hacer el seguimiento a los procesos de suministro, sin embargo surgen algunas preguntas tales como: ¿Realmente se cuenta en cualquier momento con toda la información del proceso? ¿Sabemos con rapidez donde está actualmente un embarque? ¿Podemos obtener indicadores claves?, etc.

Las soluciones informáticas que se construyen en las empresas no resuelven de manera global las deficiencias que posee y no dan mucho valor agregado a los procesos manuales.

La implantación de nuevas tecnologías sin un marco metodológico no da frutos y por el contrario podrían ocasionar mayores pérdidas a la empresa.

En respuesta a estas incertidumbres surgen soluciones informáticas con nuevos enfoques de desarrollo de software que permiten mejorar los procesos y tener un mayor control de estos.

Ese nuevo enfoque reside en aceptar la heterogeneidad de las soluciones y las necesidades informáticas, desde esa perspectiva donde emerge la Arquitectura Orientada a Servicios (SOA, Service Oriented Architecture). Este enfoque no solo brinda una respuesta que ayuda a los sistemas a mantener su escalabilidad y flexibilidad mientras evolucionan si no también brinda la satisfacción de alcanzar un facilitador que una los cambios en tecnologías de la información y los cambios en los procesos de múltiples organizaciones.

Por ello se propone en el presente trabajo de tesis, un método basado en los diferentes enfoques que existen para el desarrollo de este nuevo enfoque llamado SOA.

1.1 Antecedentes

Telefónica Gestión de Servicios Compartidos S.A.C. (tgestiona) es una de las empresas del grupo Telefónica, cuya misión es prestar servicios integrales para satisfacer las necesidades de todos sus clientes liberándolos de gestiones ajenas a su actividad, brinda soluciones en tercerización de servicios administrativos: Recursos Humanos, Gestión Inmobiliaria, Seguridad, Recaudación, Organización y Sistemas, Gestión Logística, Contabilidad y Tesorería.

Tgestiona cuenta con una dirección de Logística quien es la encargada de brindar el servicio de gestión logística a los clientes de tgestiona, esta dirección cuenta con dos jefaturas:

- Jefatura de Planificación Aprovechamiento: encargada de la gestión de comercio exterior de mercancías desde el punto inicial del embarque hasta la llegada final del embarque (Puerto, almacén)
- Jefatura de Distribución Física: encargada de la gestión de distribución física de la mercancía desde el almacén hasta el usuario final.

Actualmente tgestiona para mejorar el proceso de gestión de la cadena de suministro subcontrata a empresas especializadas en el proceso operativo de Comercio Exterior (Operador Logístico), liberándose de esta labor y ocupándose de realizar la gestión del proceso y mantener siempre informado a sus clientes. Durante todo el proceso de Comercio Exterior se va generando información en diferentes etapas de la cadena y en diferentes fuentes, internas y externas a la empresa (Sistema SAP, Operadores Logísticos, Manual)

La Jefatura de Planificación y Aprovechamiento actualmente no cuenta con un sistema que pueda integrar la información que se genera en las diferentes fuentes durante el proceso de comercio exterior.

Tgestiona como empresa tiene la necesidad de poder contar con toda la información integrada de los embarques y con ello tener el control de los mismos, ya que existen problemas por la falta de información consolidada de las órdenes de compra, por ejemplo cuando los clientes solicitan información de sus requerimientos, este se realiza por correo o teléfono lo que origina excesivas consultas.

El desarrollo de los sistemas casi siempre están enfocados a solo resolver el problema puntual que tiene cada jefatura mas no se enfocan en brindar valor agregado en aspectos de flexibilidad, integración, reusabilidad hacia así mismo o a otras jefaturas o/y empresas. Una característica que se ha dado en la Dirección de Tecnología y Sistemas de tgestiona es que las aplicaciones han ido creciendo como islas, y cuando estas aplicaciones deben ser integradas en procesos más complejos se observa que en general lo único que parece compartir es la base de datos, requiriendo así un mayor esfuerzo para lograr que interactúen.

Un desafío que se plantea es poder intégralos para reaccionar ágilmente a los cambios en los requerimientos del negocio, principalmente en dos aspectos: procesos de negocio y tecnología disponible. La definición y disponibilidad de estos servicios para toda la empresa es la base del enfoque Service Oriented Architecture (SOA), surge entonces como forma de cambiar la visión del desarrollo de las aplicaciones en la empresa.

Una característica que generalmente acompaña el surgimiento de nuevos paradigmas y tecnologías de desarrollo, es que rápidamente surgen varias interpretaciones, implementaciones y herramientas, nuevas o adaptaciones de herramientas existentes que soportan a estos paradigmas, como es el caso de Service Oriented Architecture (SOA) que desde su aparición en los años noventa, múltiples autores han brindado sus definiciones del enfoque, sin embargo la adopción de estos nuevos enfoques por parte de una empresa sin tener claro como realizar esta adopción o como mantener el enfoque, puede llevar al fracaso de la misma. En el caso del enfoque SOA resulta importante contar con una metodología de desarrollo que permita la adopción del enfoque SOA para la empresa tgestiona, en forma disciplinada basada en la conceptualización y estandarización de las prácticas asociadas con el enfoque y no con una herramienta o interpretación particular.

La Dirección de Tecnología y Sistemas de Información de tgestiona es el área encargada de brindar soluciones de TI a las diferentes Direcciones de la empresa tgestiona y a otros clientes externos.

En el contexto presentado surge la oportunidad para realizar en el presente trabajo de tesina un método que permita desarrollar aplicaciones con enfoque SOA.

1.2 Definición del problema

Actualmente en la Dirección Tecnología y Sistemas de Información se gestionan se elaboran y utilizan diferentes sistemas de información para las distintas etapas del proceso de Comercio Exterior de la Dirección de Logística, en plataformas específicas, sin ninguna posibilidad de poder ser reutilizados y sobre todo poder integrarlos en un ambiente único. Así mismo en la elaboración de nuevos sistemas se consume demasiados recursos humanos y económicos, al final se tiene una gran diversidad de sistemas para satisfacer necesidades determinadas que cada vez necesitan intercomunicarse, lo que provoca la elaboración de nuevas interfaces de comunicación específicas (por ejemplo: Batch¹).

No existe un enfoque único a nivel empresarial para el desarrollo de software que garantice los objetivos de accesibilidad, interoperabilidad², durabilidad, integración y reutilización de los mismos.

1.3 Objetivos

1.3.1 Objetivos General

Proponer un método para el desarrollo de aplicaciones con enfoque SOA teniendo en cuenta las características que presenta el enfoque y los objetivos que persigue, en el marco de la empresa Telefónica Gestión de Servicios Compartidos S.A.C. estableciendo un enfoque disciplinado para su utilización, que sirva de guía para la obtención de dichos objetivos.

1.3.2 Objetivos Específicos

- Establecer actividades y roles que permitan cubrir los planteamientos del enfoque SOA.
- Adaptar los procesos actuales de desarrollo de software en tgestiona para la incursión en SOA.

¹ Batch: Llevar a cabo una operación particular de forma automática de un grupo de archivo.

² Interoperabilidad: Utilizar en distintas plataformas componentes creados con diferentes herramientas y desde cualquier ubicación.

- Proponer plantillas que ayuden a la realización del método propuesto.

1.4 Justificación

A medida que los requerimientos aumentan, surgen nuevos modelos de procesos para realizar las actividades de la cadena de suministro, y por ende nuevas necesidades de integración de procesos.

Una manera de mantener el control de los procesos es mediante una integración vertical, lo cual implica que las empresas inviertan en infraestructura, personal, capacitación, etc, por lo que habría la necesidad de incurrir en más gastos, por ello que se requiere el uso de métodos que faciliten el diseño, el desarrollo, el despliegue y la administración de sistemas que puedan interactuar entre ellos.

Esta nueva forma de construir sistemas de información, suponen la aparición de una nueva manera de pensar a nivel de Organización de TI, de tal manera que permita la creación de aplicaciones dinámicas que resuelven gran cantidad de problemas de alto nivel fundamentales para el crecimiento y la competitividad. Es por ello que organizaciones como Telefónica Gestión de Servicios Compartidos S.A.C que están en busca de un crecimiento continuo de mercados deben adherirse a estos métodos y para ello debe hacer uso de los mecanismos que tengan al alcance, para no quedar rezagado en esta carrera y ser consiente en todo momento de la situación del mercado de TI.

1.5 Propuesta

La solución que proponemos en esta tesina es un método para desarrollar aplicaciones con enfoque SOA, que permita integrar aplicaciones de diferentes entornos, entre áreas de la organización así como fuera de ellas. Para ello se ha revisado y analizado varios enfoques permitiendo identificar sus fortalezas y debilidades.

En esta propuesta también se proponen plantillas para la elaboración del método de aplicaciones con enfoque SOA.

1.6 Organización de la Tesina

La presente tesina ha sido estructurada en 8 capítulos. En el segundo presentamos el marco teórico del estado actual del conocimiento en el área de procesos y metodologías de desarrollo y el estado actual del conocimiento en el enfoque Service Oriented Architecture (SOA), en el tercer capítulo se presenta el estado del arte de los enfoques para el desarrollo de SOA, en el cuarto capítulo se presenta la resolución de problema aplicando la técnica seleccionada, en el quinto capítulo se presenta la descripción de la solución, en el sexto capítulo se presenta las conclusiones y trabajos futuros, en el séptimo capítulo la bibliografía y en el octavo capítulo los anexos.

Capítulo 2: Marco Teórico Conceptual

2.1 Antecedentes de la investigación

Las empresas necesitan poder interconectar los procesos, personas e información con la propia organización así como atravesando sus fronteras, con subsidiarias y socios comerciales. La falta de integración entre los componentes de TI hacen difícil obtener una respuesta rápida y efectiva ante los cambios que afectan de forma natural a los negocios, la inflexibilidad genera costos, reduce la capacidad de respuesta ante los clientes, compromete el cumplimiento con las normativas legales y afecta negativamente a la productividad de los empleados.

En suma, una deficiente integración es uno de los problemas más importantes a los que las organizaciones debe hacer frente para mantener su competitividad y garantizar su crecimiento.

Una cadena de suministros es un modelo que coordina los procesos administrativos de proveedores, plantas y centros de almacenamiento y distribución, de manera que los bienes sean producidos y distribuidos en las cantidades adecuadas en los lugares y en los tiempos correctos.

Ante este problema la mayoría de las grandes empresas están teniendo que sustituir sus sistemas de información (SI) tradicionalmente centralizados por redes de negocios en las que cada uno de los participantes provee a los demás servicios especializados, por ello muchos SI actuales principalmente los SI Web, se conciben como un medio para ofrecer servicios a través de Internet, involucrando a menudo simples o complejos procesos de negocio en los que colaboran varios participantes.

“Los servicios Web son componentes de software re-usables y bajamente acoplados que encapsulan su funcionalidad en una forma semántica, y que además están distribuidos y accesibles a través de protocolos estándares de Internet”.

Desde el punto de vista orientado a negocio, los servicios web son la tecnología que se utiliza para construir nuevos modelos de comercio electrónico. Los procesos involucrados en estos nuevos modelos están compuestos de tareas las cuales están

distribuidas a través de una red de servicios de valor agregado y se representan como servicios web. A continuación una breve historia del impacto de las tecnologías de información en la cadena de suministro.

- **EDI (Electronic Data Interchange)**

El comercio electrónico, como intercambio electrónico de datos (EDI), se origina en los Estados Unidos en los 60's, como iniciativas independientes en los sectores de ferrocarriles, comestibles y fábrica de automóviles. Se diseñó para asegurar la calidad de los datos que dichos sectores intercambiaban en la cadena de suministro. EDI todavía se utiliza por las empresas con el fin de reducir costos de transacción con sus socios comerciales, sin embargo estas aplicaciones no eran eficaces debido a que utilizan el procesamiento por lote lo que hacía difícil la consistencia de los datos a lo largo de las diferentes fases de la cadena de suministro. Una alternativa fue la iniciática desarrollada por las organizaciones con base tecnológica conocida como CORBA.

- **CORBA (Common Object Request Broker Architecture)**

CORBA permite a aplicaciones hacer peticiones y recibir respuestas en un entorno distribuido, lo cual es la base para construir aplicaciones distribuidas basadas en objetos y llevar a cabo su integración en entornos heterogéneos. El principal componente de CORBA es el Intermediario de Solicitudes a Objetos (Object Request Broker, ORB) que proporciona mecanismos de comunicación entre objetos y se encarga de los detalles de sus invocaciones remotas. Sin embargo, CORBA no ofrece interoperabilidad para los procesos de negocio en una cadena de suministro, debido a que cada socio comercial debe ejecutar su Intermediario de Solicitudes de Objeto, el cual en la práctica es altamente dependiente de la implementación comercial de CORBA. Para resolver el problema de la interoperabilidad entre las implementaciones de CORBA, Microsoft propuso un enfoque alternativo conocido DCOM.

- **DCOM (Distributed Component Object Model)**

DCOM (Distributed Component Object Model) es la extensión a COM (Component Object Model). DCOM es un conjunto de conceptos e interfaces programables de Microsoft en el cual los objetos de un programa cliente pueden solicitar servicios de objetos de un programa servidor que se encuentren en otros nodos de cómputo. Sin embargo, esta tecnología de información presenta ciertas carencias de interoperabilidad en la cadena de suministro. Todos los procesos de negocio de los socios comerciales

participantes en una cadena de suministro deben ejecutarse bajo la plataforma Windows. El lenguaje de programación Java se diseñó con el fin de resolver estos problemas de interoperabilidad.

- **RMI (Remote Method Invocation)**

Java/RMI se basa en un protocolo llamado JRMP (Java Remote Method Protocol) el cual usa la serialización de objetos en Java que permite transmitirlos como una cadena de caracteres. Para realizar esto, el cliente realiza una llamada a un método de un componente del servidor, serializando sus parámetros de entrada. Después, el servidor deserializa la información y reconstruye el objeto correspondiente. Luego, el servidor procesa la llamada y devuelve la respuesta siguiendo los mismos mecanismos de serialización y deserialización usados en la invocación. Sin embargo al igual que las tecnologías de información mencionadas anteriormente, Java/RMI presenta dificultades de interoperabilidad, porque los procesos de negocio deben estar escritos en java. La W3C propuso un lenguaje de descripción independiente de la plataforma conocida como XML.

- **XML (eXtensible Markup Language)**

Por sus siglas en inglés, XML significa lenguaje de marcado extensible y es un lenguaje estándar para el intercambio de datos se desarrolló a finales de los 90's. Su característica de ser extensible se debe a que no tiene un formato fijo como EDI o HTML. XML tiene soporte para estándares como XSL, XML DOM y SAX los cuales se utilizan para la transformación, traducción y validación de documentos XML, Sin embargo se sigue usando el procesamiento por lotes Como consecuencia, los datos de los socios comerciales pueden no estar disponibles en la toma de decisiones en tiempo real.

- **Servicios Web**

Los servicios Web utilizan XML como formato de datos y a SOAP (Simple Object Access Protocol) como mecanismo de transmisión de documentos XML para lograr interoperabilidad entre componentes de software. Los servicios Web proveen un enfoque orientado a servicios y procesos, con el fin de solucionar algunos problemas de sincronización en la cadena de suministro por lo que aplicaciones basadas en servicios Web ofrecen mecanismos de integración con sistemas existentes de una empresa y procesos de negocio de proveedores.

2.2 Bases Teóricas

2.2.1 Procesos y metodologías de desarrollo de software

En esta sesión presentaremos el estado actual del conocimiento en procesos y metodologías de desarrollo, relacionadas con el trabajo de la tesina, por lo no que cubre en forma exhaustiva la vasta variedad de enfoques existentes sino únicamente aquellos que de una forma están relacionados con el trabajo.

2.2.1.1 Ciclo de Vida del Software

Se define y establece que el modelo de ciclo de vida de software sirve como definición de alto nivel de las fases que ocurren durante el desarrollo. No tiene como objetivo proveer información detallada sino resaltar las actividades claves y sus interdependencias. [SWEB04].

2.2.1.1.1 Modelos de Ciclo de Vida del Software

- **Modelo en cascada y en cascada con prototipos**

Es uno de los primeros modelos propuestos, donde las etapas se representan en cascada desde una etapa hacia la siguiente, Cada etapa de desarrollo debe completarse antes de dar comienzo a la siguiente. El mayor problema en este enfoque es que no refleja la manera en que realmente se realiza el desarrollo, donde existe un alto grado de recepción de actividades de una y otra etapa, por ejemplo cuando la solución debe actualizarse para reflejar cambios. Para contrarrestar estos aspectos se incluye el prototipado en las etapas de análisis y diseño dando lugar al modelo en cascada con prototipos.

- **Modelo en fases con desarrollo interactivo e incremental**

Una de las maneras para lograr la reducción del ciclo de vida, es el desarrollo por fases, donde el sistema se diseña para ser liberado en versiones sucesivas, en las que los usuarios disponen de determinadas funcionalidades en operación mientras en

desarrollo se sigue construyendo la versión siguiente que agregará funcionalidades nuevas o mejorará funcionalidades existentes en la versión anterior o ambas.

Dos enfoques populares para organizar el desarrollo de esta manera son el desarrollo incremental y el desarrollo iterativo. En el desarrollo incremental el sistema es particionado en subsistemas de acuerdo con su funcionalidad, y las versiones se desarrollan comenzando con un subsistema funcional pequeño y agregando funcionalidades con cada nueva versión. En el desarrollo iterativo se entrega en la primera versión un sistema completo en cuanto a cubrimiento de funcionalidades en estado básico, y luego en cada nueva versión se cambia y/o mejora la funcionalidad de cada subsistema basado en el uso del sistema.

2.2.1.2 Procesos de desarrollo de Software

Los procesos de desarrollo de software existentes en el área de la Ingeniería de Software, se pueden clasificar actualmente como procesos “pesados” y “livianos” o ágiles. Una metodología “pesada” se puede identificar por tener muchas reglas, prácticas y documentos, requiriendo disciplina y tiempo para seguirla correctamente mientras una metodología “liviana” o ágil tiene solo unas pocas reglas y prácticas que se plantea, son más fáciles de seguir. Como representante más importante de los procesos “pesados” se destaca el Rational Unified Process (RUP), y de los procesos “livianos” o ágiles se destaca el eXtreme Programming (XP).

- **Rational Unified Process (RUP)**

Es un proceso adaptable tanto a las distintas Organizaciones como a los diferentes dominios y tamaños de los proyectos en desarrollo, y desde su primera versión está en constante evolución. Provee un enfoque disciplinado para la asignación de tareas y responsabilidades en la Organización de desarrollo, así como para la ejecución y seguimiento de los proyectos en la Organización.

- **Características principales**

- *Iterativo e incremental*

Este tipo de desarrollo permite la construcción del sistema mediante sucesivas liberaciones que constituyen incrementos en las funcionalidades requeridas, a partir de

un conjunto de funcionalidades y prestaciones básicas definidas conjuntamente con el cliente, que se van obteniendo mediante iteraciones en el desarrollo, en las que se realizan las actividades de las disciplinas definidas. Para cada iteración se define el subconjunto de funcionalidades y prestaciones que integrarán la liberación de la iteración, y para éstas se realizan las actividades correspondientes a las disciplinas definidas.

- *Guiado por Casos de Uso*

La metodología que propone el RUP está basada en la especificación de las funcionalidades del sistema como Casos de Uso, los que muestran la interacción entre los actores identificados y el sistema, según los flujos definidos para cada Caso de Uso. Cada Caso de Uso se compone del flujo básico o principal de ejecución y los flujos alternativos que se definen frente a condiciones que no se cumplen en el flujo principal. También se especifican pre y post condiciones para la ejecución del Caso de Uso y los actores involucrados en el mismo. Los Casos de Uso guían todo el desarrollo ya que a partir de la especificación de los mismos se define y planifica todo el desarrollo, comenzando por la identificación de Casos de Uso relevantes para definir la Arquitectura de Software, definiendo en el Alcance del proyecto para cada fase e iteración que Casos de Uso integrarán la liberación asociada, así como que Casos de Uso compondrán la versión final a obtener.

- *Centrado en la Arquitectura*

Un aspecto clave en la metodología propuesta por el RUP es el enfoque en la definición de la Arquitectura de Software, la cual se realiza en forma temprana en las Fases Inicial y de Elaboración a partir de la identificación de Casos de Uso relevantes a la Arquitectura. Estos Casos de Uso son aquellos que al ejecutar ejercitan aspectos importantes a tener en cuenta en el diseño de la Arquitectura de Software o que son importantes desde el punto de vista del cliente como funcionalidad básica que da valor al sistema.

- **Modelado en dos dimensiones**

El RUP se modela mediante dos dimensiones: la dimensión del Tiempo que muestra los aspectos dinámicos del proceso y define Fases e iteraciones para el

desarrollo, y la dimensión de las Disciplinas que muestra los aspectos estáticos del proceso y define Disciplinas, actividades, entregables y roles para el desarrollo.

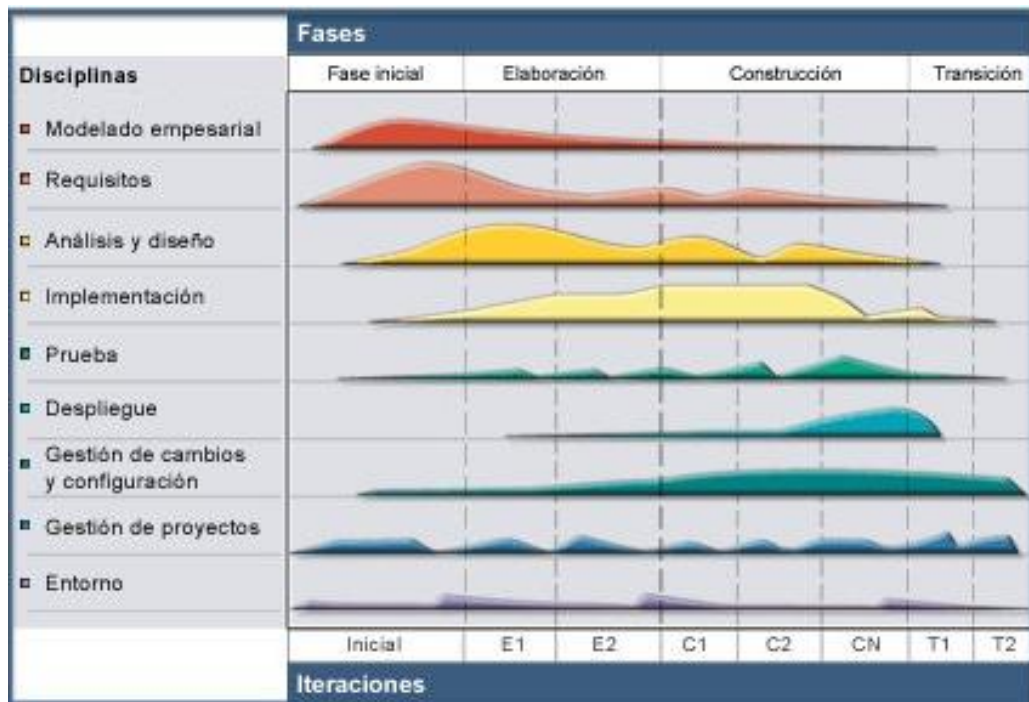


Figura 1: Dimensiones de modelado del RUP de [RUP08]

○ Dimensión del Tiempo

En la dimensión del tiempo se definen ciclos, fases e iteraciones. Un ciclo constituye una realización completa de un proyecto de desarrollo con una liberación de una versión final del sistema en construcción, que constituye una generación del producto. Cada ciclo se divide a su vez en cuatro fases que se nombran como Inicial, Elaboración, Construcción y Transición, cada una de las cuales tiene objetivos que deben cumplirse en cuanto a realización de principales actividades y obtención de entregables más importantes. Estos objetivos son evaluados al finalizar la fase según lo planificado, si los objetivos son obtenidos es posible realizar el pase a la siguiente fase, en caso de no ser así se deberá replanificar el proyecto extendiendo la fase que corresponda hasta que éstos objetivos sean cumplidos.

Fases y sus principales Objetivos

Fase Inicial, el foco del desarrollo se encuentra en la comprensión global del problema y el relevamiento de principales requerimientos, así como la planificación inicial del proyecto y la identificación de los riesgos del proyecto. Si los objetivos son

conseguidos, se deberá realizar la planificación en detalle de la Fase de Elaboración siguiente.

Fase de Elaboración, el principal objetivo consiste en la construcción de la llamada Línea Base de la Arquitectura, compuesta por los Casos de Uso más importantes o escenarios principales de los mismos.

Fase de Construcción, el objetivo más importante es la construcción y testeo del sistema según lo acordado, obteniendo paralelismo en el desarrollo y mejorando la productividad en el área de implementación, así como realizando el testeo completo de las sucesivas liberaciones que se van obteniendo en las iteraciones intermedias.

Fase de Transición, el foco se encuentra en la Disciplina de Implantación en la cual se implanta el producto en el ambiente del cliente y se realizan las pruebas de aceptación sobre el sistema definidas con el cliente. Luego de este último hito el producto es puesto en producción y el ciclo de mantenimiento post liberación comienza.

- *Dimensión de las Disciplinas*

En la dimensión de las Disciplinas se describen los aspectos estáticos del proceso, en base a cuatro elementos que hacen a la definición de proceso de desarrollo: quién está haciendo qué, de qué forma (como) y en qué momento (cuando). Las Disciplinas describen cuando y agrupan actividades en forma lógica, donde las Actividades definen el cómo; los Roles describen quién y las responsabilidades asociadas, y los Entregables describen qué, cuales son los artefactos o productos que serán generados como resultado de la realización de las actividades.

Disciplinas, Actividades, Entregables y Roles

Las Disciplinas definidas por el RUP son: Modelado del Negocio, Requerimientos, Análisis&Diseño, Implementación, Testing, Deployment, Gestión de Configuración y Cambios, Gestión de Proyecto, y Entorno o Ambiente. Para cada Disciplina se definen las actividades y entregables que corresponden a la misma, la participación de los roles en las actividades y entregables correspondientes, y cuando se deben realizar (Fase e iteración) las actividades definidas.

Una actividad se define como el conjunto de acciones en una disciplina para crear o actualizar uno o varios entregables, y consta de pre-condiciones que son los entregables

de entrada, y post-condiciones que son los entregables de salida, así como una descripción de las tareas a realizar en el marco de la actividad y los roles participantes en la misma.

Un entregable o artefacto se define como producto tangible del proyecto que es entrada y salida de las actividades, tiene una descripción del contenido a generar y en general se provee una plantilla como guía para su realización.

Un rol define el comportamiento y las responsabilidades de la/s persona/s que lo cumplen, el comportamiento está dado por las actividades en las que participa, y las responsabilidades están dadas por los entregables que tiene asociados. Un rol puede ser cumplido por más de una persona, y una persona puede cumplir más de un rol en un proyecto de desarrollo.

2.2.2 Metodología para el desarrollo de proyectos

2.2.2.1 Marco Estratégico de Gestión Orientada al Negocio (MEGON)

Metodología que da soporte a estrategia de gestión y facilita la integración con el cliente y los proyectos, usando como componente básico el punto de control. Es importante describir esta metodología en el presente trabajo de tesina por ser el estándar metodológico en la Organización del Grupo Telefónica para todos los proyectos.

Modelo de gestión Megon

Gestión orientada al cliente

En la gestión de proyectos, desde una perspectiva tradicional, predomina la orientación técnica: una gestión basada en la consecución de objetivos de productividad, en la que el cliente asume un papel de espectador. Desde esta óptica, el cliente acepta o rechaza aunque en pocas ocasiones participa.

Pero si se quieren asumir cotas más altas de servicio, es necesario diseñar un sistema donde el cliente tenga un papel esencial, el de la participación y el compromiso. La gestión de proyectos ha de reorientarse hacia un modelo integrado de gestión de clientes y proyectos, Así pues este modelo de gestión se fundamenta en:

- Unos criterios únicos de relación con los clientes

- Unos criterios únicos de calidad

El punto de Control

Es preciso que existan puntos de control que aseguren la calidad y permita la gestión de los diferentes proyectos. De esta manera el punto de control se convierte en una herramienta que:

- Crear un lenguaje común
- Potenciar el dialogo y la participación
- Proporcionar herramientas para medir la producción y calidad
- Es independiente de la tecnología utilizada

Los roles

Quien gestiona también es una cuestión que hay que resolver, En una organización dinámica, el trabajo no puede estar ligado a individuos sino a los roles que estos ejercen. De esta manera cada tarea queda asignada a un rol, pero el individuo puede ejercer un o mas roles.

Los roles que determina el modelo de gestión MEGON son: Gestor del cliente, Jefe de Proyectos, Analista, Diseñador interno, Programador, Cliente, y grupos de apoyo.

La estructura Megon

Este marco propone nueve fases divididas en tres bloques; cada fase está compuesta de actividades, puntos de control y entregables.

Bloque 1: Planificación del sistema de información del cliente. Identificación, definición y priorización de la cartera de proyectos.

Bloque 2: Provisión de soluciones. Desde el análisis de los requisitos hasta la implementación del proyecto a desarrollar.

Bloque 3: Apoyo a soluciones Operativas. Mejoramiento del proyecto implementado.

Hitos y entregables

Los productos o entregables que emita el equipo de proyectos en determinado momento (hitos) deben considerarse versiones incompletas y significativamente perfectibles a medida que avanza el proyecto. Sin embargo la emisión de estas versiones resulta necesaria para realizar un control de resultados y adoptar retroalimentación.

Los hitos y entregables claves identificados para el desarrollo de sistemas de información están orientados al cliente y se basan en los puntos de control y productos del Marco Global Megon. Ambos se encuentran clasificados según la tipología del proyecto (Nuevos Desarrollos, Mantenimiento Evolutivo y Correctivo)

Los hitos representan puntos de control donde se realiza principalmente una actividad de revisión y posterior aceptación, para asegurar la calidad de desarrollo del proyecto, en una modalidad de contención de fases, es decir no se debe continuar a la siguiente fase si no es aceptada la fase anterior.

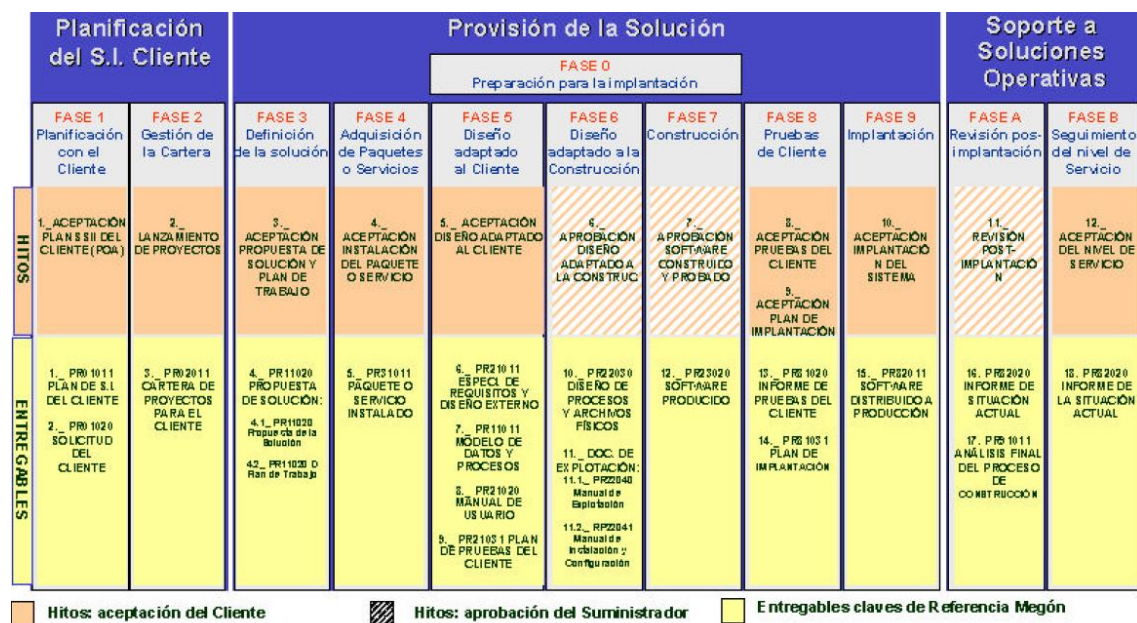


Figura 2: Hitos y Entregables Megon (resumen) [Cast08]

2.2.3 Presentación de la Arquitectura Orientada a Servicios

En las últimas décadas se han experimentado cambios importantes en las tecnologías y paradigmas de desarrollo y uso de aplicaciones, que han impactado fuertemente en la forma en que las Organizaciones realizan e implementan su negocio. Por un lado el acceso cada vez mayor a Internet pone a disposición información y software que permite comunicación entre distintas Organizaciones, sus proveedores y clientes, ampliando tanto los requerimientos del negocio como sus necesidades y realización. Las mejoras en la tecnología continúan en forma acelerada, incrementando a su vez el ritmo de los cambios en los requerimientos. El negocio debe adaptarse rápidamente para sobrevivir en el ambiente dinámico actual, y la infraestructura de TI debe permitir la habilidad de adaptación del negocio. [Endr04]

Esta evolución desde las tecnologías al negocio y viceversa, presenta una parte importante en la evolución de las Arquitecturas de Software desde los primeros sistemas monolíticos hasta la visión actual de orientación a servicios, que se muestra en la figura.

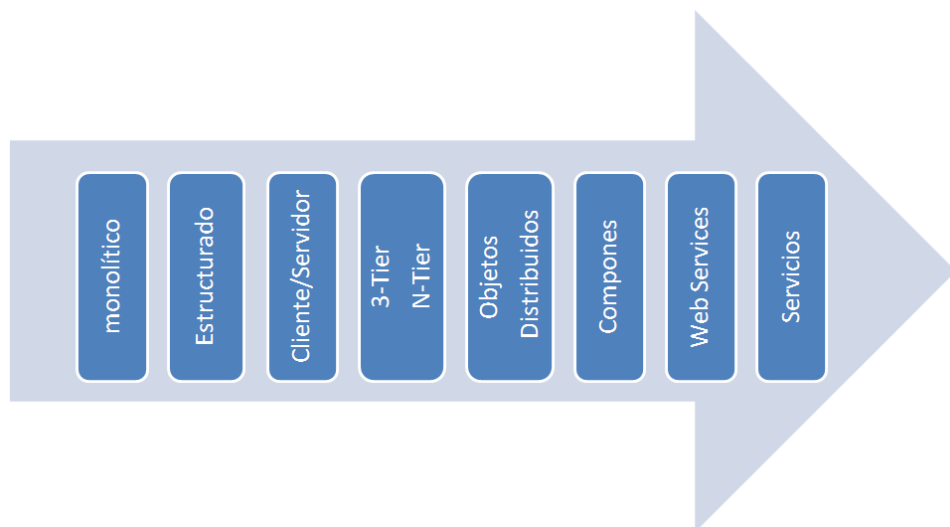


Figura 3: Evolución de Arquitecturas de Software hasta SOA [Endr04]

2.2.4 SOA (Service Oriented Architecture) Arquitectura Orientada a Servicios

En esta sección se presenta el estado actual del conocimiento en el enfoque Service Oriented Architecture (SOA), que constituye la base para el trabajo de tesis. En primer lugar en la sección 2.2.4.1 se presenta el enfoque SOA, brindando el contexto en que se inscribe, en la sección 2.2.4.2 se presentan conceptos de SOA, en la sección 2.2.4.3 se presenta los elementos que componen SOA.

2.2.4.1 Contexto de SOA

Se establece que “La lógica colectiva que define y guía una empresa es una entidad siempre en evolución que cambia constantemente en respuesta a influencias externas e internas. Desde la perspectiva de TI, esta lógica empresarial puede ser dividida en dos mitades importantes: lógica del negocio y lógica de la aplicación. Cada una existe en un mundo por sí mismo, y cada una representa una parte necesaria de la estructura organizacional contemporánea. La lógica del negocio es una implementación documentada de los requerimientos del negocio que se origina en el área del negocio de la empresa“. [Erlt05]

Además “La lógica del negocio está generalmente estructurada en procesos que expresan esos requerimientos, así como cualquier restricción asociada, dependencia e influencias externas. La lógica de la aplicación es una implementación automatizada de la lógica del negocio organizada en varias soluciones tecnológicas. La lógica de la aplicación expresa flujos de los procesos del negocio mediante sistemas adquiridos o desarrollados específicamente por la infraestructura de TI de la Organización, con restricciones de seguridad, capacidad técnica y dependencias con vendedores.”

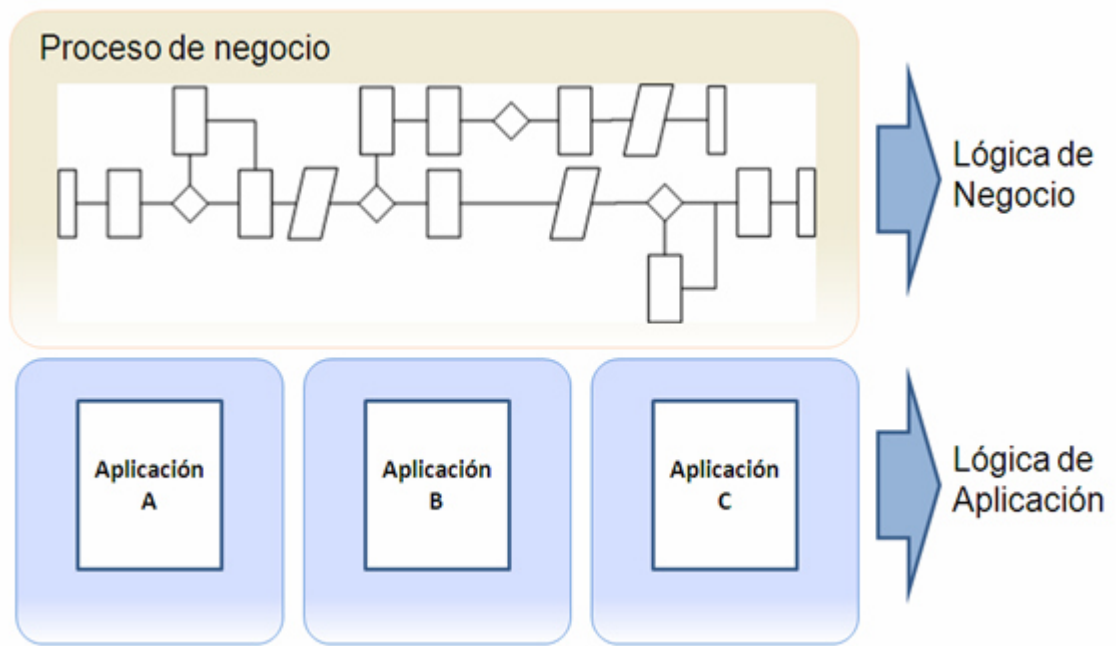


Figura 4: Dos aspectos claves de las Organizaciones [Erlt05]

“Los conceptos introducidos por la orientación a servicios son realizados mediante la introducción de servicios. Los servicios establecen una forma de abstracción de alto nivel entre las capas tradicionales del negocio y la aplicación. Al posicionarse entre medio, los servicios pueden encapsular la lógica de la aplicación física así como la lógica de los procesos del negocio. Los servicios pueden ser estratificados de forma que servicios padre puedan encapsular servicios hijos. Esto permite que la capa de servicios a su vez consista de múltiples capas de abstracción.” [Erlt05]

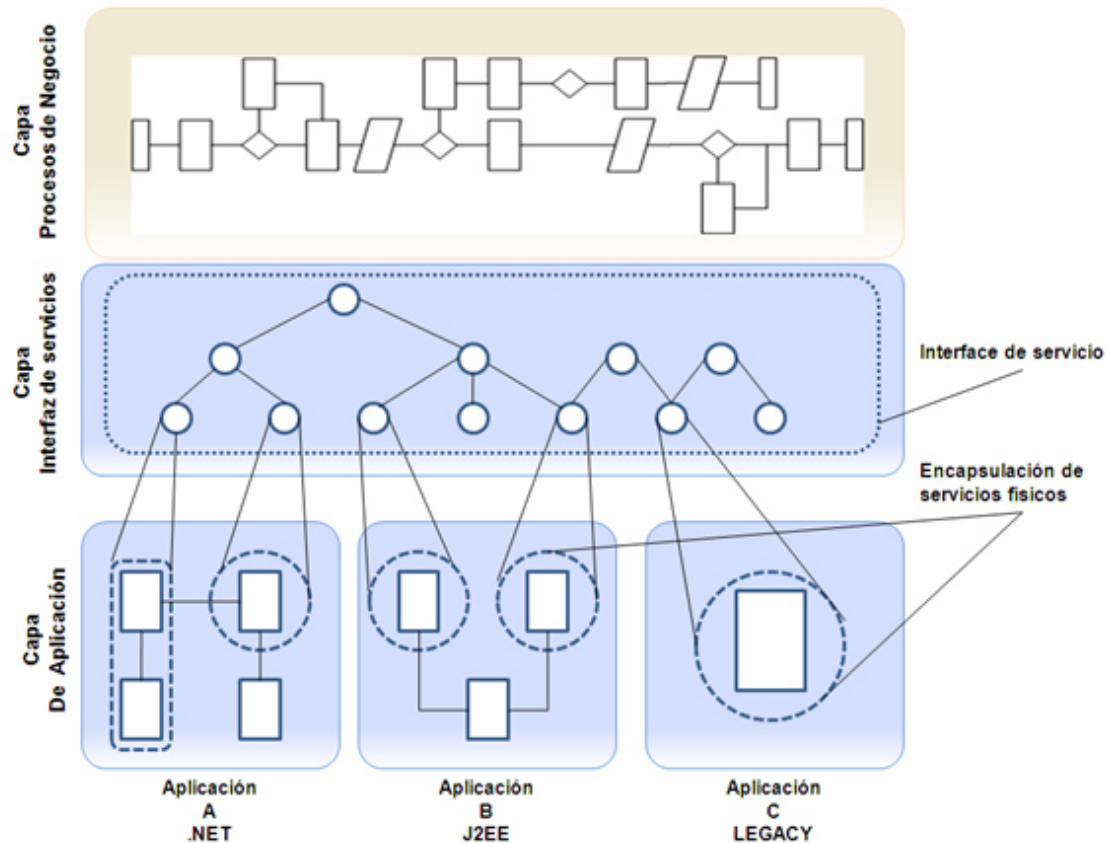


Figura 5: Capa de servicios entre los procesos del Negocio y las tecnologías de [Erlt05]

En el mismo sentido se menciona que los problemas de heterogeneidad, interoperabilidad y requerimientos cambiantes, son solucionados por una SOA que provee una plataforma para la construcción de aplicaciones basadas en servicios con las características de bajo acoplamiento, ubicación transparente de servicios e independencia de protocolos. La infraestructura esconde la mayor cantidad de detalles técnicos posible al consumidor, y diferentes tecnologías de implementación como J2EE o .NET no afectan a los usuarios de una SOA. También debe ser posible sustituir una implementación de un servicio por una mejor si la hay disponible, con mejores características de calidad de servicio, según la metadata que describe al servicio. [Endr04]

2.2.4.2 Conceptos de SOA

No existe una única definición para Service Oriented Architecture (SOA), las que sin embargo comparten algunos conceptos base. A continuación se presentan las definiciones de las Organizaciones de mayor importancia y esfuerzos para la estandarización de este enfoque: OMG, OASIS y W3C. Estas Organizaciones tienen entre sus estándares definidos, algunos tan importantes y ampliamente utilizados como UML y XML.

Para el Object Management Group (Grupo de Gestión de Objetos - OMG) Service Oriented Architecture (SOA) es un estilo de arquitectura para que una comunidad de proveedores y consumidores de servicios alcancen valor mutuo. Permite que los participantes en las comunidades trabajen juntos con una mínima co-dependencia o dependencia tecnológica, mediante la especificación de contratos a los que las Organizaciones, personas y tecnologías deben adherir para participar en la comunidad. Provee la realización por parte de la comunidad, de procesos del negocio y permite que una variedad de tecnologías sean utilizadas para facilitar las interacciones en la comunidad. [OMG]

Según la Organization for the Advancement of Structured Information Standards (Organización para la Promoción de Normas de Información Estructurada - OASIS) Service Oriented Architecture (SOA) es un paradigma para organizar y utilizar capacidades distribuidas que pueden estar bajo el control de distintas Organizaciones. Éstas crean distintas soluciones a los problemas que enfrentan en el curso de sus negocios, que constituyen las capacidades de la Organización para cubrir sus necesidades. La correlación entre necesidades y capacidades no es uno a uno, por lo que una necesidad puede requerir más de una capacidad para ser cubierta, y una capacidad puede cubrir más de una necesidad. El valor percibido de SOA es que provee un framework³ fundamental para asociar necesidades y capacidades y para combinar capacidades para cubrir esas necesidades. [OASIS]

En World Wide Web Consortium (W3C) Service Oriented Architecture (SOA) se define como una forma de arquitectura de sistemas distribuidos que se caracteriza por presentar servicios como una vista lógica abstracta de programas, bases de datos,

³ Framework: Es un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación.

procesos de negocio, cuya semántica se documenta en su descripción, los que tienden a usar un número pequeño de operaciones con mensajes relativamente grandes y complejos, lo que define su granularidad, y tienden a ser orientados a uso sobre la red, aunque esto no es un requerimiento absoluto. Los mensajes son enviados en formato estandarizado neutral a plataformas a través de las interfaces, donde XML es el formato más obvio que cumple estas restricciones. [W3C]

2.2.4.3 Elementos de SOA

SOA se compone de elementos claves que son: los application frontend, los servicios, el repositorio de servicios y el bus de servicios, donde los servicios constituyen la base del enfoque. [Kraf05]

- **Los servicios.** Un servicio consiste en una implementación que provee lógica de negocio y datos, un contrato de servicio, las restricciones para el consumidor, y una interfaz que expone físicamente la funcionalidad. La implementación del servicio provee la lógica del Negocio requerida y los datos correspondientes. El contrato de servicios provee la especificación del propósito, funcionalidad, restricciones y uso del servicio. La forma de esta especificación varía dependiendo del tipo de servicio.
- **Application frontend.** Son los elementos activos de SOA, inician y controlan las actividades en los sistemas empresariales. Pueden tener interfaces gráficas para usuarios como son las aplicaciones Web, o ser procesos batch que no interactúan directamente con los usuarios. Aunque mucha de la responsabilidad sobre los procesos del negocio se delegue en uno o más servicios, es siempre un application frontend quién inicia un proceso del negocio y recibe los resultados del mismo.
- **El repositorio de servicios.** Provee facilidades para descubrir servicios y adquirir la información para utilizarlos, particularmente si estos servicios deben ser descubiertos por fuera del alcance funcional del proyecto que los creo. Aunque mucha de la información requerida es parte del contrato de servicio, el repositorio de servicios provee información adicional como ubicación física, información sobre el proveedor, personas de contacto, restricciones técnicas, aspectos de seguridad.

- **El bus de servicios.** Conecta a todos los participantes de SOA - servicios y application frontends - entre sí. Si dos participantes necesitan comunicarse - por ejemplo, si una application frontend necesita invocar alguna funcionalidad de un servicio básico - el bus de servicios hace esto posible. En algún aspecto, el bus de servicios es similar en concepto al bus de software como está definido en el contexto de CORBA. Sin embargo, existen diferencias significativas entre estos conceptos. La más importante, es que el bus de servicios no está compuesto necesariamente de una única tecnología, sino que preferiblemente se compone de una variedad de productos y conceptos. Como características principales debe proveer: conectividad entre los elementos de SOA, heterogeneidad de tecnología, heterogeneidad de conceptos de comunicación por ejemplo comunicación sincrónica y asincrónica, y servicios técnicos como seguridad, transformación de mensajes, auditorías, entre otros.

Según Thomas Erl SOA se compone de mensajes, operaciones, servicios y procesos (e instancias de procesos). [Erlt05]

- **Los servicios.** Son unidades de lógica independientes. Para retener su independencia, los servicios encapsulan lógica dentro de un contexto distinto. Este contexto puede ser específico a una tarea del negocio, a una entidad del negocio, o alguna otra agrupación de lógica. El interés tratado por un servicio puede ser pequeño o grande. Por lo tanto, el tamaño y alcance de la lógica representada por el servicio puede variar.
- **Un mensaje.** Representa los datos que son requeridos para completar alguna o todas las partes de una unidad de trabajo. Los Message Exchange Patterns (MEPs) definen patrones de intercambios de mensajes comúnmente utilizados. Una operación representa la lógica requerida para procesar los mensajes de forma de completar una unidad de trabajo.
- **Un proceso.** Contiene las reglas del negocio que determinan que operaciones de servicio son utilizadas para completar una unidad de automatización, en otras palabras, un proceso representa una pieza más grande de trabajo que requiere que se completen unidades de trabajo más pequeñas. Adicionalmente los servicios se adhieren a un acuerdo de comunicación, definido en forma colectiva

por una o más descripciones de servicios y documentos asociados, llamados contratos de servicios. Son diseñados para ser descriptivos de forma que puedan ser encontrados y evaluados vía mecanismos de descubrimiento disponibles, por ejemplo registro de servicios.

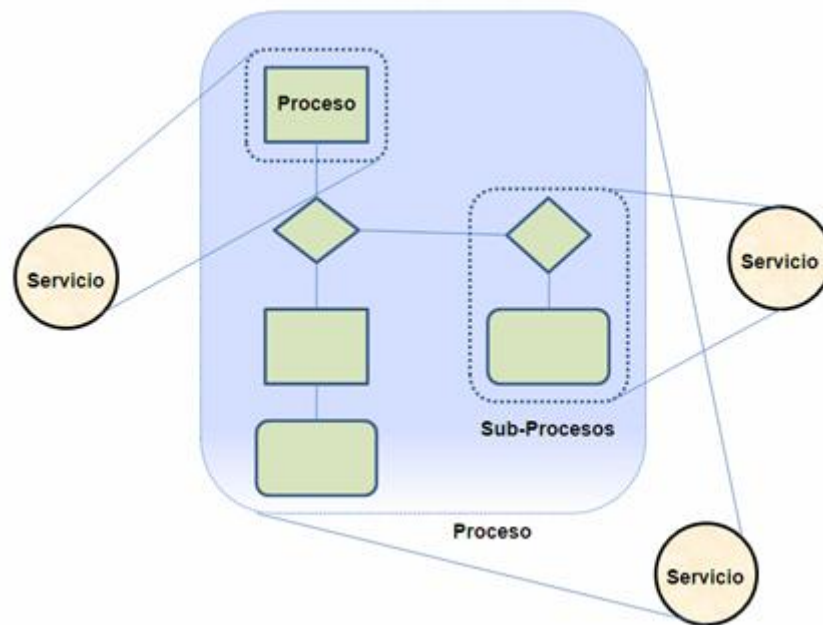


Figura 6: Granularidad de servicios en SOA de [Ert05]

Según Stevens los elementos de que se compone una SOA son: los consumidores y proveedores de servicios, el registro de servicios, el contrato de servicio, el proxy⁴ del servicio y el arrendamiento del servicio. [Stev04]

- **Un consumidor de servicios.** Puede ser una aplicación, otro servicio, o cualquier elemento de software que requiera un servicio. Inicia la ubicación de un servicio en el registro, y una vez que obtiene la ubicación física del mismo, lo ejecuta enviando una solicitud formateada de acuerdo al contrato del servicio.
- **Un proveedor de servicios.** Es el servicio, la entidad direccionable en la red que acepta y ejecuta las solicitudes de los consumidores. Puede ser cualquier tipo de componente de software. El proveedor de servicios publica su contrato en el registro para que sea accedido por los consumidores de servicios.

⁴ Proxy: Es un programa que realiza la acción en representación de otro.

- **El contrato de servicio.** Es una especificación de la forma que un consumidor de un servicio deberá interactuar con el proveedor del servicio. Especifica el formato de la solicitud y respuesta del servicio, precondiciones, postcondiciones, y podría especificar características de calidad de servicio (QoS), por ejemplo, tiempo de ejecución del servicio.
- **El registro de servicios.** Es un directorio en la red que contiene los servicios disponibles. Es una entidad que acepta y almacena contratos de servicios de los proveedores de servicios y provee dichos contratos a los consumidores de servicios interesados.
- **El arrendamiento del servicio.** También es entregado por el registro al consumidor y especifica el tiempo en que el contrato entregado es válido.

En conclusión para Stevens, SOA consiste en los elementos mencionados que configurados en conjunto proveen el soporte para el paradigma base de SOA “descubrir, ligar e invocar”.

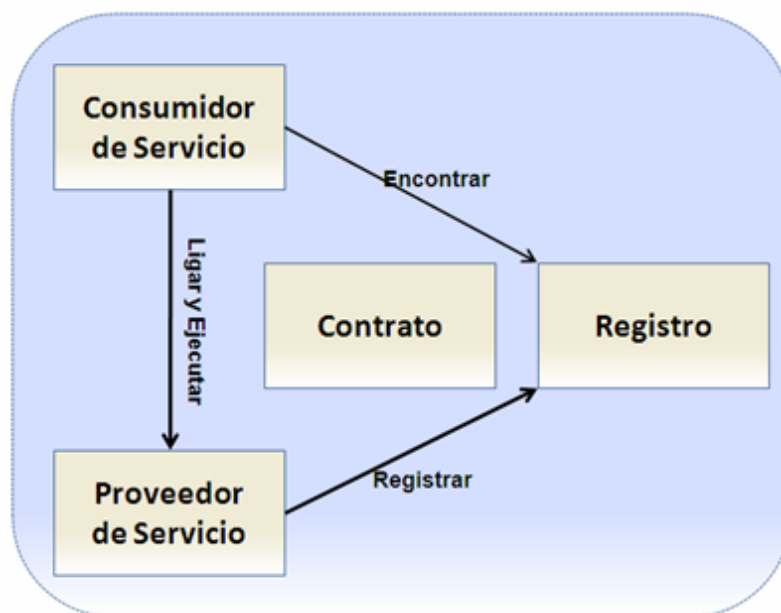


Figura 7: Paradigma “descubrir, ligar e invocar” [Stev04]

2.2.4.4 Clasificación y relaciones entre servicios

Existen varios enfoques para la clasificación de servicios. Cualquiera sea la clasificación o la granularidad de la misma, su importancia radica en el hecho de que sirven como guía al momento de diseñar los servicios.

Una clasificación los separa en básicos, intermediarios, centrados en procesos y empresariales públicos. [Kraf05]

- **Los servicios básicos.** Son la base de SOA, son servidores puros y no mantienen el estado conversacional de la sesión. Se dividen a su vez en servicios centrados en los datos y centrados en la lógica.
 - **Los servicios centrados en los datos,** tienen como propósito manejar los datos persistentes, almacenamiento, recuperación, mecanismos de lockeo⁵ y gestión transaccional. Se comportan similar a la capa de acceso a datos de una aplicación tradicional. La diferencia es que la capa de acceso a datos maneja los datos de la aplicación entera, mientras que un servicio centrado en los datos trata con una sola entidad del negocio.
 - **Los servicios centrados en la lógica,** proveen encapsulamiento para cálculos complejos o reglas del Negocio, tradicionalmente encapsulados en bibliotecas y framework del negocio.
- **Los servicios intermediarios.** Son servicios sin estado que hacen de puente entre las inconsistencias técnicas o discrepancias conceptuales en el diseño. Son tanto consumidores como proveedores, mediando entre los distintos elementos que deben funcionar juntos. Se dividen a su vez en technology gateways, adapters, facades y functionality-adding.
 - **Los servicios Technology gateways,** hacen de puente entre las tecnológicas, incorporando dos o más tecnologías para comunicación o codificación de datos. Hacen de proxys para sus servicios de Negocio y representan la funcionalidad de los servicios que están por debajo en un ambiente que es distinto tecnológicamente que el ambiente de ejecución del servicio de Negocio original.

⁵ Lockeo: Atributo dado por el motor de cualquier objeto del mismo que impide que se pueda realizar ciertas cosas con dicho objeto.

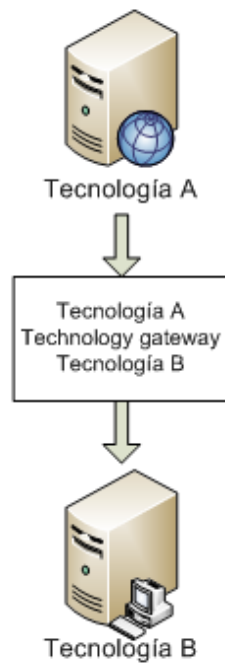


Figura 8: Clasificación de servicio Technology gateways.

- **Los servicios Adapters**, son un tipo especial de servicio intermediario que mapea las firmas y formatos de mensaje de un servicio a los requerimientos de un cliente.

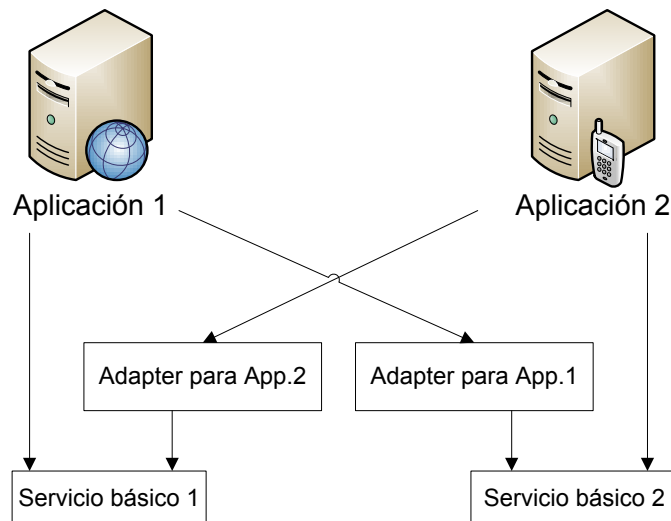


Figura 9: Clasificación de servicio Adapters.

- **Los servicios Facade**, tienen como propósito proveer una vista distinta (y posiblemente agregada) de uno o más servicios existentes, por lo que también pueden actuar como technology gateways y/o adapters. Un facade puede utilizarse para proveer una vista específica de un conjunto de servicios que se encuentran por debajo, que en general son servicios básicos.

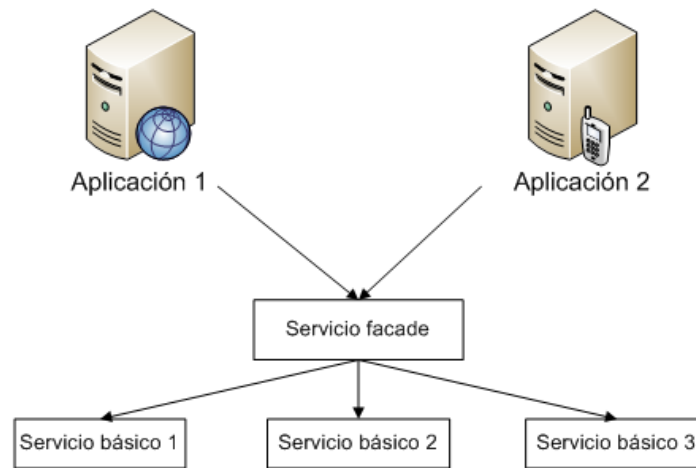


Figura 10: Clasificación de servicio Facade.

- **Los servicios Functionality-adding**, se usan cuando se quiere agregar funcionalidad a un servicio existente sin cambiar el servicio mismo. En este caso se crea un servicio que provea la misma funcionalidad del servicio original y agregue las nuevas características requeridas.
- **Los servicios centrados en procesos.** Encapsulan el conocimiento de los procesos de Negocio de la Organización y controlan y mantienen el estado del proceso en ejecución. Son tanto consumidores como proveedores y preservan el estado. Una de las principales ventajas que presentan estos servicios es que separan la lógica de los procesos. Los procesos del negocio se realizan en estos servicios centrados en procesos, que se componen de secuencias definidas de invocaciones a otros servicios, mediante lo que se conoce como orquestación de servicios.
- **Los servicios empresariales públicos.** A diferencia de los anteriores son servicios que una empresa ofrece a socios y clientes externos. Estos servicios tienen requerimientos específicos de facturación de uso, interfaz,

desacoplamiento y seguridad ya que las empresas deben acordar claramente como se realiza el uso de los mismos.

En la siguiente figura se muestra la ubicación en capas de servicios y las relaciones entre servicios según la clasificación.

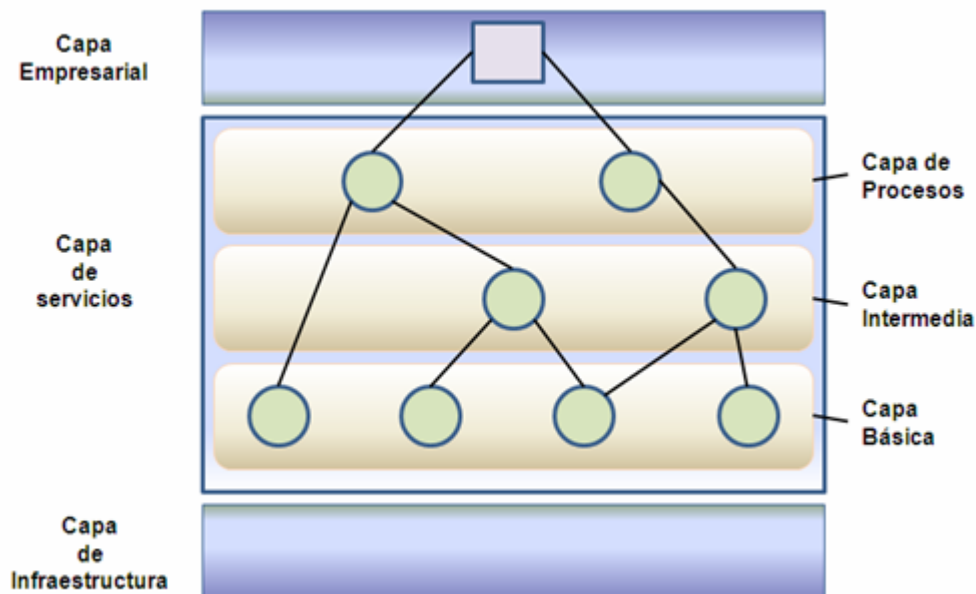


Figura 11: Capas de servicios según la clasificación de [Kraf05]

Thomas Erl, define servicios de aplicación, servicios del negocio y servicios de procesos. [Erlt05]

- **Los servicios de aplicación.** Exponen funcionalidad en un contexto específico de procesamiento, están sobre los recursos que proveen las plataformas, son genéricos y reusables, presentan distinta granularidad en su interface, pueden consistir en una mezcla de servicios desarrollados en forma particular y servicios de terceros comprados o arrendados. Un servicio de aplicación puede componer otros servicios de aplicación de menor granularidad en una unidad de mayor

granularidad de lógica de la aplicación. Los servicios de aplicación se dividen a su vez en servicios de utilidad y wrappers⁶.

- **Los servicios de utilidad**, son servicios que ofrecen lógica reutilizable que es genérica y no específica a una aplicación. Cuando existe capa de servicios del negocio por encima se tiende a que los servicios de aplicación sean servicios de utilidad, proveyendo operaciones reusables que pueden componerse por los servicios del negocio para realizar requerimientos de procesamiento centrados en el negocio. Cuando no existe capa de servicios del negocio por encima, muchas veces se agrega lógica del negocio en los servicios de aplicación, dando lugar a los servicios híbridos, si bien no recomendados, son comunes.
- **Los servicios wrapper**, encapsulan y exponen lógica que reside en un sistema legado. Son generalmente utilizados para propósitos de integración. Consisten en servicios que encapsulan alguna o todas las partes de un ambiente legado para exponer funcionalidad legada a los servicios solicitantes.
- **Los servicios del negocio.** Representan lógica del negocio en la forma más pura posible. Sin embargo esto no previene que según las variaciones de lo que implementa, un servicio del negocio pueda ser clasificado también como un servicio controlador (de integración) o un servicio de utilidad (híbrido). Los servicios del negocio se dividen a su vez en servicios del negocio centrados en tareas y servicios del negocio centrados en entidades.
 - **Los servicios del negocio centrados en tareas**, encapsulan lógica del negocio específica a una tarea o proceso del negocio. Este tipo de servicios en general es requerido cuando la lógica del proceso del negocio no está centralizada en servicios de procesos en la capa por encima.
 - **Los servicios del negocio centrados en entidades**, encapsulan una entidad del negocio específica. Son útiles para crear servicios del negocio altamente reusables independientes de los procesos del negocio,

⁶ Wrapper: Es un programa para controlar el acceso a un segundo programa.

que luego puedan componerse en servicios de procesos o servicios del negocio centrados en tareas, o ambos.

- **Los servicios de procesos.** Componen servicios del negocio y de la aplicación de acuerdo a reglas del negocio y lógica del negocio embebida en la definición del proceso. Componen otros servicios que proveen conjuntos específicos de funciones independientes de las reglas del negocio, con lógica específica de un escenario requerida para ejecutar una instancia de un proceso. Los servicios de procesos son también servicios controladores ya que requieren componer otros servicios para ejecutar lógica de los procesos del negocio.

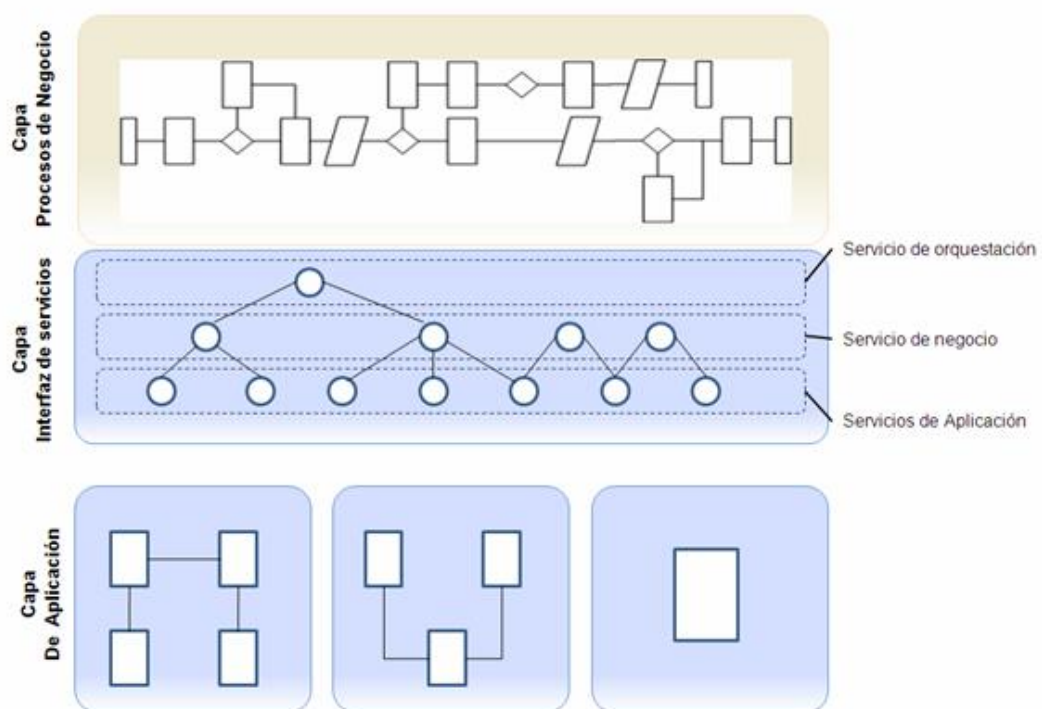


Figura 12: Capas de servicios según la clasificación de [Erlt05]

2.2.4.5 Procesos de negocio

En las secciones anteriores se mencionó brevemente como los procesos del negocio pueden ser representados y expresados a través de servicios de procesos o centrados en procesos. Según [Erlt05] “particionar la lógica del negocio en servicios que pueden componerse tiene implicaciones significativas sobre cómo se pueden modelar los procesos del negocio. Los analistas pueden influenciar estas características incorporando una extensión de orientación a servicios a los procesos del negocio para su

implementación mediante SOAs. Los servicios además pueden ser diseñados para expresar lógica del negocio. Modelos de Business Process Management (BPM), de entidad, y otras formas de inteligencia del negocio pueden ser representadas con precisión a través de la composición coordinada de servicios centrados en el negocio.

2.2.4.6 Orquestación de Servicios

Una orquestación es una forma de composición de servicios que permite definir procesos del Negocio específicos para un proyecto, mediante una secuencia definida de invocaciones a servicios existentes [Erlt05].

Según Thomas Erl “Una orquestación expresa el cuerpo de la lógica de un proceso del negocio que es típicamente propiedad de una Organización. Establece un protocolo de negocio que define formalmente la definición de un proceso del negocio. La lógica del workflow en una orquestación es separada en una serie de actividades básicas y estructuradas que pueden ser organizadas en secuencias de flujos. El alcance de una única orquestación, entonces puede ser clasificada como una actividad compleja y posiblemente de larga duración. La orquestación provee un modelo de automatización donde la lógica del proceso aunque centralizada, es aun extensible y componible. Mediante el uso de orquestaciones los ambientes de soluciones orientadas a servicios se vuelven inherentemente extensibles y adaptables. La orquestación en si misma típicamente establece un punto común de integración para otras aplicaciones.” [Erlt05].

2.2.4.7 Coreografía de servicios

Para Thomas Erl “El objetivo de un coreografía es establecer un tipo de colaboración organizada entre servicios que representan distintas entidades de servicios, solo que ninguna entidad (u organización) necesariamente controla la lógica de la colaboración. Las coreografías entonces proveen el potencial para establecer patrones de interoperatividad universal para tareas del negocio comunes inter-organizacionales.” [Erlt05].

Tanto las orquestaciones como las coreografías son formas de composición de servicios que permiten realizar procesos del Negocio dentro de una Organización o entre varias Organizaciones. “Mientras ambas representan patrones de intercambio de mensajes complejos, hay una distinción común que separa los términos orquestación y coreografía. Una orquestación expresa un flujo del negocio específico a una

organización. Esto significa que la organización es dueña y controla la lógica detrás de la orquestación, incluso si esa lógica involucre interacciones con socios de negocio externos. Una coreografía, por otro lado, no necesariamente es propiedad de una única entidad. Actúa como un patrón de intercambio de la comunidad utilizado para propósitos colaborativos por servicios de distintas entidades proveedoras. “[Erlt05]

2.2.4.8 Niveles de realización de SOA

Para que una organización, que cuenta con varios sistemas separados en tecnologías heterogéneas, incorpore el enfoque SOA, se hace necesario realizar en forma previa una evaluación de la organización y una planificación de cómo será la incorporación de los distintos elementos que componen SOA.

Existe una propuesta de utilizar un “architectural roadmap” o itinerario arquitectónico que sirve como base para tener en cuenta las distintas etapas por las que debería transitar una organización para incorporar SOA en forma segura. [Kraf05]

- **Fundamental SOA:** Este nivel consiste únicamente de dos capas de abstracción, la capa básica compuesta por servicios básicos y la capa empresarial donde se encuentran las application frontend.
- **Networked SOA:** En este nivel se agrega la capa de abstracción intermediaria que se compone de servicios intermediarios, que pueden incluir servicios del tipo facades, technology gateways, adapters y functionality-adding. Como se mencionó, los servicios intermediarios encapsulan parte de la complejidad de los servicios básicos, y permiten la integración de conjuntos de software en forma independiente de la tecnología.
- **Process-enabled SOA:** El nivel de process-enabled SOA es el nivel más completo de SOA, donde los procesos del Negocio se realizan con servicios centrados en procesos que orquestan servicios de las capas por debajo. Agrega la capa de abstracción de procesos que se compone de servicios centrados en procesos que encapsulan la complejidad de los procesos del Negocio, comparten el estado entre múltiples clientes y manejan procesos de larga duración. La

complejidad de las aplicaciones en el backend⁷ es manejada por los servicios intermediarios y los servicios centrados en procesos.

Según Thomas Erl no se proponen niveles sino que se plantea como realizar las etapas del ciclo de vida de creación hasta puesta en producción de los servicios. Para esto se plantea utilizar enfoque top-down, bottom-up y ágil o “intermedio”. [Erlt05]

- **Enfoque top-down:** Esta estrategia promueve la definición formal de modelos del negocio corporativos antes de modelar los servicios. Es un enfoque de “primero análisis” que requiere que no solo los procesos del Negocio se vuelvan orientados a servicios, sino que también promueve la creación (o realineamiento) del modelo del negocio entero de una Organización. Soporta la creación de las tres capas de servicios presentadas.
- **Enfoque bottom-up:** Esta estrategia se basa en la liberación de servicios de aplicación según sea necesario. Esencialmente fomenta la creación de servicios como forma de cumplir con requerimientos centrados en las aplicaciones. Los servicios son construidos según sea necesario y modelados para encapsular lógica de aplicación para servir mejor a los requerimientos inmediatos de la solución.
- **Enfoque ágil o intermedio:** Propone una combinación de los enfoques top-down y bottomup. El desafío consiste en encontrar un balance aceptable entre la incorporación de principios de diseño orientado a servicios en los ambientes de análisis del negocio sin tener que esperar por aspectos tecnológicos asociados. Para muchas organizaciones puede ser útil contar con un enfoque intermedio entre los otros dos extremos que se proponen.

2.2.4.9 Aspectos técnicos de la realización de SOA

Existen varios aspectos técnicos que deben ser tenidos en cuenta al momento de incursionar en la puesta en práctica del enfoque SOA, principalmente asociados a la

⁷ Backend: comprende los componentes que procesan la salida del front-end.

definición, implementación y distribución de servicios, comunicación entre estos, incluyendo confiabilidad, forma y protocolos a utilizar, notificación, manejo de eventos, publicación y descubrimiento de servicios, integridad transaccional, compensaciones, seguridad, entre otros. Estos aspectos tienen distintas características según sea la implementación de SOA que se elija. A continuación se mencionan brevemente aspectos de la modelización con Service Component Architecture (SCA) e implementación con Web Services.

- **Modelización con Service Component Architecture (SCA)**

SCA es una especificación que describe un modelo para construir aplicaciones y sistemas utilizando Service Oriented Architecture. Mientras los sistemas basados en SOA pueden tener servicios individuales implementados utilizando tecnología orientada a objetos (entre otros enfoques), el diseño global del sistema es orientado a servicios. SCA está basado en una especificación abierta, permitiendo que múltiples vendedores implementen el soporte para SCA en sus herramientas de desarrollo y ejecución. Esto es particularmente importante para el despliegue, administración y configuración de las aplicaciones basadas en SCA.

Características:

- Permite una variedad de implementaciones de componentes y tipos de interfaces.
- Complementa la implementación con Web Services, proveyendo una forma para empaquetar servicios en un sistema del negocio, así como un modelo de construcción de servicios. Sin embargo SCA reconoce que utilizar Web Services no es la única forma de implementar una SOA por lo que SCA soporta una amplia gama de tecnologías.
- Los componentes pueden ser contruidos con una variedad de tecnologías como EJB, CORBA y lenguajes de programación incluyendo Java, PHP, C++.
- Los componentes SCA pueden ser conectados también mediante una variedad de ligamientos como WSDL/SOAP para web services, Java Message Service para middleware con orientación a mensajes, y J2EE Connector Architecture para servicios de información empresarial.

- Provee soporte para comunicación sincronía, asíncrona, conversacional, también para la definición de políticas por ejemplo de seguridad.
- SCA divide la implementación en dos pasos principales, primero la implementación de componentes que proveen servicios y consumen otros servicios, segundo el emsamblaje de componentes para construir la aplicación mediante la conexión de referencias a servicios con los servicios.
- El modelo plantea también la utilización de Service Data Objects (SDO) para el intercambio de datos de manera uniforme entre servicios, encapsulando la forma de acceso a los mismos, que consiste en documentos XML que definen los datos y la forma de obtenerlos.

- **Implementación con Web Services**

Aunque no es la única forma para implementar servicios, es una forma que brinda mayor interoperabilidad debido a la estandarización de los protocolos para el diseño y la comunicación entre los Web Services, lo que resulta muy importante principalmente para las relaciones interorganizacionales, ya que utilizando Web Services es posible comunicar software implementado en distintas plataformas.

Existen estándares definidos que permiten que las organizaciones proveean y consuman servicios.

- Lenguaje de marcas común para comunicación: un lenguaje de marcas común facilita la comunicación entre proveedores y consumidores, ya que cada parte puede leer y comprender la información intercambiada basada en las marcas embebidas. Los Web Services utilizan el eXtensible Markup Language (XML) como lenguaje de marcas común y XML Schema para definición de la estructura de los documentos XML intercambiados.
- Formato de mensajes común para intercambio de información: para la comunicación las partes deben acordar el intercambio de mensajes basado en un formato acordado. Teniendo ese formato, las partes aunque sean desconocidas, pueden comunicarse en forma efectiva. El formato común para mensajes utilizado por los Web Services es el Simple Object Access Protocol (SOAP).

- Formato de especificación de servicios común: para describir los detalles de un servicio, como el tipo de servicio, como accederlo, y otros, se debe utilizar un mecanismo estándar para que los proveedores especifiquen los servicios y los consumidores puedan entenderlo y utilizarlos. El Web Services Description Language (WSDL) provee a los Web Services un formato de especificación común.
- Forma común para búsqueda de servicios: para descubrir y obtener los detalles de un servicio, los consumidores deben tener una forma común de hacerlo. Esto se logra teniendo ubicaciones comunes bien conocidas donde los proveedores registran sus servicios y los consumidores los buscan. De esta forma, los servicios pueden ser accedidos universalmente por los proveedores y consumidores. La especificación Universal Description, Discovery and Integration (UDDI) define una forma común de búsqueda de Web Services.

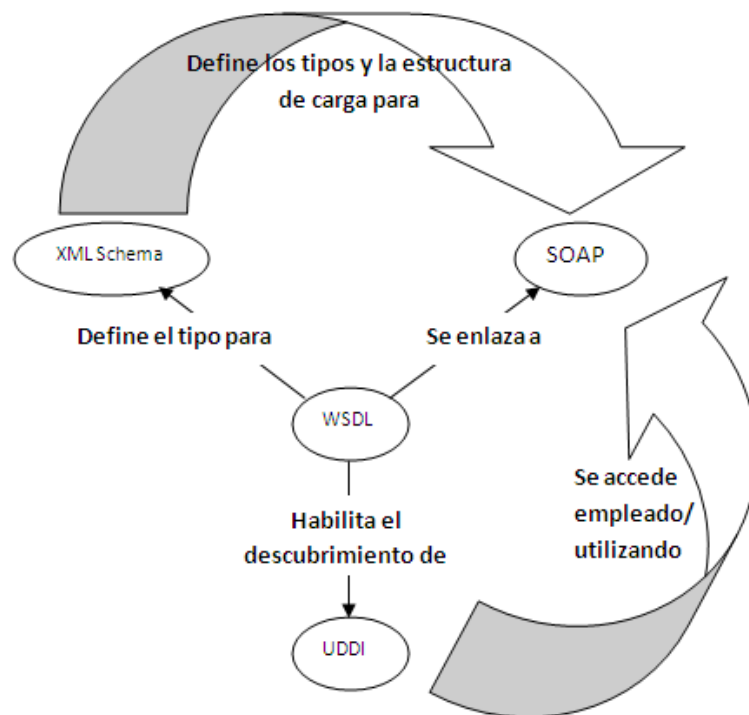


Figura 13: Marco de referencia de implementación con Web Services de [Ertt05]

Existen otros esfuerzos de estandarización del uso de Web Services como el de Web Services Interoperability Organization (WS-I) que es un grupo formado por

organizaciones que cruzan varias industrias creado para promover la interoperabilidad de Web Services a través de distintas plataformas, sistemas operativos y lenguajes de programación.

- **Herramientas Asociadas**

En general la mayoría de los grandes proveedores de herramientas de desarrollo y software de base, tanto comerciales como open source, han incorporado elementos relacionados con la inclusión de Web Services con capacidades de generación de los mismos, y definición y ejecución de procesos del Negocio con modelado desde los IDE de desarrollo, generación de BPEL y ejecución en motores de procesos. Como principales herramientas se pueden mencionar las incluidas en IBM SOA Foundation, Oracle SOA Suite, Microsoft SOA Platform, Sun Java CAPS, Jboss SOA Platform y Eclipse SOA Tools Platform (STP).

2.2.5 Business Process Modeling (BPM)

Como se vio anteriormente uno de los aspectos principales de una SOA completa es que los procesos del negocio se realicen como servicios centrados en procesos, que mediante orquestación de otros de menor granularidad, ejecuten los flujos definidos para dichos procesos.

Aunque los servicios centrados en procesos pueden ser realizados de varias maneras distintas (y por lo tanto pueden tomar muchas formas distintas), Business Process Management (BPM) representa posiblemente el enfoque más consecuente para lograr una process-enabled SOA [Kraf05].

A continuación se presenta brevemente el enfoque Business Process Management (BPM), en 2.2.5.1 se introducen conceptos del enfoque, en 2.2.5.2 se presenta el modelado de procesos del negocio que propone, y finalmente en 2.2.5.3 se muestra como se incluye el enfoque en SOA. No es la intención en esta sección entrar en detalles del enfoque, sino presentar sus principales conceptos para poder establecer cómo y porqué se relaciona con SOA, y la importancia que tiene en SOA la inclusión de este enfoque.

2.2.5.1 Presentación de BPM

“Durante la ola de reingeniería de procesos de los años 90, diversos libros de gestión contaban experiencias de empresas para utilizar como guía de transformación del negocio. A pesar de que la teoría subyacente estaba basada en sentido común “de siempre” y teoría general de sistemas propuesta cincuenta años atrás, no se ofrecía un camino para su ejecución. En contraste, la organización que gestiona sus procesos, toma control de sus procesos internos y se comunica con un lenguaje de procesos universal que permite que los socios ejecuten una visión compartida, comprendiendo las operaciones de cada uno, conjuntando diseño de procesos y gestionando el ciclo de vida completo de sus iniciativas de mejora del negocio” [Shfp03]

Business Process Management Initiative (BPMI) define BPM como el “conjunto de actividades que realizan las Organizaciones para optimizar o adaptar sus procesos de negocio a las nuevas necesidades organizacionales”.

BPM introduce el concepto de “procesamiento de procesos” y enfatiza que este concepto no está limitado a la ejecución automática de modelos de procesos digitales. [Kraf05]

2.2.5.2 Modelado de Procesos del Negocio con BPM

Business Process Management Initiative (BPMI) promueve tres estándares: Business Process Modeling Notation (BPMN) para modelado de procesos, como estándar de notación para especificarlos; Business Process Modeling Language (BPML) para ejecución de procesos, como estándar de Business Process Execution Language (BPEL) y Business Process Query Language (BPQL) para distribución y ejecución de procesos de e-Business, como interface de gestión estándar. [BPMI]

Una característica crucial y fundamental que distingue los estándares propuestos por BPMI es que han sido desarrollados con una base matemática sólida. La rama de Pi-Calculus del Process Calculi fue utilizada, que es un método formal de computación que provee la base de procesos dinámicos y móviles. Esto hace que dichos estándares sean análogos en su base matemática a la teoría relacional de conjuntos.

2.2.5.3 Business Process Modeling Notation (BPMN)

Business Process Modeling Notation (BPMN) es un estándar para modelar visualmente flujos de procesos que tiene como objetivo proveer notación común a analistas del negocio que crean los flujos iniciales de los procesos y desarrolladores de software responsables por la tecnología e implementación de los procesos.

Está basado entre otros en Diagrama de Actividad de UML y Diagrama de Flujo Actividad-Decisión, por lo que tiene similitud con ambos, pero su público objetivo son los analistas del negocio. BPMN especifica un único tipo de diagrama, el Business Process Diagram (BPD). [BPMN06]

BPMN provee un número de ventajas para modelar procesos del negocio sobre UML. Primero, ofrece una técnica de modelado del flujo de los procesos que es más propicia a la forma de modelado de los analistas del negocio. Segundo, su base matemática sólida está expresamente diseñada para mapear a Business Process Execution Languages (BPEL), mientras UML no. BPMN puede mapear a UML, y proveer un frontend de modelado del negocio sólido para sistemas diseñados con UML [Owen03].

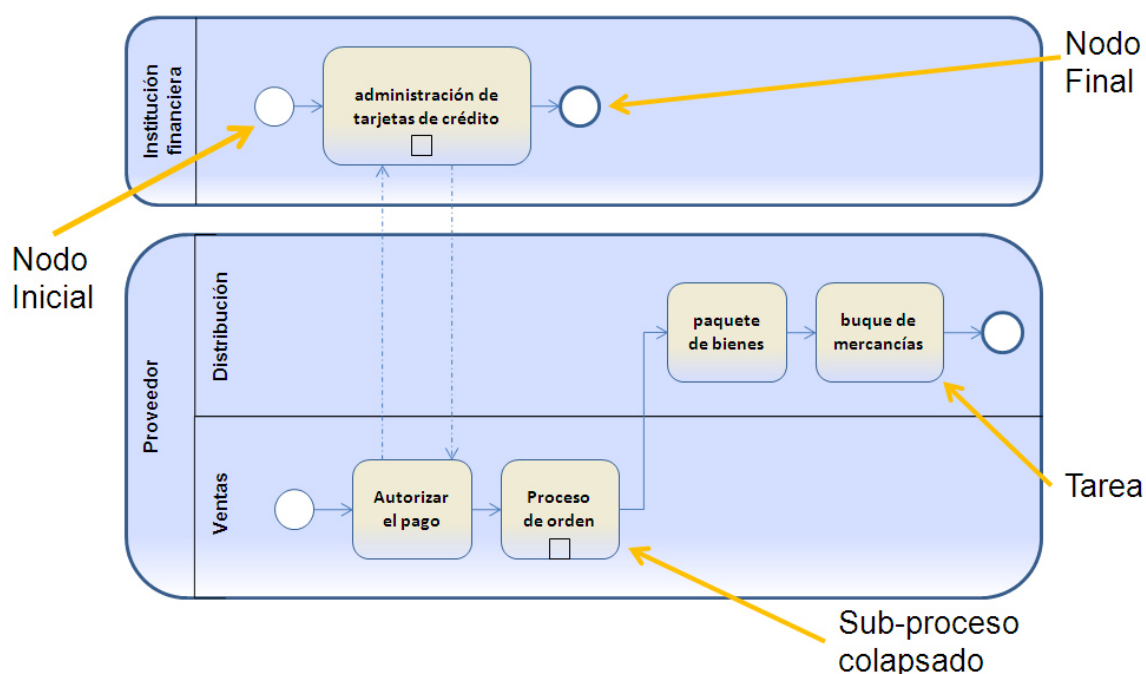


Figura 14: Ejemplo básico de Business Process Diagram (BPD) en BPMN de [BPMN06]

BPMN incluye tres tipos básicos de submodelos en un modelo completo en BPMN:

1. Procesos del Negocio Privados (internos), son internos a una Organización específica y son el tipo de procesos que generalmente han sido llamados workflows o procesos BPM.
2. Procesos Abstractos (públicos), representan las interacciones entre procesos del Negocio privados y otro proceso o participante (por ejemplo humano).
3. Procesos de Colaboración (globales), Un proceso de colaboración muestra las interacciones entre una o más entidades de negocio.

Estas interacciones se definen como una secuencia de actividades que representan los patrones de intercambio de mensajes entre las entidades involucradas [BPMN06]. Los primeros corresponderán a orquestaciones y los últimos a coreografías, los procesos abstractos dependerá de quién es el otro participante el alcance del mismo.

2.2.5.4 Business Process Modeling Language (BPML)

Business Process Modeling Language (BPML) es un estándar para lenguajes de ejecución de procesos (BPEL) basado en XML. Establece un formato estándar para expresión e intercambio de procesos independiente de la implementación. BPMN se mapea directamente a Business Process Modeling Language (BPML) y también a Business Process Execution Language for Web Services (BPEL4WS).

BPML es una especificación tanto para modelar procesos del Negocio como para construir sistemas de gestión de procesos. Provee el modelo abstracto para todos los procesos, junto con sintaxis y XML-Schema basados en estándares para expresar y gestionar procesos de Negocio. Un aspecto clave de BPML es que es directamente ejecutable en una infraestructura de TI. BPML es ejecutado por una “máquina virtual de procesos” en el BPMS. El diagrama representa el código y el código representa el diagrama. No hay pérdida de información al moverse de uno al otro.” [Shfp03].

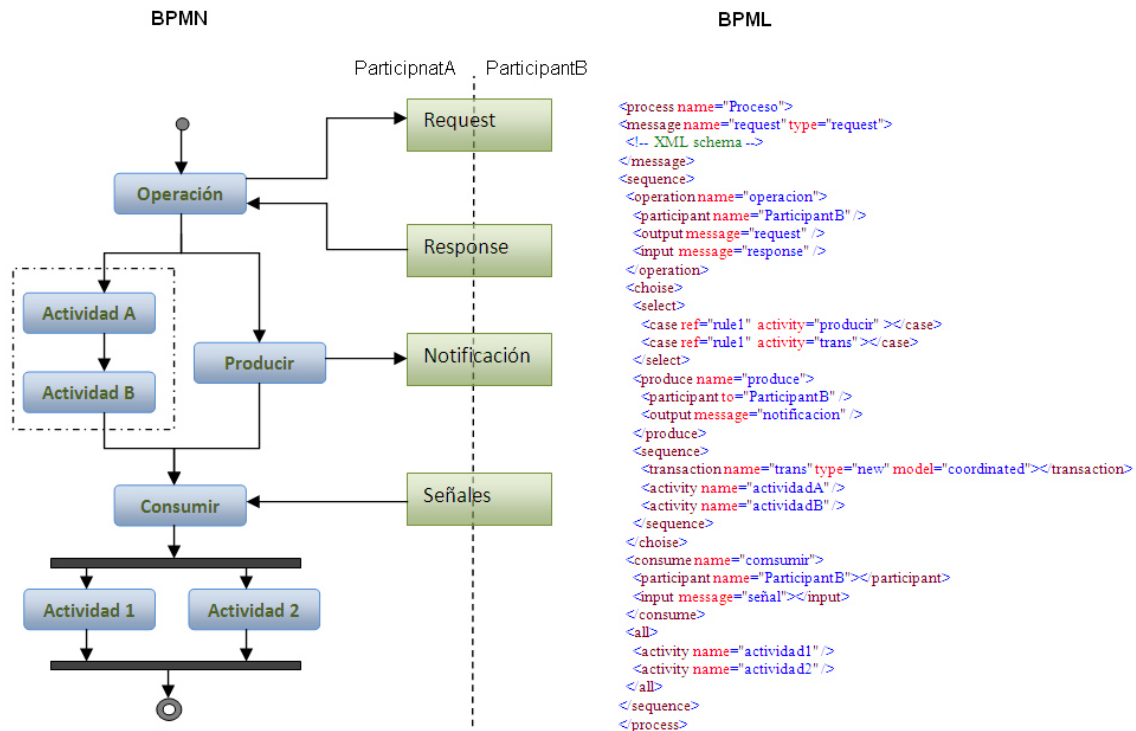


Figura 15: Correspondencia uno a uno entre BPMN y BPML de [Shfp03]

2.2.5.5 Business Process Management Systems (BPMS)

“Un producto BPMS debe permitir a los analistas del negocio, desarrolladores de software y administradores de sistemas, modelar y distribuir (deploy) procesos del negocio (en tiempo de desarrollo) e interactuar, monitorear y analizar instancias de procesos (en tiempo de ejecución)”. [Kraf05]

“BPMS permiten una metodología de tres pasos hacia la integración de los procesos de Negocio, en cuya aplicación se verán involucrados distintos roles, principalmente los analistas del negocio durante todo el proceso, pero también desarrolladores de software y administradores de sistemas. El primer paso involucra el modelado del proceso del Negocio mediante una interface gráfica (GUI) y los patrones de diseño de procesos subyacentes son almacenados en un repositorio de procesos, accesible a varios usuarios en la red. En segundo lugar los procesos almacenados en el repositorio son instalados en el servidor de procesos mediante herramientas automáticas, para lo cual el servidor de procesos no tiene porqué ser interrumpido, permitiendo entonces agregado y modificación de procesos del negocio en forma dinámica. Mediante herramientas provistas también se puede consultar el estado de cualquier instancia de procesos y del

servidor de procesos. En tercer lugar los analistas del negocio y administradores de sistemas pueden gestionar los procesos que se están ejecutando, utilizando lenguajes de consulta de procesos estándares.” [Shfp03].

2.2.6 SOA y BPM juntos

Garther plantea que “BPM es un alto nivel de gestión, conjunto de disciplinas y principios y SOA es un nivel inferior, conjunto de principios técnicos”. [Garther07]

Se plantea también que “la visión de BPM es una visión fuerte, en vez de incluir directamente en el código información y reglas sobre procesos del Negocio importantes, esta información es retirada de los sistemas de aplicación y puesta bajo el control de un BPMS. BPM facilita la modificación, reconfiguración, y optimización de definiciones de procesos con herramientas gráficas que pueden ser utilizadas por analistas del negocio con menor orientación tecnológica. SOA provee la funcionalidad del backend que es requerida por un BPMS para implementar la funcionalidad de sus procesos. Todos los conceptos discutidos previamente respecto a SOA, incluyendo el desarrollo evolutivo de capas básica, intermediaria y de procesos, tienen sentido en este contexto. SOA se convierte en la infraestructura que permite la Organización orientada a procesos.” [Kraf05]

Así como existen diferencias entre BPM y SOA también existen similitudes por ejemplo, hacer frente a los procesos de negocio, y es más importante que se complementen. Procesos de Negocio gestionados por BPM puede utilizar (consumir) los servicios a las empresas que se definen y se administren a través de SOA.

Un aspecto importante de BPM y SOA es la que normalmente los procesos (o un punto de acceso a los mismos) que se definen en el BPM puede ser publicado como un servicio Web. Tal servicio puede luego ser consumidos por otros servicios en las composiciones y orquestaciones o por otros procesos de negocio en BPM. Por lo tanto, BPM y SOA forman un bi-direccional y recursivo vínculo en términos de la edición y el consumo de los servicios Web: los servicios Web (en SOA) puede ser consumida por los procesos de negocio (BPM). A cambio, los procesos de negocio en BPM pueden ser publicados como servicios Web y consumido por otros servicios ó procesos.

Además, la vinculación de BPM a SOA también ayuda a abordar la mayoría de las cuestiones relacionadas con el gobierno de SOA y servicios. Los servicios de software con más facilidad se pueden compartir y recombinar en un SOA. Es cada vez más deseable poner proceso explícito del negocio en ejecución combinando servicios reutilizables.

Por las razones indicadas arriba, la atadura de servicios reutilizables a los procesos explícitos del negocio hace que SOA sea más eficaz.

Según Garther “Aunque SOA puede tener éxito sin herramientas de software BPM, uno nunca debe hacer SOA sin la participación de la administración de empresas y analistas de negocios”. [Garther]

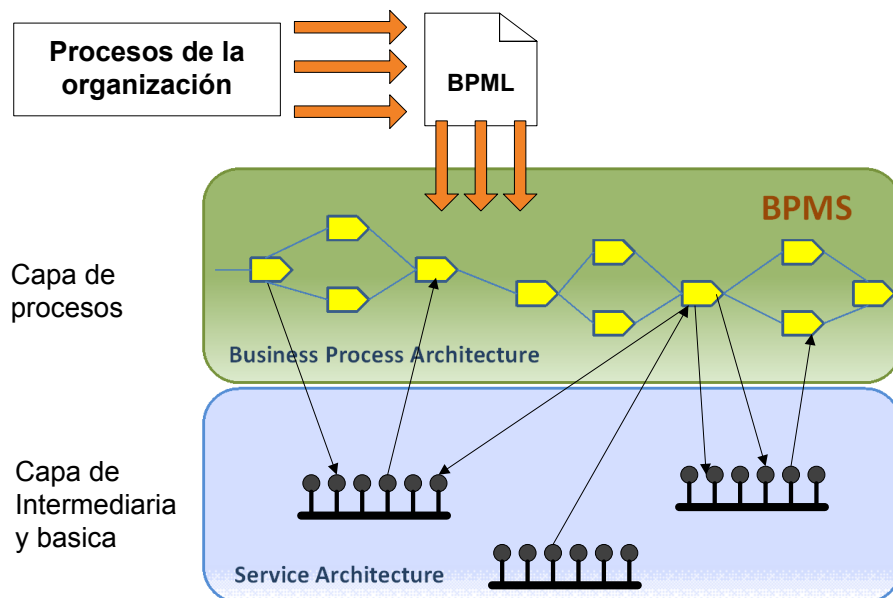


Figura 16: SOA provee la infraestructura para BPM adaptada de [Kraf05] [Endr04]

A continuación se mostrar una comparación entre SOA y BPM

Criterio	BPM	SOA
Enfoque	Disciplina de gestión de los procesos en los que los procesos empresariales se consideran como activos a ser gestionado, diseñado y mejorado continuamente para mejorar la agilidad empresarial y desempeño operacional. Alto nivel de gestión, conjunto de de disciplinas y principios.	Estilo arquitectónico que es modular, distribuibles, compartibles y con poco acoplamiento. Conjunto de principios técnico.
Gestionabilidad	Guía a los administradores a identificar los procesos que necesita cambiar más a menudo e identificar las fuerzas del cambio.	Permite la identificación y categorización de servicios, aprovisionamiento y despacho, así como monitoreo y seguimiento.
Agilidad Empresarial	El uso de métodos, políticas, prácticas de gestión y herramientas de software permite administrar y optimizar continuamente las actividades de una organización y los procesos.	Identifican claramente piezas reutilizables de la funcionalidad y mejorar la integración de aplicaciones mediante el uso de la interfaz común de las técnicas y, por lo tanto, la facilidad de mantenimiento de los sistemas.
Flexibilidad	Mayor flexibilidad y agilidad para adaptación al cambio de los procesos.	Reusabilidad de código, aplicaciones y servicios existentes en lugar de tener que reconstruir a partir de cero.

Interoperatividad	El uso de estándares como BPEL reduce costos futuros de integración. Los Modelos de Procesos son independientes de las tecnologías que puedan utilizarse en su despliegue.	Uso de estándares como SOAP, XML, WSDL, etc. permite la interoperabilidad entre aplicaciones y organizaciones.
Uso de recursos	Enfoque a nivel de proceso más no técnico.	Permite una mejor interacción entre recursos computacionales y el intento de reutilizar estos recursos.
Seguridad	Utilización de contratos bien definidos entre los servicios.	Utilización de contratos bien definidos entre los servicios

Tabla 1: Comparación entre BPM y SOA

Como se puede observar en el cuadro ambos enfoques comparten muchas características y beneficios en cuanto a Flexibilidad, Agilidad Empresarial, Interoperabilidad, etc.

Como se señala en la sección 2.2.6 SOA y BPM ofrecen una mejor solución si se complementan, pero para el presenta trabajo nos focalizaremos en el enfoque SOA.

Capítulo 3: Enfoques para el desarrollo de SOA

3.1 El enfoque SOA basado en patrones para e-business

En esta sección se propone un enfoque SOA que “consiste en siete pasos, donde las actividades no son necesariamente lineales y secuenciales, sino exploratorias e iterativas incrementando la funcionalidad y comprensión del equipo a medida que el proyecto avanza” Este enfoque combina patrones de negocio de IBM para e-business con los conceptos SOA, y puede ser utilizado en el proceso de desarrollo existente en la organización, ya que provee agregados para el diseño de SOA, incluyendo artefactos a obtener. Propone un método para el descubrimiento de servicios top-down, y el aprovechamiento de los servicios bottom-up. Utiliza casos de uso del negocio para modelar los procesos de negocio. [Endr04]

1. Descomposición del dominio: en primer lugar se descompone el dominio en su cadena de valores de áreas funcionales, luego cada área funcional se descompone en procesos, sub-procesos y casos de uso del negocio, estos últimos se utilizan para refinar la descomposición del dominio, ubicándolos en sus áreas funcionales. Luego cada área funcional es mapeada en uno o más subsistemas de la arquitectura. Luego se aplican patrones del negocio y de integración de aplicaciones, clasifican y definen formas de interacción entre los elementos identificados.

Ejemplo de patrones de negocio: Self-Service (User-to-business), son las solicitudes que los usuarios realizan al interactúan con una empresa a través de Internet o intranet.

Ejemplo de patrones de integración: Application Integration, es la integración de múltiples aplicaciones y fuentes de datos sin que el usuario directamente invoque a ellos.

2. Meta-modelo de creación de Servicios: plantean que los Casos de Uso del Negocio son buenos candidatos a ser servicios, por lo que la identificación de servicios objetivos parte de éstos. Se identifican los objetivos y sub-objetivos que deben ser cumplidos para alcanzar objetivos de más alto nivel del negocio, asociando los sub-objetivos con servicios requeridos para obtenerlos.

3. **Análisis de subsistemas:** refina los Casos de Uso del Negocio en Casos de uso del Sistema que soportan un proceso del Negocio dado. Se identifican componentes del Negocio y técnicos analizando los flujos de los procesos entre subsistemas para descubrir/identificar componentes del negocio candidatos. Se utilizan los requerimientos no funcionales para encontrar componentes técnicos. Se identifica la funcionalidad requerida para cada componente del negocio, esto es los Casos de Uso del sistema que cada componente debe soportar. Luego se aplican patrones de aplicación para estructurar los servicios del negocio y técnicos necesarios.
4. **Asignación de servicios:** asegura que todos los servicios identificados tengan un componente de implementación asignado. Estos componentes pueden ser creados nuevos o pueden existir como funcionalidad implementada pero no en forma de componente.
5. **Especificación de componentes:** para cada componente identificado se especifican propiedades como reglas, servicios, atributos, otros componentes que usa, puntos de variación por ejemplo requerimientos de configuración.
6. **Estructuración de componentes y servicios utilizando patrones:** se aplican patrones de tiempo de ejecución del negocio, para establecer la estructura necesaria del middleware⁸ para soportar los servicios identificados y la asignación de cada servicio a un nodo lógico del middleware.
7. **Mapeo de realización tecnológica:** se definen los mecanismos de implementación adecuados para los servicios y componentes especificados. Esto refiere tanto a decisiones del tipo realizar el desarrollo desde cero o hacer outsourcing⁹ como solución llave en mano, o intermedio de análisis “implementar, reutilizar, comprar” ya visto, y alternativas que pueden combinar estas opciones, y también refiere a elecciones del tipo, implementar un wrapper¹⁰ de un sistema legado con un Web Service o con un servicio de cola de mensajes. Luego es posible realizar un análisis de ventajas y desventajas de las

⁸ Middleware: Es un software de computadora que conecta componentes de software o aplicaciones para que puedan intercambiar datos entre éstas

⁹ Outsourcing: Contratación de servicios profesionales externos para satisfacer necesidades empresariales específicas.

¹⁰ Wrapper: Es un programa para controlar el acceso a un segundo programa.

decisiones tomadas, y aplicar mapeos de productos para seleccionar software de base e infraestructura necesaria y cubrimiento de las necesidades por cada vendedor.

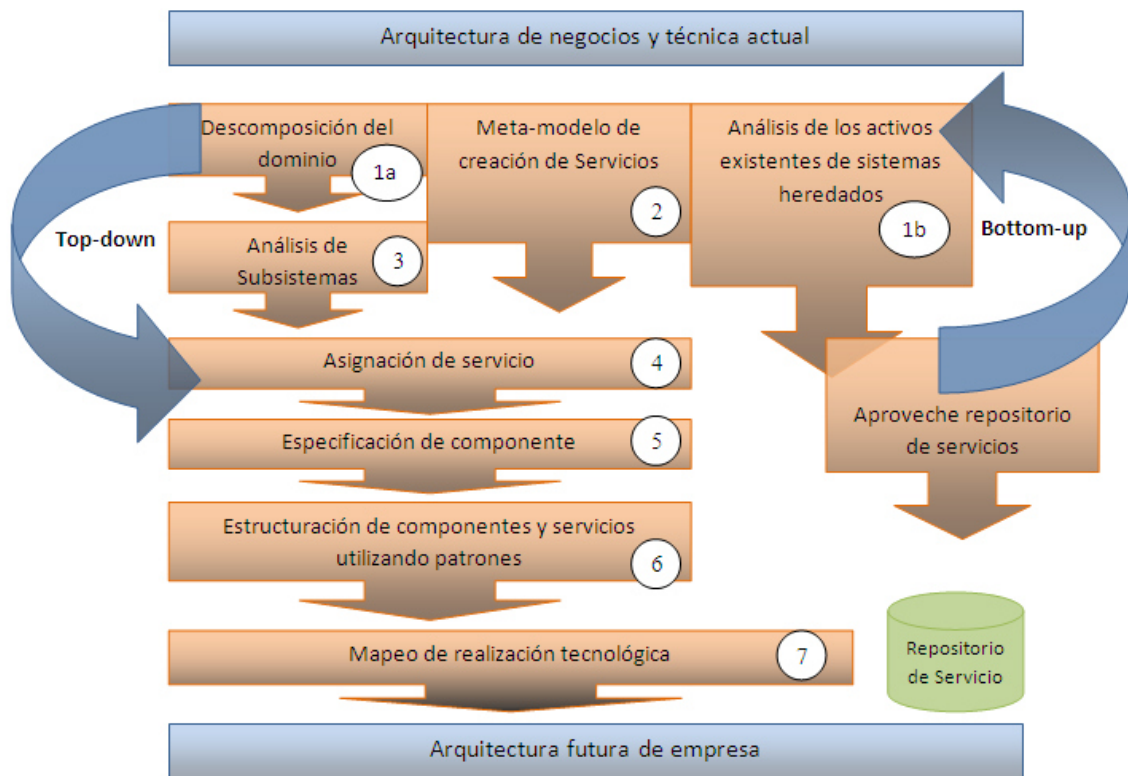


Figura 17: Pasos del enfoque SOA propuesto en [Endr04]

3.2 Enfoque SOA basado en mejores Prácticas de Krafzig Banke, Slama

El enfoque presentado por Krafzig Banke, Slama es una guía que no establece un proceso de desarrollo, sino aspectos a tener en cuenta en el agregado de la orientación a servicios en un desarrollo de software. El primer punto que plantea es referido a los procesos de desarrollo a los cuales es posible agregar el enfoque SOA, describiendo las similitudes y diferencias entre procesos “pesados” y “ágiles”, como el RUP y XP. [Kraf05]

- Dada la naturaleza compleja y frecuencia de cambios en los requerimientos del Negocio en proyectos del tipo SOA, se recomienda que el enfoque elegido sea

del tipo iterativo incremental, fundamental para tratar con la complejidad y con los requerimientos inestables. Además, dada la importancia estratégica de este tipo de desarrollos, es esperable que la metodología elegida tienda hacia el lado de los procesos “pesados”, donde se puedan ir obteniendo varios entregables intermedios en los hitos establecidos.

- Este enfoque plantea incluir diseño de servicios en la definición del proyecto ya que son fundamentales tanto para la Arquitectura como para la planificación del proyecto.
- Un entregable clave de la fase de definición del proyecto debe ser una Arquitectura preliminar, en la que se hayan identificado las aplicaciones frontend, los servicios externos, y los servicios básicos más importantes.
- Además de la identificación de servicios en el diseño se realiza también la clasificación o categorización de los mismos, como guía para la definición de los mismos.
- Para el diseño e implementación se propone la aplicación del modelo “hilo fino” (thin thread) de desarrollo, que aplica el enfoque de desarrollo iterativo incremental de aplicaciones. Un hilo representa un corte vertical del sistema completamente funcional expandiendo todas las capas del sistema desde el frontend al backend. Los contratos de los servicios involucrados en un hilo fino definen el alcance y guían el desarrollo y el testing. Los hilos que se construyen con funcionalidad básica se van “engrosando” en fases siguientes hasta cumplir completamente con los requerimientos definidos para ese servicio, donde se van desarrollando distintos hilos en paralelo.
- Para la gestión de proyecto el enfoque propone basarse en los contratos de los servicios. Porque los servicios tienen el nivel ideal de granularidad y orientación a servicios para servir como base para guiar los aspectos más importantes de un proyecto, incluyendo: estimación de tiempos y costos, planificación de iteraciones y desarrollo, sincronización del desarrollo, planificación del testing e implantación.
- Para la gestión de configuración se recomienda mantener los servicios en diferentes proyectos dentro del repositorio de configuración, ya que éstos serán

compartidos por varias aplicaciones, por lo que es importante mantener su independencia de versiones también.

- Para el testing de servicios se propone que la unidad de testeo sea el servicio, realizando sobre éste tanto testeo funcional como de carga. En primer lugar para el testing funcional, testeo unitario sobre cada servicio para asegurar un primer cumplimiento de funcionalidades, luego testeo de integración sobre varios servicios que deban interactuar, y finalmente testeo del sistema de extremo a extremo, para lo cual es posible introducir herramientas que automaticen las pruebas funcionales.

3.3 Enfoque SOA de Thomas Erl

El enfoque que presenta Thomas Erl plantea que “proyectos de desarrollo para soluciones orientadas a servicios, en la superficie, son similares a proyectos de desarrollo para aplicaciones distribuidas. Cuando nos adentramos en las capas de la orientación a servicios, sin embargo, encontramos que para construir y posicionar apropiadamente los servicios como parte de SOA, los ciclos de proyectos tradicionales requieren algunos ajustes“. Se propone un proceso de desarrollo que consta de seis fases. [Erlt05]

1. Fase de análisis orientado a servicios, los objetivos principales son establecer las capas de servicios, y modelar servicios individuales como servicios candidatos que componen una SOA preliminar. Como objetivos específicos se plantean lo siguiente:
 - La definición de un conjunto preliminar de operaciones de servicios candidatas.
 - La agrupación de estas operaciones candidatas en contextos lógicos que representarán los servicios candidatos.
 - La definición de los límites preliminares de los servicios para que no se superpongan con servicios existentes o planificados.
 - La identificación de lógica encapsulada con potencial de rehusó.
 - La evaluación de que el contexto de la lógica encapsulada es apropiada para el uso definido.

- La definición de modelos de composición preliminares conocidos.

La fase de análisis se compone de tres pasos:

- 1.1 Definir los requerimientos de automatización del negocio: los requerimientos del negocio deben permitir la definición de un proceso de automatización de alto nivel. La documentación de este proceso del negocio será usada como punto de entrada para el proceso de modelado de servicios del paso 3.
- 1.2 Identificar sistemas de automatización existentes: la lógica de aplicación que ya se encuentra, de alguna manera, automatizando cualquiera de los requerimientos identificados en el paso 1 debe ser identificada. En general este paso tiende a soportar esfuerzos de modelado de soluciones orientadas a servicios de larga escala.
- 1.3 Modelar servicios candidatos: en este paso se identifican operaciones candidatas que se agrupan en contextos lógicos que pueden dar lugar a servicios candidatos. La razón de distinguir que son candidatos es debido a que luego en el diseño son sujetos a las realidades de la arquitectura técnica en la que deben residir. Al incluir restricciones, requerimientos, limitaciones específicas al ambiente de implementación, el diseño final de un servicio puede ser bastante distinto del candidato original. Este es el paso más importante en la fase de análisis y se divide a su vez en doce subpasos que guían el proceso de análisis hacia sus objetivos.

2. Fase de Diseño, se definen los siguientes objetivos:

- Determinar el conjunto núcleo de extensiones arquitectónicas.
- Establecer los límites de la arquitectura.
- Identificar los estándares de diseño requeridos.
- Definir el diseño de las interfaces abstractas de servicios.
- Identificar las composiciones de servicios potenciales.
- Evaluar el soporte a principios de orientación a servicios.
- Explorar soporte para características tecnológicas de la implementación de SOA (específicamente con Web Services).

La fase de diseño se compone de cinco pasos, los que a su vez se dividen en subpasos para alcanzar los objetivos planteados:

- 2.1 Componer SOA: se eligen las capas de servicios que serán utilizadas, así como los estándares y tecnologías para la implementación de los servicios (específicamente soporte para XML y WS), y extensiones que se utilizarán (específicamente de las especificaciones WS-* para seguridad, encriptación, transaccionalidad, entre otros)
- 2.2 Diseñar servicios del negocio centrados en entidades: estos servicios representan la capa de servicios menos influenciada por otras. Debe representar las entidades de datos definidas en los modelos del negocio. Se diseñan para ser reutilizados por cualquier aplicación que necesite acceder o manejar información asociada con una entidad en particular.
- 2.3 Diseñar servicios de aplicación: estos servicios representan servicios responsables de realizar procesamientos requeridos por las capas de negocio y orquestación. No se requiere conocimiento del negocio para su diseño, ya que son una abstracción pura orientada a servicios del ambiente tecnológico de una organización. Están sujetos a cambios constantes según las actualizaciones o cambios de la tecnología y la lógica relacionada es construida o alterada.
- 2.4 Diseñar servicios del negocio centrados en tareas: el proceso para diseñar servicios centrados en tareas usualmente requieren menos esfuerzo que los dos anteriores, ya que el rehusó no es una consideración primaria. Solamente las operaciones candidatas de servicios identificadas en el análisis se tratan en estos servicios.
- 2.5 Diseñar procesos del Negocio orientados a servicios: consiste en interpretar apropiadamente los requerimientos de los procesos del negocio que se han recolectado e implementarlos acertadamente. Hay que tener en cuenta las posibles variaciones de las actividades de los procesos, esto significa entender no sólo que puede ir mal, sino como el proceso debe responder a condiciones no esperadas o anormales. En este paso se diseñan los detalles del workflow, así como la interface del servicio de procesos.

3. Fase de Desarrollo de servicios, se introducen las características específicas de las plataformas, sin importar el tipo de servicio. Específicamente, la elección del lenguaje de programación y del ambiente de desarrollo determinarán la forma física que toman los servicios y los procesos del negocio orquestados, de acuerdo al diseño definido.
4. Fase de Testing de servicios, se indica que dada la naturaleza genérica y el potencial para ser reutilizados y compuestos en distintas situaciones, se requiere que los servicios pasen por un testing riguroso antes de su pasaje al ambiente de producción. Sin embargo, no se introducen subpasos ni actividades, se brinda una checklist con preguntas para guiar el testing, como por ejemplo, que tipo de consumidores de servicios podrían potencialmente acceder a un servicio, entre otras.
5. Fase de Deployment de servicios, se deben instalar y configurar los componentes distribuidos, las interfaces de servicios y cualquier otro producto de middleware asociado, en los servidores de producción. Se brinda una checklist de guía que incluye preguntas como por ejemplo, cómo serán distribuidos los servicios, y se incluyen ejemplos en las plataformas.
6. Fase de Administración de servicios, ocurre luego de que se ha realizado el deployment de servicios. Son similares en naturaleza a la administración de aplicaciones distribuidas, basadas en componentes, excepto que se deben aplicar a los servicios como un todo. No se incluyen subpasos ni actividades específicas, y se brinda una pequeña checklist a modo de ejemplo sobre los temas a considerar.

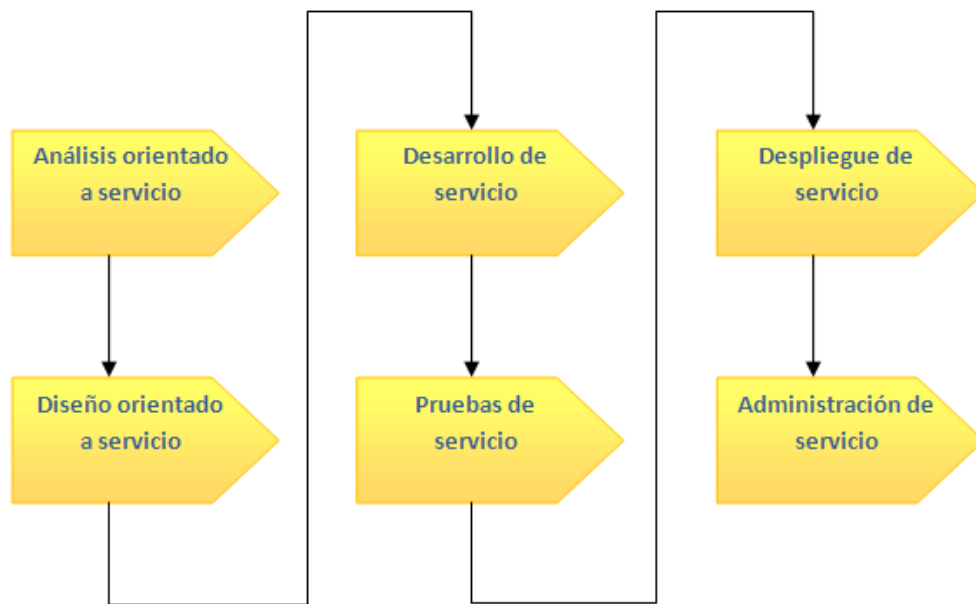


Figura 18: Fases del proceso de desarrollo propuesto en [Erlt05]

3.4 Caso de Existo de aplicaciones SOA

Caso Yansha [Yansha]

Reto empresarial

Con el fin de mantener su posición de liderazgo en el mercado, la empresa china Yansha que posee una cadena de centro comercial, requiere nuevos medios necesarios para aumentar su competitividad frente a los minoristas locales y los nuevos competidores extranjeros en un contexto cada vez más desregulado la industria minorista de China. Yansha vio que la mejor manera de avanzar era a transformar la manera de trabajar con sus socios de la cadena de suministro mediante la adopción de nuevos procesos de negocio, la automatización y la inteligencia de negocios.

Antecedentes

La industria minorista en el vasto mercado chino es muy diferente a la de Occidente. Los minoristas han sido históricamente muy regionalizados, y el conjunto del mercado ha estado prácticamente cerrado a la competencia de empresas extranjeras. Con el rápido crecimiento de la economía china y la relajación de la normativa, que está cambiando rápidamente. Las fusiones y adquisiciones están en aumento y conducen a la consolidación de la industria y la expansión más allá de las fronteras regionales, y los minoristas de China ahora debe compartir el mercado con compañías globales.

A esto se suma a una presión de la competencia sin precedentes. Yansha es una marca de lujo, con ingresos de miles de millones de yuan al año y 215,000 pies cuadrados de espacio comercial en su Youyi Shopping en la ciudad de Beijing.

Yansha desde hace tiempo muestran el liderazgo de la industria. Inaugurado en 1992. En el 2000, implementó IBM Business Partner ONE eFuture del POS-suite ERP, dotándolo de nuevos niveles de eficiencia interna. En el 2003, la suite ERP fue actualizado con capacidades SCM.

Problemática

Con el rápido ritmo de cambio en el paisaje minorista, Yansha sabía que tenía que hacer algo más para mantener su posición de liderazgo. Ante la perspectiva de tener que competir con los muy eficientes competidores extranjeros, Yansha da cuenta de que había de optimizar su cadena de suministro y mejorar la eficiencia entre sus más de 1.800 locales e internacionales, proveedores, y que la tecnología era la manera de

hacerlo. Esto representa un enorme cambio radical para Yansha, que al igual que sus competidores locales, los negocios se siguen haciendo manualmente en su mayor parte, que confían en el papel de los procesos basados en la interacción por teléfono y fax.

Yansha enfrenta dos problemas fundamentales en su esfuerzo por optimizar su cadena de suministro: en primer lugar, para simplificar y automatizar sus procesos de negocio, y en segundo lugar, de encontrar una forma de llegar a todos sus proveedores a comprar en una nueva y más eficiente manera de hacer negocios.

A pesar de la aplicación de un conjunto de empresas, una serie de obstáculos de la continua dependencia de los procesos manuales, la naturaleza de muchos de sus sistemas y la falta de integración con los socios de la cadena de suministro-Yansha, obstaculizan la capacidad para realizar un seguimiento e integrar la información en tiempo real y generar acciones concretas de inteligencia de negocios de la misma. Más de 30 procesos clave de la cadena de suministro se manejan manualmente. El resultado fue la baja productividad, los altos índices de error y los datos inexactos de negocios, todo lo cual lleva a una reducción de la competitividad.

Solución

Para transformar sus procesos con mayor conocimiento para la toma de decisiones, Yansha colaboró con el Laboratorio de Investigación de IBM China (CRL) y eFuture para desarrollar Blue Motor, un proceso impulsado por SCM plataforma diseñada específicamente para el minorista. Blue Motor incorpora tanto la automatización de los procesos de negocio (tales como órdenes de compra, notificaciones de envío, facturación, el pago y la devolución de la mercancía) y la nueva funcionalidad basada en IBM WebSphere Process Server ® e IBM DB2 ® middleware.

Mediante la integración de la información de los sistemas existentes con los nuevos procesos automatizados a través de una arquitectura orientada a servicios (SOA), Blue Motor logra varios objetivos clave:

- Automatiza la gestión de la cadena de suministro relacionados con los procesos de negocio clave entre las personas, a través de múltiples aplicaciones existentes, y entre sus proveedores y Yansha.
- Mejora la visibilidad en tiempo real a través de la supervisión de los procesos de negocio, la generación de los indicadores clave de rendimiento (KPI)-tales como la tasa de aceptación de la orden y de entrega a tiempo.

- Proporciona los proveedores con mejores servicios de información en línea para que el comportamiento de compra del cliente, las ventas y la tendencia proceso de información transparente, permitiendo a los proveedores ajustar y optimizar sus operaciones para satisfacer la demanda del mercado, exhortándolos a alejarse de las interacciones manuales de este modo mejorar la eficiencia de la cadena de suministro global.

Justificación

Debido a la necesidad de la integración eficiente de la información, procesos y sistemas en toda la empresa y fuera en la cadena de suministro, la adopción de SOA es una parte fundamental del Blue Motor. Proporciona la flexibilidad para crear rápidamente nuevas soluciones (y cambiar los ya existentes) que se basa en necesidad comercial inmediata. SOA también permite una mayor interacción con los proveedores basadas en la Web a través de la prestación de servicios, lo que fomenta una mayor capacidad de respuesta. Según el Sr Ai Jie Ma, Director del Departamento Técnico y de Información de Yansha, "Esta solución nos ayudará a construir una plataforma de información junto con nuestros proveedores en su sistema de información interna. El intercambio de datos y nuestra estrecha interacción nos permita responder adecuadamente al mercado".

Beneficios

Con Blue Motor, muchas cosas de Yansha basados en documentación manual de los procesos se han convertido en una cosa del pasado. Las actividades relacionadas con el flujo de trabajo, tales como la emisión de órdenes de compra y control de inventario son empujados directamente a los usuarios apropiados a través de de mensajes móviles, de trabajo basada en navegador y el correo electrónico de activación de alertas cuando se retrasa la actividad de procesamiento, y mejorar considerablemente la ejecución de procesos de negocio. Los usuarios de las empresas también pueden emplear la colaboración de servicios públicos tales como los servicios móviles de mensajes cortos para encontrar a la persona adecuada para resolver los posibles problemas proceso, que ayuda a racionalizar los recursos y evitar la revisión.

Esta tecnología permite a Yansha la visualización de la gestión para obtener una medida exacta, en tiempo real de su cadena de suministro, emitido a través de un "panel" que muestra continuamente actualizado los indicadores clave de rendimiento. Esto, a su vez,

proporciona una base sólida para la toma de decisiones de negocios y optimización de procesos.

Más importante aún, de información en tiempo real permite a Yansha trabajar más eficaz y eficientemente con sus proveedores. El minorista tiene ahora la información que necesita para clasificar con precisión sus proveedores sobre la base de un número de factores, incluyendo su rentabilidad, la venta mensual de las tendencias y nivel de actividad. Esto ha permitido a Yansha elaborar una nueva forma de trabajar con proveedores, basada en los valores y el modelo de fijación de precios de proveedores que le da un trato preferencial a los vendedores que son más valiosos y rentables para la empresa. De esta manera, los proveedores reciben un incentivo para trabajar de manera eficiente: un rendimiento superior es recompensado con un mejor trato.

La disponibilidad de nueva información proporcionada por la solución también ha permitido Yansha dar mayor tasa basada en servicios de información en línea a sus proveedores, que no sólo alienta a adoptar los nuevos métodos, sino que también genera ingresos adicionales para el minorista.

Capítulo 4: Resolución del problema aplicando la técnica seleccionada

4.1 Evaluación y selección de la técnica

En la siguiente sección presentaremos una evaluación de las alternativas de solución y la evaluación de los enfoques para el desarrollo de SOA en la empresa Telefónica Gestión de Servicios Compartidos S.A.C.

4.1.1 Criterios de evaluación:

Los criterios para evaluar los enfoques son:

- Interactivo, dada la naturaleza compleja y cambiante de los requerimientos del negocio de Comercio Exterior y en toda la cadena de suministro de la empresa Telefónica Gestión de Servicios, se recomienda un método interactivo como una forma de lidiar con estos cambios.
- Ligado a Roles, dada la naturaleza de las empresas del grupo Telefónica en el uso de la metodología MEGON, el método debe estar orientado al uso de asignación de roles.
- Reusabilidad de activos de TI, permitir que la técnica pueda utilizar los activos de TI existentes y los nuevos activos de TI.
- Documentación, el método debe permitir la obtención de entregables intermedios para realizar el control en los hitos establecidos, siguiendo la metodología MEGON.
- Utilización de estándares, la solución no debe estar ligado a una tecnología propietaria.
- Clasificación de elementos, el método debe contemplar la clasificación de los elementos que componen una SOA, en este caso clasificación de servicios.

4.1.2 Selección de la técnica usada:

Se presenta a continuación un cuadro comparativo donde los tres enfoques descritos en el capítulo 3, se evalúan de acuerdo a los criterios planteados en la sección 4.1.1.

Criterios de evaluación.	Enfoque basado en patrones de negocio de IBM	Enfoque basado en mejores prácticas, Dirk Krafzig, Karl Banke, Dirk Slama	Enfoque de Thomas Erl,
Interactivo	Plantea siete pasos con actividades exploratorias e interactivas.	Recomienda un proceso interactivo e incremental.	Plantea fases en forma secuencial.
Roles	No describe ningún rol ni responsabilidades.	No manifiesta el uso de roles en ninguno de sus aspectos a tener en cuenta para el desarrollo de SOA.	No se indican roles intervinientes en ninguna de las fases.
Reusabilidad de activos de TI	No plantea ninguna actividad para la búsqueda o clasificación de servicios existentes	Recomienda distinguir los servicios construidos desde cero, modificaciones y extensiones de servicios existentes.	Plantea la identificación de sistemas de automatización existentes.
Documentación	No provee ninguna guía, ni se describen las entradas y salidas en cada una de las actividades.	Propone guías para los aspectos de diseño de servicios incluyendo la clasificación de servicios.	No se indican los artefactos de entrada y salida de las actividades, tampoco se brinda plantillas.
Utilización de	Utiliza como el RUP	Propone seguir un	Menciona la

estándares.	Casos de Uso de negocio para definir los procesos del negocio. Plantea que son buenos candidatos a ser servicios.	proceso con características similares a los procesos pesados como el RUP.	necesidad de modelar los procesos, pero no como se realizara este modelado, esto es con BPMS, Diagramas de Actividad de UML, Casos de Uso, etc.
Clasificación de Servicios	No propone ninguna clasificación para los servicios.	Recomienda se realice una identificación, clasificación de servicios.	Si propone una clasificación de servicios que además servirán para guiar el proceso de desarrollo.

Tabla 2: Cuadro comparativo de enfoques SOA

Comentarios adicionales al cuadro comparativo

Enfoque basado en patrones de negocio de IBM

- En el criterio de utilización de estándares, el enfoque plantea que los casos de uso de negocio son buenos candidatos a ser servicios, pero esto sería solo aplicable a servicios centrados en proceso ya que los servicios que luego se orquesten deben ser de menor granularidad.
- La aplicación de distintos patrones puede agregar complejidad al desarrollo ya que se debe capacitar a los involucrados para su correcta aplicación.

Enfoque de Thomas Erl

- En el criterio de Interactivo, el enfoque plantea un estilo en cascada, si bien no plantea vueltas de feedback a las fases anteriores, podría realizarse con algunas adaptaciones.

- En el criterio comparativo de clasificación de servicios, el enfoque plantea una clasificación de servicios que definirá el proceso de la fase de diseño lo hace que el proceso sea muy específico.

Resultado de la evaluación

Como se puede ver en el cuadro comparativo los tres enfoques cumplen algunos de los criterios establecidos, unos más que otros como es el caso de segundo enfoque orientado a mejores prácticas por Dirk Krafzig, Karl Banke, Dirk Slama. Sin embargo este enfoque no propone un proceso de desarrollo para realizar estas prácticas.

Por consiguiente se propondrá un método para seguir estas prácticas, además se tomaran en cuenta los aportes de los otros dos enfoques, como por ejemplo la utilización de Casos de Uso para describir los procesos de negocio propuesto por el enfoque de patrones de negocio de IBM.

4.2 Adaptación o aplicación de herramienta teórica para resolver el problema

En esta sección se presentara el método propuesto que se deberán seguir para la adaptación de la alternativa selecciona al problema, estos proceso han sido adaptados en base a los enfoques anteriores, tomando en consideración sus aportes e incluso proponiendo una alternativa a sus desventajas.

4.2.1 Contexto de la propuesta

Un método para el desarrollo orientado a servicios no requiere necesariamente una nueva metodología de trabajo para proyectos en una organización, si no que se puede construir sobre el que siga la organización, agregando actividades y artefactos específicos para el desarrollo orientado a servicios, para nuestro caso de estudio Telefónica Gestión de Servicios Compartidos S.A.C. utiliza el Megon como marco de trabajo para cualquier tipo de proyecto, al cual hasta le fecha se han realizados adecuaciones según evolucionan los requerimientos, sin embargo la base del megon se divide en tres bloques que se muestran en la figura 20.

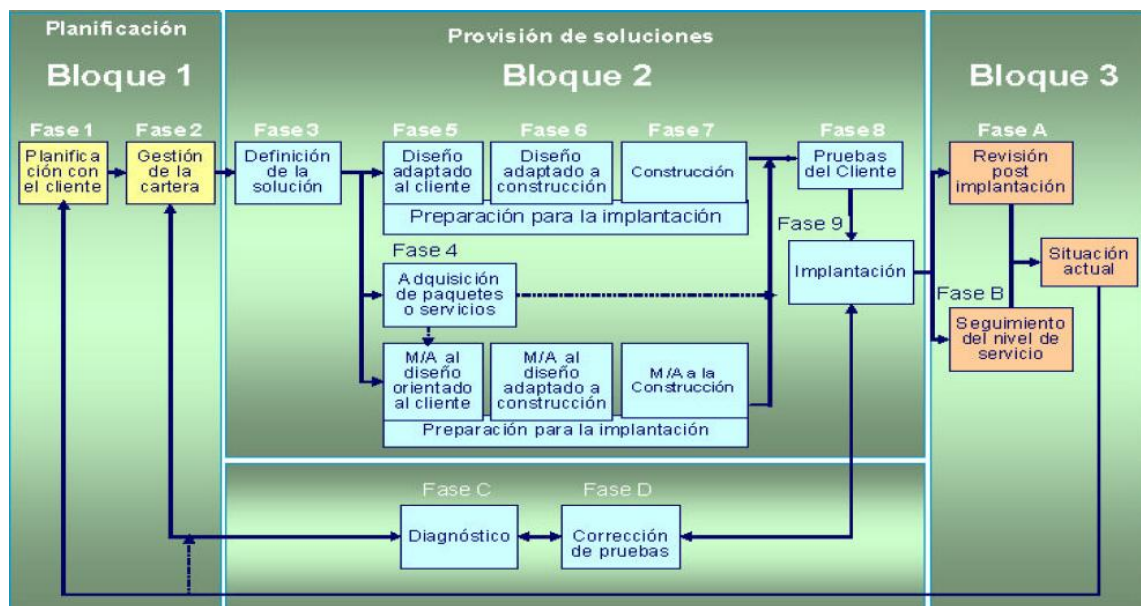


Figura 19: Bloques y fases Megon [Cast08]

Siguiendo el enfoque de mejores prácticas, se propone que se siga un enfoque interactivo e incremental para lidiar con los cambios en la naturaleza de los negocio, este enfoque será enmarcado dentro del bloque 2 ("Provisión de Soluciones") del Megon.

Proceso como el Rational Unified Process (RUP) presentado en la sección 2.2.1.3 es clasificado como "proceso pesado" y eXtremme Programming (XP) presentado como referencia en la sección 2.2.1.3 clasificado como "proceso liviano", ambos presentan el enfoque interactivo e incremental, que resulta apropiado para el desarrollo orientado a servicios.

Dada la importancia estratégica de asumir un proyecto de desarrollo con enfoque SOA donde los conceptos asociados al enfoque son novedosos y poco conocidos, resulta más apropiado un proceso base más del tipo de los "pesados", de forma de ir obteniendo productos claves en el avance del mismo para ir controlando el desarrollo del proyecto y la aplicación del método propuesto.

En base a lo planteado se decide entonces adoptar el proceso de desarrollo RUP para la adaptación en la Metodología de trabajo de tgestiona para proyectos de sistemas basados en SOA. Por tanto se estaría cumpliendo la primera práctica del enfoque seleccionado [Kraf05].

Tanto RUP como Megon comparten algunas características como por ejemplo la asignación de Roles y Entregables, esto facilitara al método propuesto.

Para incluir desarrollos con enfoque SOA dentro del proceso RUP se requiere la realización de actividades definidas en las disciplinas del proceso base del RUP, agregando las disciplinas y actividades para SOA, así como los roles involucrados y entregables específicos,

4.2.2 Proceso base del RUP Adaptado.

El proceso de desarrollo base es principalmente una adaptación del RUP que mantiene las características más importantes del mismo mencionadas en la sección 2.2.1.3, es iterativo e incremental, basado en Casos de Uso y centrado en la Arquitectura. Se modela como el RUP en dos dimensiones: el tiempo y las disciplinas.

Dimensión del tiempo

En la dimensión del tiempo se mantienen las cuatro fases: Inicial, Elaboración, Construcción y Transición.

Dimensión de las disciplinas

En la dimensión de las Disciplinas se definen las disciplinas tradicionales del desarrollo de software: requerimientos, diseño, implementación y verificación, y configuración e implantación. En cada disciplina se definen actividades, roles encargados y participantes de las actividades, y entregables de entrada y salida de dichas actividades.

En cada disciplina se mantienen sus actividades principales pero se eliminan algunas que no aplican y se agregan otras que tienen que ver específicamente con el enfoque SOA.

- Disciplina de Requerimientos, las Actividades que se mantienen son el Modelo de Casos de Uso y la realización del Modelo de Dominio.
- Disciplina de Diseño, se mantiene la Definición de la Arquitectura de Software, el Diseño de Casos de Uso y del Modelo de Datos.
- Disciplina de Implementación, se mantiene la implementación de Clases y Subsistemas definidos.

- Disciplina de Verificación, se mantiene la realización de pruebas Unitarias, de Integración y del Sistema sobre cada liberación en cada iteración y la planificación de la verificación.

Se definen también entregables principales, para los que se establecen los roles participantes y un rol responsable por la generación del entregable, y se brinda además una plantilla del contenido asociado como guía para la realización del mismo.

Se definen roles básicos que son: Analista, Arquitecto, Especialista Técnico, Implementador, Responsable de Verificación, Responsable de SQA, Documentador, los cuales tienen asociada además una descripción de tareas y perfil para cumplirlo.

Para el caso de tgestionona este sería su equivalente:

RUP	MEGON (tgestionona)
Analista	Analista (Gerencia de Proyectos)
Arquitecto	Arquitecto (Gerencia de Producción e Infraestructura)
Especialista Técnico	Especialista Técnico (Gerencia de Proyectos o Gerencia de Producción e Infraestructura)
Implementador	Desarrollador (Gerencia de Proyectos)
Responsable de SQA.	Testing (Gerencia de Producción e Infraestructura)
Documentador	Ofimático

Tabla 3: Cuadro de roles en RUP y Megon (tgestionona)

Además se define una combinación de roles según la carga de trabajo en el tiempo y la compatibilidad de tareas combinadas, por ejemplo: el Arquitecto/Asistente de Verificación es solamente Arquitecto hasta finales de la Fase de Elaboración, y en la Fase de Construcción también pasa a ser Asistente de Verificación para aprovechar su conocimiento del producto en desarrollo en la búsqueda de errores. A cada persona en el

equipo se le asigna un rol combinado con indicación en el tiempo de qué actividades correspondientes a qué rol siempre debe realizar en cada Fase e iteración.

4.2.3 Extensión del proceso RUP Adaptado para desarrollo SOA

El proceso de extensión para el desarrollo SOA implica seguir de forma similar al proceso base en cuanto a desarrollo interactivo e incremental, guiado por Casos de Uso y centrado en Arquitectura. Además se ejecutan las actividades de la extensión SOA.

- En primer lugar se agrega una disciplina que está en el RUP pero no en el proceso base, que es el Modelado del Negocio. Como se mencionó un aspecto principal en una SOA es el modelado de los procesos de la Organización como forma de conjuntar los aspectos del negocio con las soluciones informáticas provistas.
- En segundo lugar se agregan cinco actividades en la disciplina de diseño, que son las actividades claves del enfoque, en las que se identifican, categorizan, reutilizan, especifican, y definen los servicios de la aplicación, y la orquestación de los mismos para proveer los procesos del negocio modelado. Por tanto se estaría cumpliendo una segunda práctica del enfoque [Kraf05].
- En tercer lugar se agregan en la disciplina de Implementación la actividad de implementación de servicios según fueron definidos en la etapa de diseño. Por tanto se estaría cumpliendo una tercera práctica del enfoque [Kraf05].

Además para cada actividad se definen los roles responsables, participantes y los entregables entrada y salida de las mismas.

Como entregables se definen dos para la Disciplina Modelado del Negocio, el documento de Evaluación de la Organización Objetivo y el Modelo de Casos de Uso del Negocio, y uno para la Disciplina Diseño, el Modelo de Servicios, que se compone de varias secciones donde realizar la definición, categorización, especificación, etc.

Entre los roles definidos los más importantes para la Extensión SOA son: el Arquitecto, quién tiene responsabilidad en el área de Diseño pero participa también en el

relevamiento de requerimientos y como coordinador del desarrollo junto con el Jefe de Proyectos; el Analista, quién principalmente participa en el relevamiento de requerimientos; el Especialista Técnico, quién tiene a su cargo el estudio e investigación de las tecnologías disponibles a utilizar y el Implementador o desarrollador, quién tiene a su cargo la codificación de la aplicación.

Esta extensión del proceso RUP adaptado para SOA se incursionara en bloque 2 de la Metodología Megon.

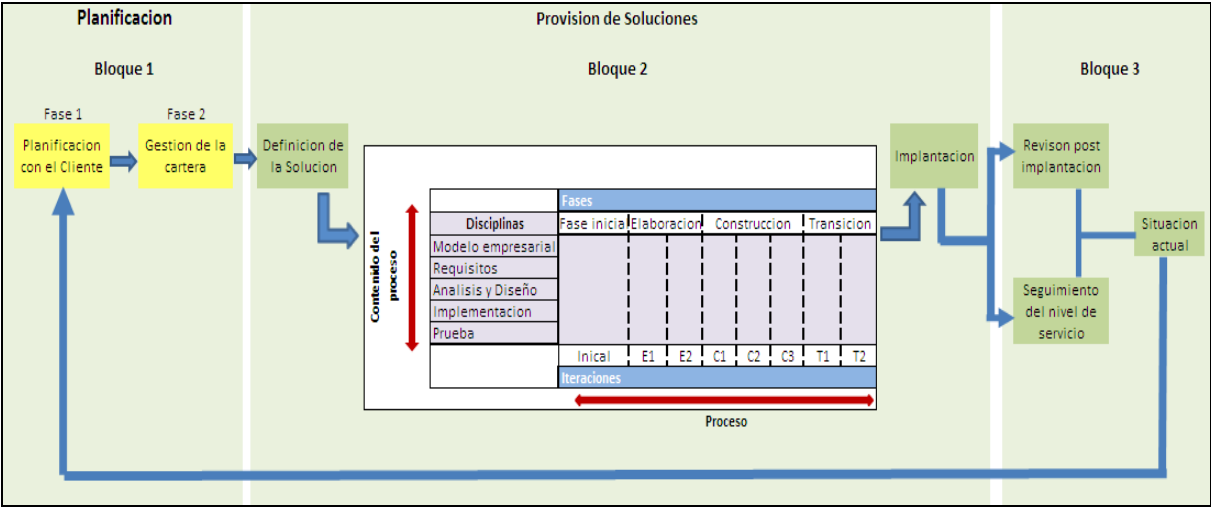


Figura 20: Bloques y fases Megon adaptado para SOA

4.2.3.1 Disciplinas para el proceso de adopción de SOA

A continuación se muestran los elementos de modelado correspondiente a las Disciplinas, actividades, entregable y roles definido en el método propuesto.

1. Disciplina de Modelado de Negocio

El propósito de esta disciplina es obtener un mapa de la organización y sus procesos para ganar un mejor entendimiento de los requerimientos de la aplicación en desarrollo, El modelado de negocio es realizado en la Fase Inicial. Los entregables en esta disciplina son la evaluación de la organización Objetivo que contiene aspectos claves de la misma y el modelado de casos de uso del negocio donde se especifican los procesos identificados en la organización.

Las actividades que se describen en esta disciplina son: Evaluar la Organización Objetivo, Identificar los procesos de negocio.

1.1. Evaluar a la Organización Objetivo

Objetivo: Involucrar al equipo de proyecto con la organización para la cual se está realizando el desarrollo.

Descripción: Se debe describir el estado actual de la organización en la cual la aplicación será implantada, en términos de sus procesos actuales, herramientas, competencias de las personas, clientes, desafíos tecnológicos, problemas y áreas de mejora, Identificando claramente los involucrados en el esfuerzo de modelado del negocio.

Roles: El rol responsable de esta actividad es el Analista y el Arquitecto.

Entregables de entrada: Esta actividad tiene como entrada la información recabada con los clientes del negocio sobre diversos aspectos de la organización, a los que llamamos actas de reunión en la metodología Megon.

Entregable de salida: Como salida se tiene el documento de Evaluación de la Organización Objetivo donde se detallan los aspectos claves revelados.

Justificación: Como se describe en el capítulo 2, es importante conocer el estado del desarrollo de software en la Organización donde se realizará el desarrollo que se comienza, en particular es importante conocer la infraestructura tecnológica con que se cuenta, los sistemas que informatizan sistemas existentes, la estructura organizacional incluyendo las jerarquías en las distintas áreas, el relacionamiento entre las áreas del Negocio y de TI. Este conocimiento que se obtendrá como parte de relevamiento de necesidades de la organización puede ser importante para otro proyecto que no sea SOA. De esta manera se estaría cumpliendo con un criterio “Reusabilidad de activos de TI” establecido en el cuadro comparativo del Capítulo 4.

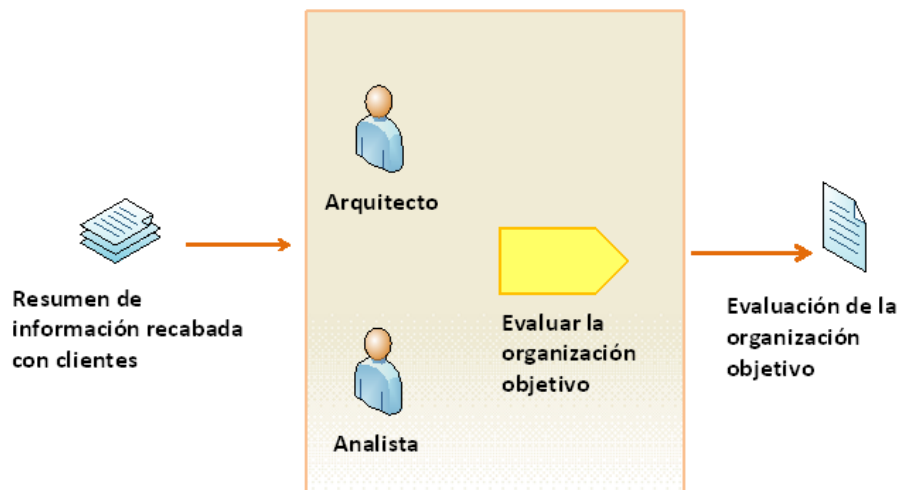


Figura 21 Diagrama de la actividad Evaluar la Organización Objetivo

1.2. Identificar los procesos de negocio

Objetivo: Describir y entender los procesos que se realizan en el negocio en lo que tiene que ver principalmente con la aplicación que se va a desarrollar. Para esto los procesos se describen como caso de uso del negocio con los flujos correspondientes y se modelan con diagrama de procesos.

Descripción: Esta actividad está compuesta en varias sub-actividades que en conjunto permiten identificar los procesos del negocio, describiendo los actores principales y el flujo de ejecución de los mismos en la organización. Es importante definir los límites del negocio y los términos que se manejan en el negocio.

Roles: El rol responsable de esta actividad es el Analista y participa también el rol de Arquitecto.

Entregables de entrada: Esta actividad tiene como entrada el documento de Evaluación de la Organización Objetivo y actas de reuniones realizadas con el cliente.

Entregable de salida: Como salida se tiene el modelo de Casos de Uso del Negocio donde se definen los flujos de los procesos identificados en la organización.

Justificación: Esta actividad se agrega ya que resulta imprescindible para desarrollos con enfoque SOA. Como se describe en la sección 2.2.4.5, los procesos de negocio serán implementados como secuencia de invocaciones a servicios de menos

granularidad, en lo que se conoce como orquestación de servicios en la Organización, o coreografías de servicios entre varias organizaciones. El modelado de estos procesos como Casos de Uso del negocio es una forma de modelarlos según el enfoque basado en patrones de negocio.

Los Casos de Uso del Negocio definidos en base al estudio de los procedimientos de la organización, se analizan para identificar los Casos de Uso del Sistema, que en general serán de mayor granularidad dado que identifican interacciones puntuales de los actores con el sistema, y estas se encuentran inmersas en procesos de mayor duración y que mayormente cruzan las fronteras de departamentos de la organización o incluso de la organización misma.

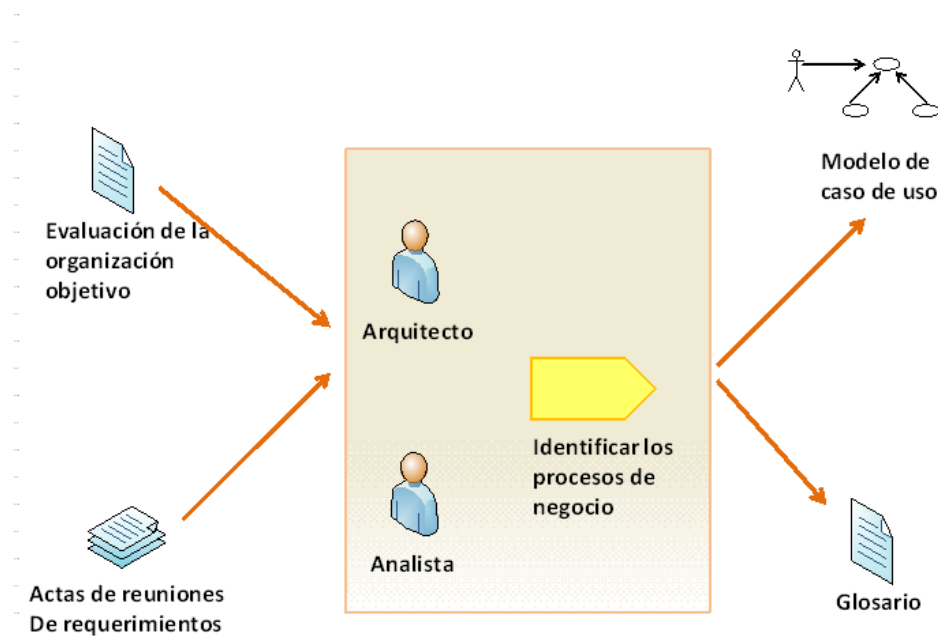


Figura 22: Diagrama de la actividad Identificar Procesos de Negocio

Relación de la disciplina con el proceso RUP

En el proceso base la primer Disciplina de trabajo es la de Requerimientos, en la cual se deben realizar actividades como Especificar los requerimientos y Definir el Modelo de Casos de Uso del Sistema. La Disciplina de Modelado del Negocio, se realiza en forma previa a la de Requerimientos y constituye una entrada fundamental para la misma.

Los Casos de Uso del Negocio definidos en base al estudio de los procedimientos de la Organización, se analizan para identificar los Casos de Uso del Sistema, que en general serán de menor granularidad dado que identifican interacciones puntuales de los actores con el Sistema, y éstas se encuentran inmersas en procesos de mayor duración y que generalmente cruzan las fronteras de Departamentos de la Organización o incluso de la Organización misma.

2. Disciplina de Diseño

El propósito de esta disciplina es identificar y catalogar los servicios necesarios para cumplir con los procesos identificados de la organización en los Casos de Uso del Negocio, definiendo sus interfaces y sus operaciones, así como los componentes que los implementan y la reutilización de otros servicios y componentes.

Un aspecto importante de esta disciplina es definir la secuencia de invocaciones a servicios necesaria para la ejecución de los procesos del negocio identificados como Casos de Uso del Negocio, en una orquestación de servicios si el proceso se encuentra bajo el control de la Organización, o en una coreografía de servicios si el proceso corresponde a más de una Organización.

Las actividades que se describen en esta disciplina son: Identificar y categorizar los servicios, Especificar los servicios, Investigar servicios existentes, asignar servicios a componentes, definir la orquestación de servicios.

2.1. Identificar y categorizar servicios

Objetivo: Identificar los servicios necesarios para realizar los procesos del negocio, clasificándolos según el tipo de servicio: básico, intermediarios, centrados en proceso y públicos (clasificación de [Kraf05]).

Descripción: Teniendo en cuenta la Descripción de la Arquitectura, los Casos de Uso del Sistema y la Lógica, se realiza en primer lugar un análisis de los subsistemas definidos, asegurándose que existe un mapeo de por lo menos un subsistema a cada área funcional del Negocio. De la misma forma se identifican claramente en que subsistema se realiza cada Caso de Uso del Sistema, comenzando con los Casos de Uso identificados como relevantes a la arquitectura. A partir de esta correspondencia se

identifican los servicios que deben ser provistos por cada subsistema para realizar los distintos Casos de Uso del Sistema y seguidamente los Casos de Uso del Negocio.

Roles: El rol responsable de esta actividad es el Arquitecto y participan también los roles de Analista y Especialista Técnico.

Entregable de entrada: Esta actividad tiene como entrada los entregables de Descripción de la Arquitectura, y el modelo de Casos de Uso del Negocio, para contar con la información técnica del software en diseño y los procesos del negocio identificados para la organización.

Entregables de salida: Como salida se tiene el Modelo de Servicios con la sección correspondiente a la identificación y categorización de servicios.

Justificación: Esta actividad resulta imprescindible para desarrollos con enfoque SOA identificar y categorizar servicios, es una de las actividades más importantes, como se describe en los conceptos teóricos presentados en el capítulo 2.

La categorización de los mismos permite ordenar conceptual y jerárquicamente la ubicación de estos servicios en la capa de abstracción de servicios, de forma de guiar el desarrollo y evitar la proliferación de servicios que no tengan la granularidad adecuada.

Un aspecto importante a tener en cuenta en la identificación y categorización de servicios es el bajo acoplamiento que debe existir entre estos, por lo que esta definición debe realizarse con sumo cuidado para permitir obtener servicios auto contenidos que no dependan de otros para realizar sus funciones.

Además se propone hacer uso de la clasificación de servicios realizada por [Kraf05], por ser la más completa. De esta manera se estaría cumpliendo con un criterio “Clasificación de Servicios” establecido en el cuadro comparativo del Capítulo 4.

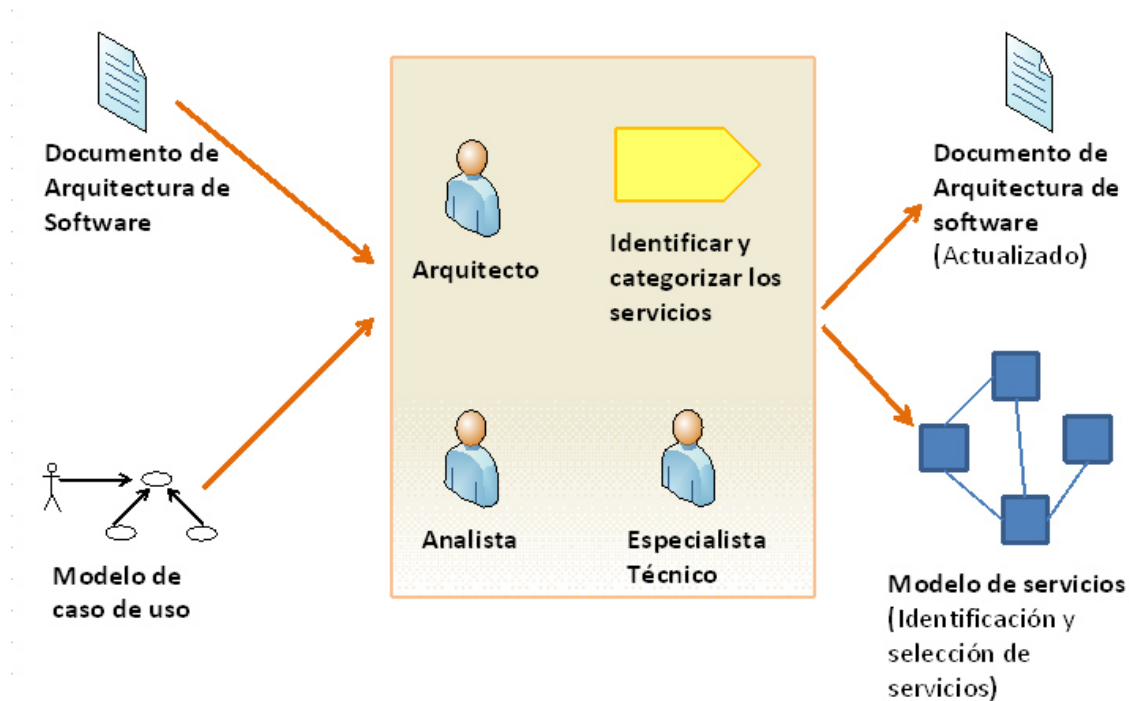


Figura 23: Diagrama de la actividad Identificar Servicios

2.2. Especificar servicios

Objetivos: Especificar los servicios identificados, definiendo los contratos de servicio para cada uno incluyendo las interfaces que brindara y sus operaciones, parámetros, etc.

Descripción: Para cada servicio identificado se realiza el contrato funcional especificando para cada interface definida los métodos que deben ser implementados para proveer el servicio acordado, para cada operación en la interface se debe especificar la información siguiente:

1. Nombre del método
2. Parámetro requerido por el método y para cada parámetro el nombre, tipo y descripción
3. Valor retornado, indicando el nombre, tipo y descripción
4. Lista de excepciones levantadas
5. Breve descripción de la funcionalidad provista
6. Pre-condiciones requeridas para la ejecución de la operación

7. Post-condiciones que serán validas luego de la ejecución del método

Roles: El rol responsable de esta actividad es el Arquitecto y participa también el rol del Analista.

Entregables de entrada: Esta actividad tiene como entrada el Modelo de Servicios.

Entregables de salida: Como salida tiene el Modelo de Servicios con la sección correspondiente a la especificación de servicio informada.

Justificación: Como se describe en los conceptos teóricos presentados en el capítulo 2, la especificación de servicios en los contratos de servicios es una parte fundamental para definir las relaciones entre estos, mediante la identificación de operaciones, parámetros, etc. Por consiguientes es una actividad obligatoria.

La publicación de los contratos por los proveedores de servicios, permitirá que estos sean reutilizados por otras aplicaciones.

La especificación de servicios permitirá abstraer la definición de servicios de los componentes que los implementaran, permitiendo que las implementaciones puedan ser cambiadas sin afectar las interfaces de los mismos.

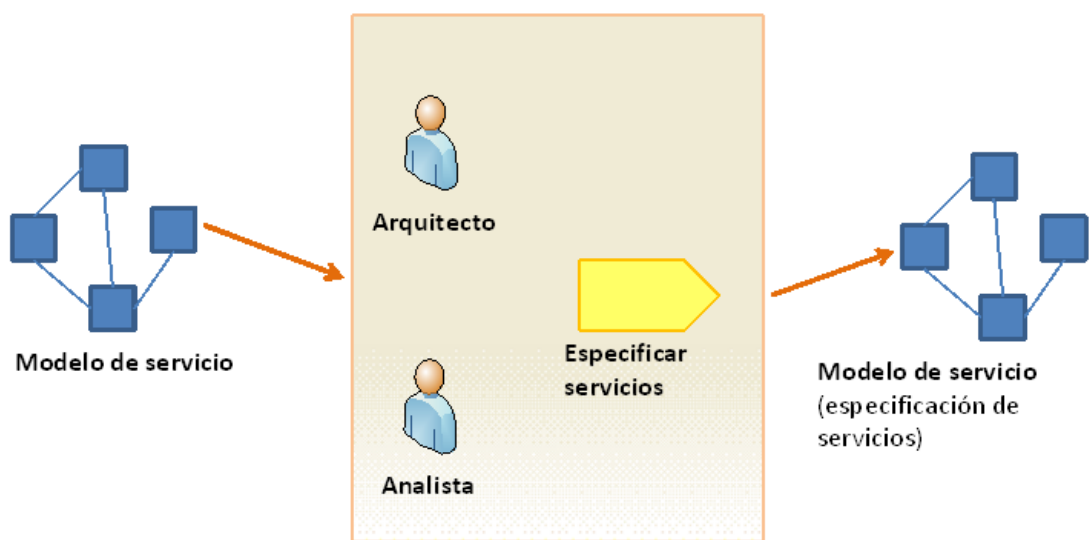


Figura 24: Diagrama de la actividad Especificación de Servicios

2.3. Investigar servicios existentes

Objetivo: Tiene por objetivo encontrar servicios que ya estén implementados en la plataforma SOA y puedan o deban ser reutilizados en la aplicación en desarrollo. En general los servicios que serán reutilizados son los servicios básicos.

Descripción: Una vez que los servicios han sido identificados y especificados, se deben investigar los servicios con que cuenta la organización, tanto básicos como de otros niveles, para reutilizar en la aplicación en desarrollo.

Los servicios básicos como los de acceso a datos, autenticación, seguridad, sesión, etc. deberán ser reutilizados por todos los servicios definidos en niveles superiores de la SOA.

Roles: El rol responsable de esta actividad es el Arquitecto y el Analista.

Entregables de entrada: Esta actividad tiene como entrada la Evaluación de la Organización Objetivo, donde están las capacidades existentes en la organización.

Entregables de salida: Como salida se tiene el Modelo de Servicios correspondientes a los servicios existentes.

Justificación: Como se describe en los conceptos teóricos presentados en el capítulo 2, una vez que la organización ha aplicado en forma sistemática el enfoque a servicios para la informatización de sus procesos del Negocio, se contara con servicios de granularidad fina y gruesa que proveen funcionalidades relacionadas con el Negocio de la Organización.

Estos servicios ya existentes deben ser tenidos en cuenta para nuevos desarrollos, de forma de reutilizar el diseño e implementación realizada en otros desarrollos previos, evitando la duplicación de definiciones y código para las mismas funcionalidades.

Eventualmente se podrían investigar también servicios existentes en otras organizaciones para ser incorporados en coreografías de servicios.

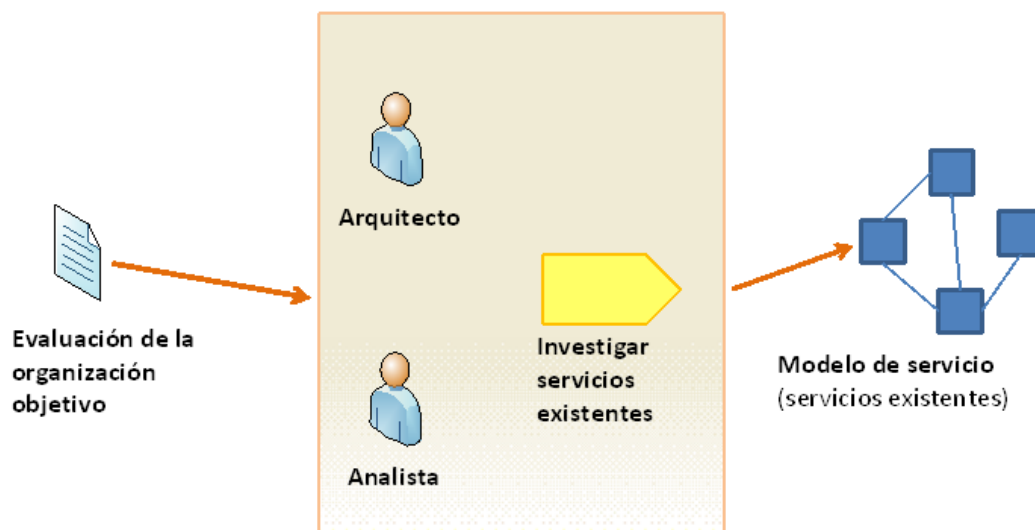


Figura 25: Diagrama de la actividad Servicio Existentes

2.4. Asignar servicios a componentes

Objetivo: Definir los componentes que deberán ser implementados para proveer los servicios especificados.

Descripción: Para asignar servicios a componentes se debe responder la pregunta de quién proveerá la implementación y gestión de los servicios definidos según las interfaces especificadas. Se debe tener en cuenta los servicios existentes identificados si existen componentes que los brinden o funcionalidades existentes para las cuales se pueda crear algún servicio intermediario que las provea. Para cada componente necesario se realiza la especificación del componente, indicando regla del negocio y servicios que implementa, así como otros componentes que utiliza.

Roles: El rol responsable de esta actividad es el Arquitecto y el Especialista Técnico.

Entregables de entrada: Esta actividad tiene como entrada el Modelo de Diseño y el Modelado de Servicios, con la información del software diseñado y los servicios identificados, categorizados, especificados y existentes.

Entregables de salida: Como salida se tiene el Modelo de Implementación y el Modelo de Servicios, con la asignación de componentes a servicios correspondientes en cada caso.

Justificación: Esta componentización de servicios estará dada por el entorno de programación que se elija, y puede ser clase Java, componentes J2EE o .NET, componentizaciones de sistemas legados, procesos definidos en Business Process Modeling Language (BPML) específicamente en BPEL, entre otros. El enfoque fundamental es que la definición de servicios está separada de los componentes que lo implementan permitiendo la integración de diferentes tecnologías.

Esta actividad constituye otro aspecto importante del enfoque a servicios planteado por la metodología SOA propuesta, y que fuerza en forma explícita la separación entre la definición de los servicios y la implementación de los mismos.

Es el cuarto paso enfocado en servicios que se propone en el método, que prescribe la asignación de componentes a servicios para su implementación, aportando a la separación entre los elementos del Negocio que deben realizar los servicios definidos, y las plataformas tecnológicas que se utilicen en la Organización.

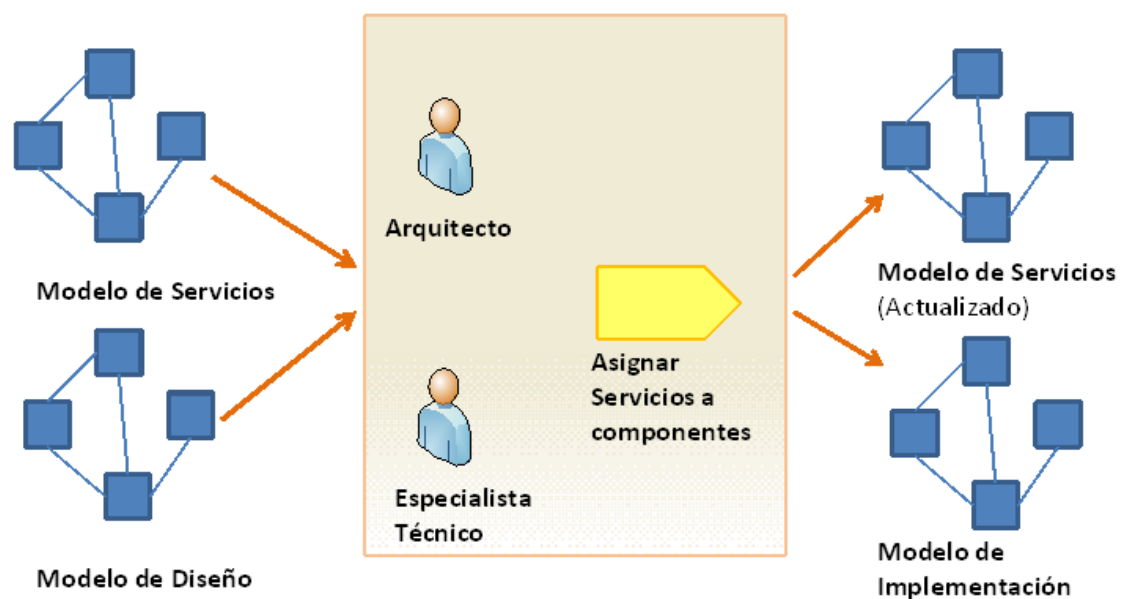


Figura 26: Diagrama de la actividad Asignar servicios a componentes

2.5. Definir orquestación de servicios

Objetivo: Definir la secuencia de interacción entre servicios que es necesaria para realizar los procesos del negocio identificados y descritos como Casos de Uso del Negocio.

Descripción: La orquestación de servicios para la implementación de procesos del Negocio se muestra para cada proceso en un diagrama de secuencia que describa la interacción entre los distintos servicios involucrados. Además se utiliza preferiblemente una herramienta BPMS (Business Process Management System) para definir, ejecutar y gestionar la orquestación del servicio centrado en proceso, mediante la utilización de BPML (Business Process Modeling Language) como BPELWS y el workflow de ejecución de procesos del Negocio descrito.

Roles: El rol responsable de esta actividad es el Arquitecto, el Analista y el Especialista Técnico.

Entregable de entrada: Esta actividad tiene como entrada el Modelo de Casos de Uso del Negocio y el Modelo de Servicios.

Entregable de salida: Como salida tiene el Modelo de Implementación y el Modelo de Servicios con los mapeos correspondientes realizados y la orquestación de servicios definida para los servicios centrados en proceso.

Justificación: En esta actividad se debe definir la secuencia de invocaciones de servicios que se realiza por parte del servicio orquestador, para realizar los procesos del Negocio identificados y modelados como Casos de Uso del Negocio. Se utilizará un motor de ejecución de procesos que sea capaz de interpretar y ejecutar los pasos definidos en el proceso del Negocio modelado. Para esto los procesos serán definidos en Business Process Modeling Language (BPML) específicamente en BPEL, a partir del modelado gráfico de los mismos que convierte dicha definición la especificación en BPEL correspondiente.

Pero existen herramientas que mediante el diagrama de actividad UML, que permite luego crear la definición en BPEL asociada. Este es el caso por ejemplo de la herramienta JBPM provista como parte del servidor de aplicaciones JBoss.

Actualmente en la Dirección de Tecnología y Sistemas de Información se gestiona aun no se ha definido que herramienta se utilizara para realizar esta actividad.

Esta actividad constituye otro aspecto importante del enfoque a servicios planteado por el método propuesto, que fuerza a la realización de procesos del Negocio mediante la invocación de servicios de menor granularidad definidos, cumpliendo con las características del enfoque SOA presentado en el capítulo 2.

Es el quinto paso enfocado en servicios que se propone en el método, que presenta en forma explícita la necesidad de realizar los procesos del Negocio con servicios orquestadores de otros servicios. Es por esta razón que se incluye como una actividad específica en el método propuesto, de forma que se realice obligatoriamente la orquestación de servicios para realizar los procesos del Negocio definidos.

Se utilizara un motor de ejecución de procesos que sea capaz de interpretar y ejecutar los pasos definidos en el proceso de negocio modelado. Para esto los procesos serán definidos en Business Process Modeling Language (BPML) específicamente en BPEL.

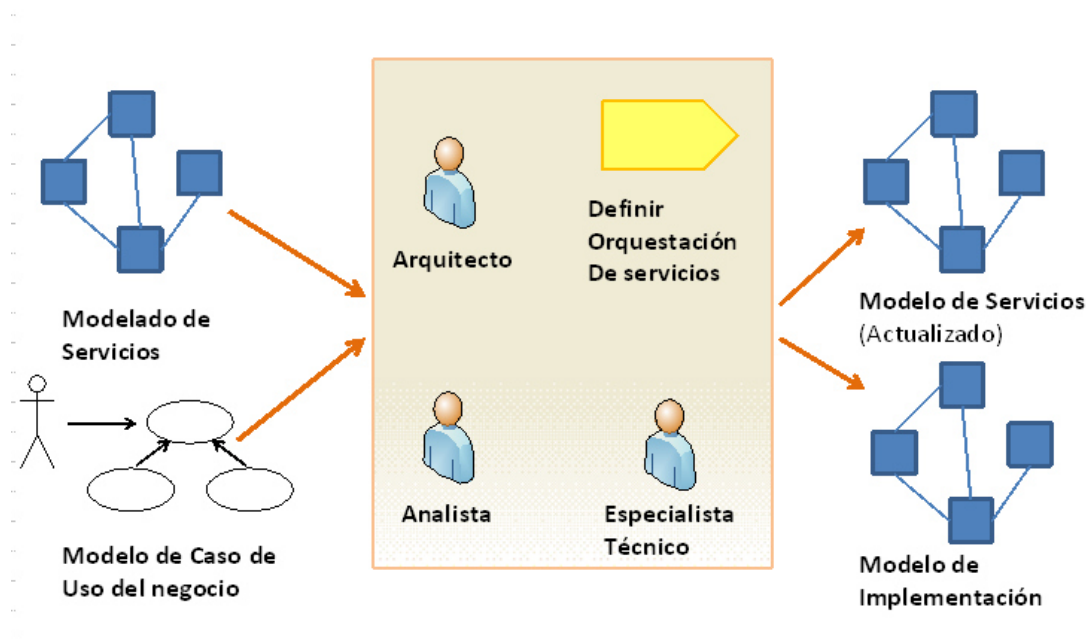


Figura 27: Diagrama de la actividad Orquestación de Servicios

No se define en esta primera propuesta una actividad para definir coreografías de servicios, por estar todavía en coordinaciones con los Operadores Logísticos y además aun no se definido que procesos serán ejecutados por las otras organizaciones.

Relación de la disciplina con el proceso RUP Adaptado

La Disciplina de Diseño en el proceso base cuenta con una importante cantidad de actividades que se deben realizar en coordinación con las que plantea el método para el enfoque SOA. Se define la agrupación de actividades especificada como Definir Servicios que corresponde a la realización conjunta de las actividades, identificar y

categorizar servicios, especificar servicios y investigar servicios existentes, ya que las tres se realizan en forma conjunta por la relación que tienen entre sí, la cual se realiza iterando con la actividad del proceso base identificada como Describir la Arquitectura.

La definición de la Arquitectura plantea la identificación de los Casos de Uso del Sistema relevantes a la Arquitectura, que son aquellos que por sus requerimientos tanto funcionales como no funcionales, definen el esqueleto de la aplicación a construir en cuanto a definición de subsistemas e interacción entre los mismos y con otras aplicaciones, definiendo por ejemplo la necesidad de aplicar patrones de diseño como la encapsulación del acceso a un dispositivo de hardware o a otro sistema sobre el cual no se tiene control, o el patrón de acceso a datos, entre otros.

En base a dichos Casos de Uso se realiza la modularización del software en subsistemas con alta cohesión y bajo acoplamiento que permitan identificar todas las áreas que involucra el desarrollo, ejercitando distintas hipótesis sobre el comportamiento del mismo. Es teniendo en cuenta esta modularización, más los Casos de Uso del Sistema identificados y los Casos de Uso del Negocio que describen los procesos de la Organización, más la clasificación de servicios presentada, que es posible pensar y definir los servicios en los distintos niveles, que seguirán luego el flujo de trabajo de la disciplina de Diseño en el método planteado.

3. Disciplina de Implementación

El propósito de esta disciplina es implementar los componentes proveedores de servicios según lo establecido por la asignación de servicios a componentes, implementar el ligamiento de servicios en las aplicaciones frontend y los servicios que llaman a otros servicios.

La actividad que se describe es esta disciplina es: Implementar servicios.

3.1. Implementar servicios

Objetivo: Implementar servicios definidos, para lo cual se deben tener en cuenta el tipo de servicio, las interfaces diseñadas, la interacción, con otros servicios (con o sin repositorio de servicios).

Descripción: Se debe implementar los componentes definidos para los servicios identificados, de acuerdo a las interfaces y componentes especificados en la Disciplina

de Diseño. Se debe proveer para cada interface los métodos definidos respetando los parámetros (nombre, tipo) especificados, valor de retorno (nombre, tipo) y la utilización de otros servicios para la realización de las operaciones definidas.

Roles: El rol responsable de esta actividad es el Implementador o Desarrollador que programa los servicios definidos.

Entregables de entrada: Esta actividad tiene como entrada el Modelo de Implementación y el Modelo de Servicios.

Entregable de salida: Como salida se tiene el servicio implementado con el componente definido.

Justificación En esta actividad se deben implementar los componentes asignados a servicios, según fue definido en las actividades de diseño correspondientes. Se cuenta con la definición de servicios en el Modelo de Servicios, y con la definición de los componentes asignados a servicios realizada, por lo que, la implementación debería ser directa.

Esta actividad representa el resultado de las actividades definidas en la Disciplina de diseño, y fuerza a que se respete el diseño realizado de servicios y componentes correspondientes, de forma de mantener las características del enfoque SOA presentado en el capítulo 2. Es por esta razón que se incluye como una actividad específica en el método propuesto.

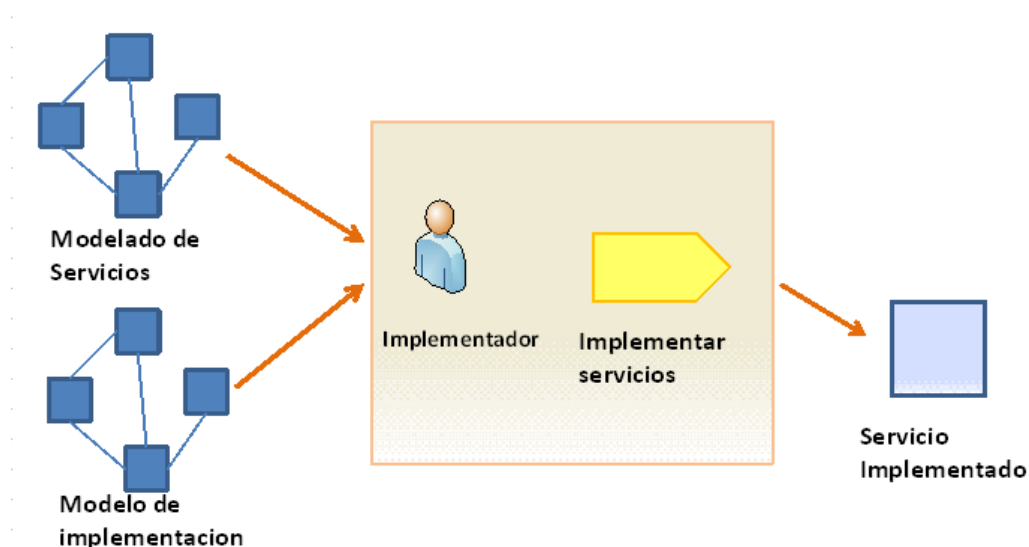


Figura 28: Diagrama de la actividad Implementar servicios

3.2. Planificar la integración de la iteración

Objetivo: El objetivo de esta actividad es elaborar el Plan de integración de la iteración. Para esto se deben establecer claramente las fechas en que se realizarán las integraciones de los componentes asignados a los servicios, en forma sucesiva hasta el final de la iteración, así como las responsabilidades sobre los distintos servicios a integrar.

Descripción: Se deben definir que componentes asignados a servicios se integrarán en la iteración, de acuerdo al plan de desarrollo definido para la iteración, ajustando si fueran necesarias las estimaciones de tiempos de implementación definidas para incorporar las integraciones de componentes asignados a servicios. Se deben identificar claramente las dependencias entre los componentes asignados a los servicios que se integrarán en esta iteración, para definir la estrategia más adecuada a utilizar.

Roles: Será el Implementador que tomara este rol para esa iteración, el Arquitecto que tiene el conocimiento del diseño y la implementación, el Jefe de proyecto que es el responsable del plan de desarrollo.

Entregables de entrada: Esta actividad tiene como entregables de entrada el Plan de Desarrollo, la Descripción de la Arquitectura, el Modelo de Diseño, el Modelo de Servicios, y el Modelo de Implementación, para la identificación de dependencias entre los componentes asignados a los servicios. Adicionalmente se podrán utilizar los Modelos de Casos de Uso del Negocio y de Casos de Uso del Sistema cuando surjan dudas sobre dependencias o definiciones.

Entregables de salida: Esta actividad tiene como entregable de salida el Plan de Integración de la iteración.

Justificación: Esta actividad se agrega para dar soporte a la gestión del desarrollo basado en servicios, donde la planificación del desarrollo, integración y verificación de la iteración, se realiza referenciando los componentes asignados a servicios que se obtendrán como resultado de la misma. Es recomendable que esta planificación este en el plan de desarrollo del sistema.

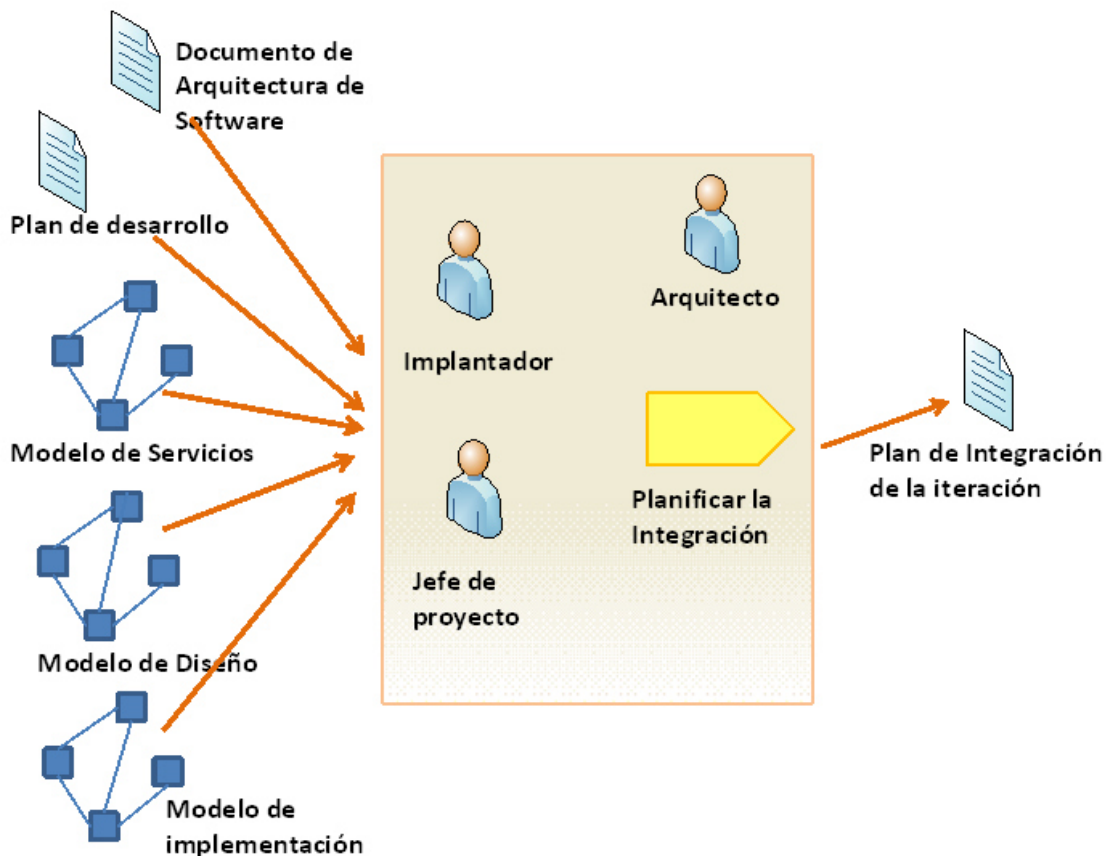


Figura 29: Diagrama de la actividad Planificar la Integración de la iteración

3.3. Integrar el sistema

Objetivo: El objetivo de esta actividad es realizar la integración de servicios como está previsto en el Plan de Integración de la iteración. Por lo que, hasta llegar a la integración final de la iteración, se ejecuta varias veces para realizar integraciones intermedias, sobre las que se van agregando componentes asignados a servicios en cada integración según está definido en dicho plan.

Descripción: Realizar la integración de los componentes asignados a servicios en cada momento, como está definido en el Plan de Integración. Para cada integración que se realiza se debe asegurar la existencia de los elementos necesarios para simular componentes que no están disponibles. Para obtener el incremento sobre el sistema que corresponde a cada integración, se deben integrar los componentes definidos generando un sistema intermedio, que se tomará para la siguiente integración y se le agregarán los componentes que corresponda para obtener el nuevo incremento. Tanto los sistemas

intermedios como el ejecutable final de la iteración son liberados a verificación para verificar la integración realizada.

Roles: Será el Implementador que toma este rol para esa iteración, los implementadores que tienen asignados componentes clave, que puedan ayudar a resolver los problemas que ocurran en la integración, el Arquitecto por su conocimiento del diseño e implementación y el Jefe de Proyecto por las planificaciones de desarrollo e integración.

Entregables de entrada: Esta actividad tiene como entregables de entrada el Plan de Integración, la Descripción de la Arquitectura, el Modelo de Diseño, el Modelo de Servicios y el Modelo de Implementación, para la identificación de dependencias entre los componentes asignados a los servicios. Adicionalmente se podrán utilizar los Modelos de Casos de Uso del Negocio y Casos de Uso del Sistema cuando surjan dudas sobre las dependencias o las definiciones realizadas.

Entregables de salida: Esta actividad tiene como entregable de salida el sistema integrado con los componentes asignados a servicios definidos para integrar en la iteración.

Justificación: Esta actividad se agrega para dar soporte a la gestión del desarrollo basado en servicios, de forma de enfocarse en los componentes asignados a servicios que se obtendrán como resultado de la misma. Se debe seguir la planificación realizada en base a servicios para obtener la integración de componentes asignados a servicios correspondiente a cada etapa definida.

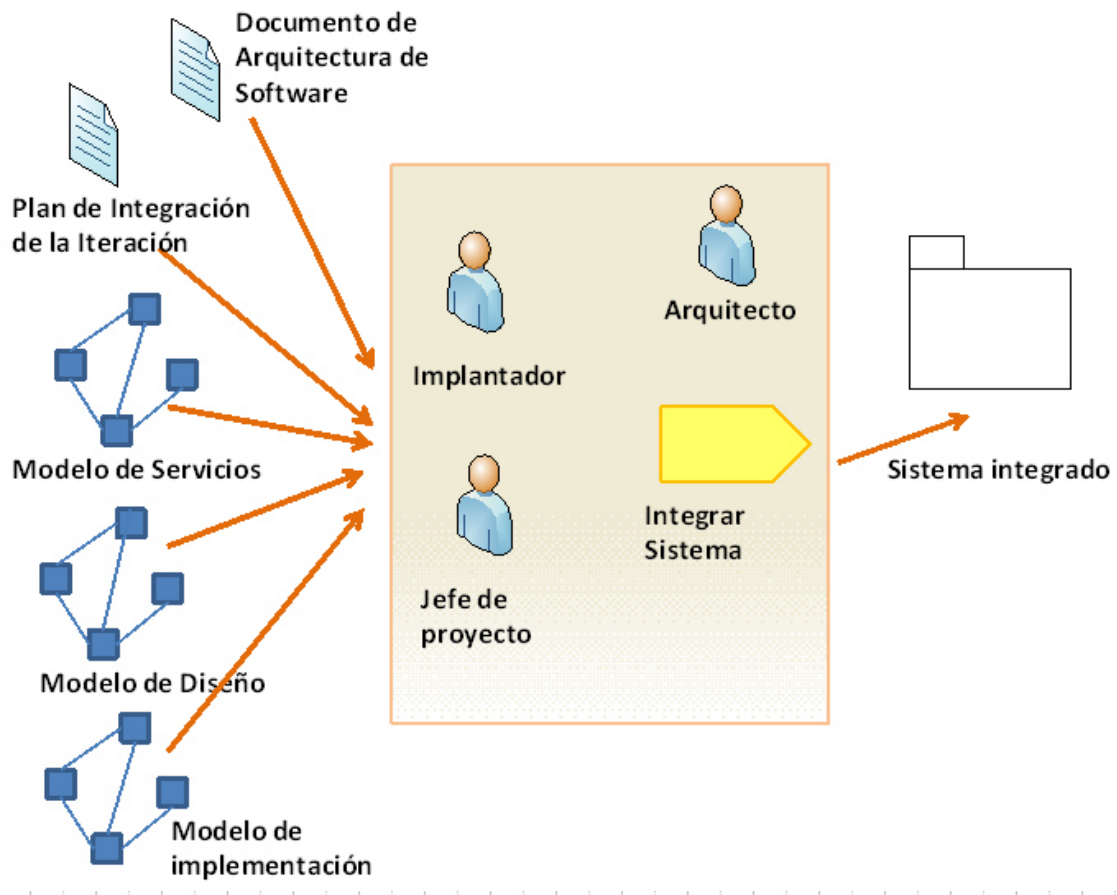


Figura 30: Diagrama de la actividad Integrar Sistema

3.4. Realizar pruebas internas

Objetivo: El objetivo de esta actividad es realizar la verificación unitaria de los componentes asignados a servicios en el ambiente de desarrollo, para poder detectar fallas en los mismos antes de integrarlos con el resto de los servicios.

Descripción: En esta actividad se debe realizar la verificación unitaria de los componentes asignados a servicios y los mismos servicios, según la especificación del servicio se definen las pruebas a realizar sobre el componente que lo implementa, verificando la correctitud de los parámetros (nombre, tipo) especificados, valor retornado (nombre, tipo) y la utilización de otros servicios para la realización de las operaciones definidas.

Roles: El rol responsable de esta actividad es el Implementador o desarrollador, el jefe de proyectos debe verificar que las pruebas se ejecuten.

Entregable de entrada: Esta actividad tiene como entregables de entrada el componente que implementa el servicio a probar, el servicio mismo, el Modelo de Servicios y el Plan de Pruebas Internas que es parte del plan de desarrollo del proyecto.

Entregable de salida: Esta actividad tiene como entregable de salida el Reporte de pruebas unitarias.

Justificación: Esta actividad se agrega para dar soporte a la calidad de los servicios liberados desde implementación para su integración con el resto de los servicios definida para la iteración y hasta conformar el sistema como un todo..

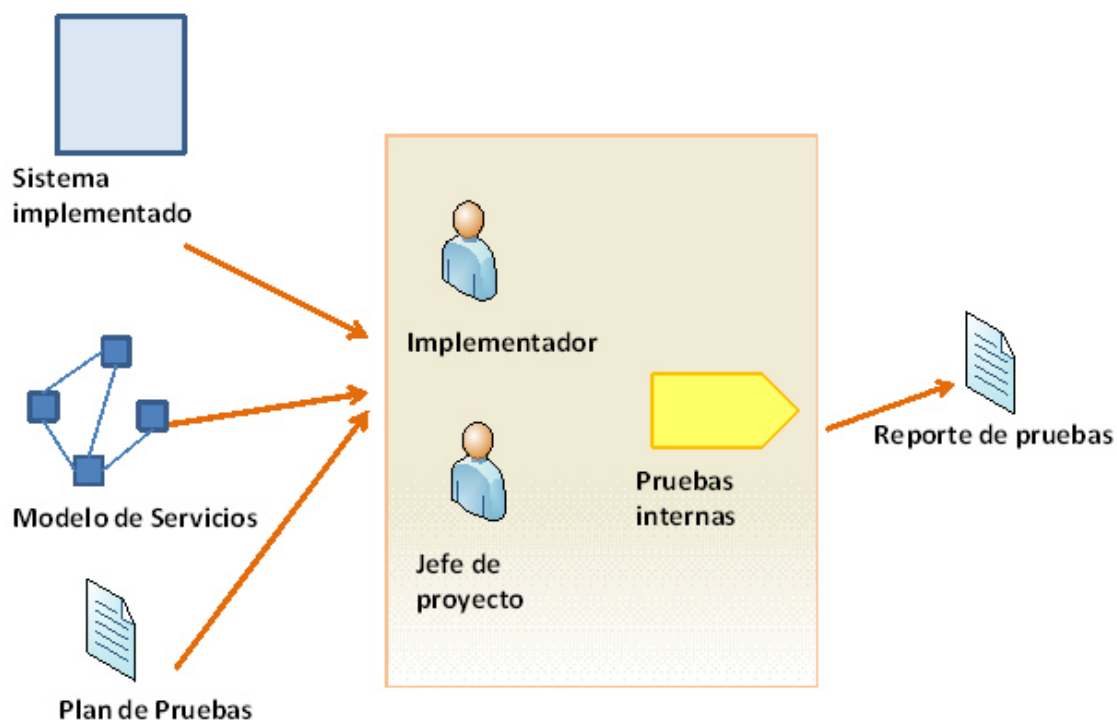


Figura 31: Diagrama de la actividad Pruebas internas

4. Disciplina de Verificación o Testeo

El propósito de la disciplina de Verificación en el método propuesto agrega los siguientes objetivos a los definidos en el proceso base: utilizar como unidad de verificación los servicios para realizar las pruebas del software, realizando verificación unitaria, de integración y de sistema, así como pruebas de regresión para asegurar que se mantiene el mismo funcionamiento del software en versiones siguientes a la ya testeada.

En esta Disciplina los roles definidos en el proceso base Responsable de Verificación que tendrá a su cargo la gestión administrativa y técnica de las actividades de

verificación será el Área de testing perteneciente a la gerencia de Producción e Infraestructura y el jefe de proyecto. Para cumplir con los objetivos agregados para la Disciplina de Verificación, se contemplan las siguientes actividades existentes en el proceso base del RUP y en la Metodología Megon: Planificar las pruebas, Especificar los Casos de Prueba y Ejecutar las pruebas.

4.1. Planificar las Pruebas

Objetivo: Esta actividad tiene como objetivo planificar las pruebas para la iteración. Para esto se deben establecer claramente las fechas en que se realizarán las verificaciones sobre los componentes asignados a los servicios y sus integraciones, en forma sucesiva hasta el final de la iteración, así como las responsabilidades sobre la verificación de los distintos elementos.

Descripción: Se deben planificar las pruebas de la iteración, teniendo en cuenta las planificaciones realizadas de desarrollo, con las que se deben coordinar las fechas y procedimientos de trabajo. Esta planificación debe incluir los elementos que se deben verificar en la iteración, esto es los componentes asignados a servicios y las sucesivas integraciones que se realizarán de los mismos en la iteración, quien debe hacer cada verificación, cuando se debe comenzar a realizar cada verificación y en qué fecha debe entregar el informe de verificación correspondiente.

Roles: El rol responsable de esta actividad es el Jefe de proyecto y área de Testing juntos en coordinación se elabora el plan de pruebas de testing.

Entregables de entrada: Esta actividad tiene como entrada el Plan de desarrollo y el Plan de Pruebas de Integración, Definición de la Solución, Casos de prueba, planificaciones con las que debe estar alineada la verificación a realizar en la iteración, y el Modelo de Casos de Prueba para confirmar si existen o no los casos de prueba correspondientes a los componentes asignados a servicios que se probarán en la iteración.

Entregables de salida: Esta actividad tiene como entregable de salida el Plan de Pruebas de Testing.

Justificación: Esta actividad, de la misma forma que la actividad Planificar la integración de la iteración, se agrega para dar soporte a la gestión del desarrollo basado en servicios, donde la planificación del desarrollo, integración y verificación de la

iteración, se realiza referenciando los componentes asignados a servicios que se obtendrán como resultado de la misma.

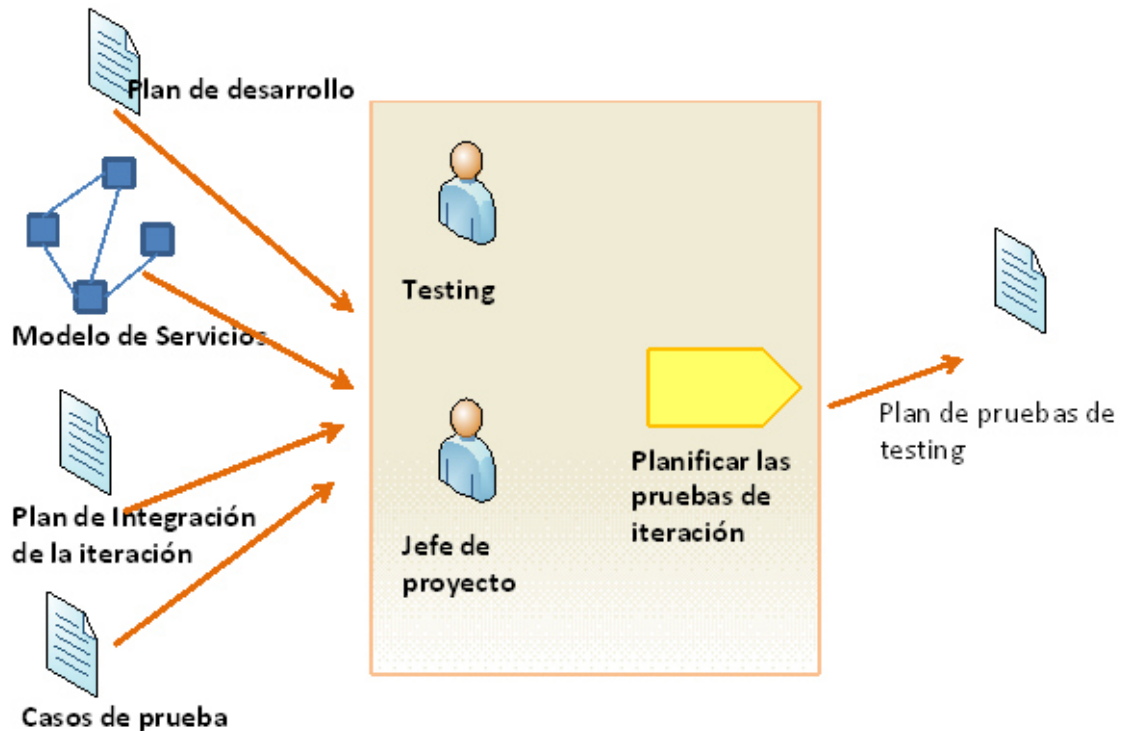


Figura 32: Diagrama de la actividad Planificar Pruebas

4.2. Especificar los Casos de Prueba

Objetivo: El objetivo de esta actividad es realizar la especificación completa de los casos de prueba para el sistema bajo desarrollo. Los casos de prueba deben corresponder a las pruebas unitarias a realizar sobre los servicios definidos en la Disciplina de Implementación, a las pruebas de integración de componentes asignados a servicios a realizar en las sucesivas iteraciones, y a la prueba del sistema de extremo a extremo, incluyendo las pruebas correspondientes a los procesos del Negocio definidos como Casos de Uso del Negocio.

Descripción: Los implementadores o desarrolladores, Analista, y Arquitecto realizarán la especificación de los Casos de Prueba para la verificación de los servicios implementados en componentes del sistema, integración de los mismos y el sistema en sí.

Roles: El rol responsable de esta actividad es el Analista, Implementador o desarrollador y el Arquitecto.

Entregables de entrada: Esta actividad tiene como entrada el Modelo de Casos de Uso del Negocio, Modelo de Casos de Uso del Sistema, el Modelo de Diseño, el Modelo de Servicios, el Modelo de Implementación y el Plan de Pruebas de Testing para tener la información de los componentes asignados a servicios y funcionalidades que serán probadas en dicha iteración.

Entregables de salida: Esta actividad tiene como salida el Documento de Casos de Pruebas, documento Megon.

Justificación: Esta actividad se agrega para dar soporte a la calidad de los servicios construidos, desde el punto de vista de la verificación del cumplimiento de los mismos por parte de los componentes asignados a servicios implementados.

Esta actividad es importante para la liberación de un producto compuesto por servicios que cumplen su especificación y para la confianza que se pueda tener por parte de otros proyectos que los reutilicen, sobre el funcionamiento de los mismos y el cumplimiento de los contratos publicados, por parte de los componentes que los implementan.

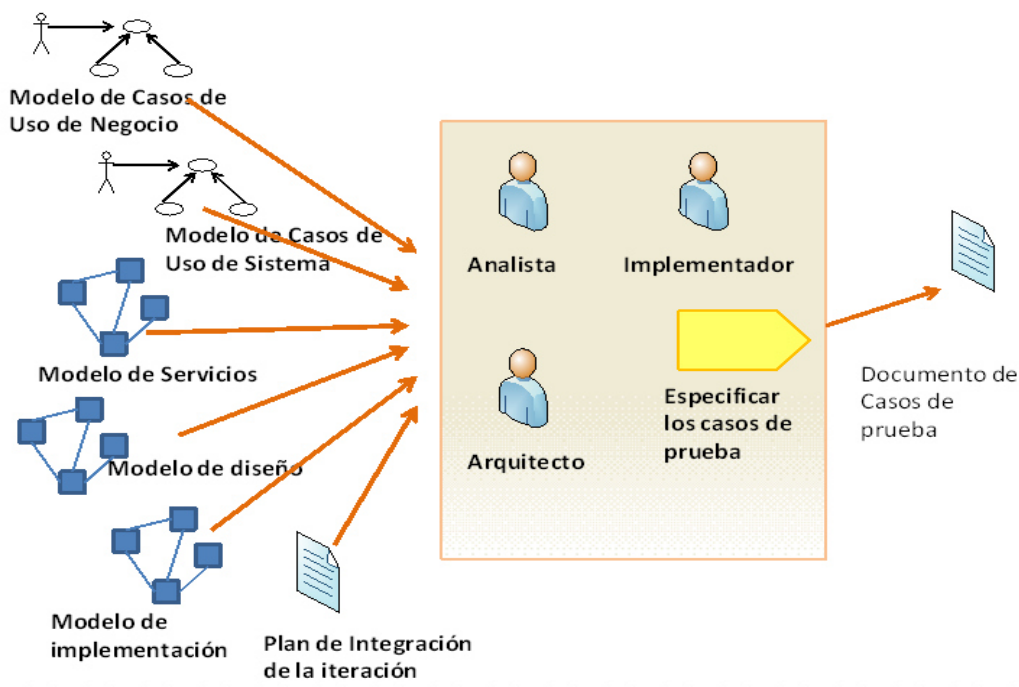


Figura 33: Diagrama de la actividad Especificar Casos de Prueba

Nota: Cabe mencionar además que antes de dar inicio a la disciplina de Verificación o Testeo y después de la disciplina de Implementación existe una actividad Megon que se debe realizar que son las pruebas de funcionales de usuario, terminada esta actividad recién se da inicio a las pruebas de testing para luego pasar al proceso de implantación en producción. Estas actividades no se ha descrito en este método propuesto por no estar en el alcance del trabajo.

Capítulo 5: Descripción de la solución

En este capítulo se presenta la descripción de la solución propuesta para el caso de estudio, aplicando el método propuesto para el enfoque SOA.

5.1 Contexto del caso Telefónica Gestión de Servicios Compartidos S.A.C.

El método propuesto está actualmente en proceso de prueba en La Dirección de Tecnología y Sistema de Información de tgestiona, como una iniciativa a un requerimiento de solución informática por la Dirección de Logística, donde existe un sistema llamado COMEX que intenta concentrar los proceso de comercio exterior de distintas fuentes de información y que en coordinación con la otras áreas del proceso y organizaciones externas (proveedores) se plantea la necesidad de buscar una nueva forma de comunicación e integración de los sistemas o fuentes de información.

Cabe mencionar que el requerimiento de la Dirección de Logística aun no es oficial por estar dentro del plan de sistemas del 2009, mencionar además que la prueba del método propuesto se encuentra aun en ejecución en la Direccione de Tecnología y Sistemas de Información, por lo que las disciplinas de implementación y testeo aún no han sido realizadas.

5.2 Principales Requerimientos del Sistema

El sistema de Comercio Exterior (COMEX), se debe poder integrarse con los operadores logísticos quienes brindaran la información de los embarques de las órdenes de compra, independientemente de la tecnología que estos utilizan, a su vez también el sistema COMEX debe poderse integrarse con el sistema SAP donde se encuentra la información del detalle de las órdenes de compra y los pagos realizados a los diferentes Actores del proceso de Comercio Exterior (Aduanas, Operador Logístico, Proveedor de material).

El sistema COMEX también de poder brindar servicios hacia otras áreas de la cadena logística, como por ejemplo al área de Distribución Física que cuenta con el sistema de Distribución Física (SDF), también debe brindar servicios a los mismo operadores

logísticos, por ejemplo debe poder enviar la información necesaria de la orden de compra para que los operadores puedan iniciar el proceso operativo de importación en sus propios sistemas y evitar la espera de los documentos físicos y transcribir nuevamente la información.

Cuando se registre una orden de compra en el sistema COMEX se deberá asignar a un analista responsable de hacer el seguimiento la orden de compra, así también se deberá asignar al operador encargado de realizar el proceso operativo de la importación.

El sistema COMEX debe permitir mostrar un modulo la información general de la orden de compra y el detalle las posiciones de la orden de compra que se encuentran en SAP, así también debe de poder mostrar en este modulo la cantidad de embarques que contiene la orden de compra por parte del operador logístico y además mostrar a que embarque pertenecen las posiciones de la orden de compra, también debe permitir genera Anticipos (documentos de pago) desde el sistema COMEX hacia SAP por cada embarque que se requiera.

5.3 Desarrollo y seguimiento de la prueba

Antes de dar inicio a la prueba se reunió el grupo de trabajo del área de e-Business de la Gerencia de Proyectos de la Dirección de Tecnología y Sistemas de Información, para definir los roles que tendría cada uno y de los roles que le corresponderían a las otras áreas de la dirección.

Roles específicos asignados a un área de la dirección:

- *Arquitecto:* En la dirección de Tecnología y Sistemas de Información este rol estará a cargo de la gerencia de Producción e Infraestructura el cual posee una jefatura llamada Comité de Arquitectura conformado por personas encargadas de evaluar y aprobar las arquitecturas planteadas para cada proyecto que se inicie.
- *Evaluador o Verificador:* En la dirección de Tecnología y Sistemas de Información este rol estará a cargo de la gerencia de Producción e Infraestructura el cual posee una jefatura llamada Testing conformado por personas encargadas de evaluar y aprobar las pruebas de calidad de las aplicaciones antes de pasar a producción.

5.4 Revisión de entregables

Los entregables de la Extensión SOA que se agregan son los correspondientes a las Disciplinas de Modelado del Negocio y Diseño, donde se encuentran las actividades que componen el núcleo del método. Dichos entregables tienen definida la Fase, iteración y semana en que deben ser entregados, según la realización de actividades que los generan.

Durante el desarrollo de la prueba esta planificación se vio desfasada por los atrasos que el grupo tuvo debidos a diversas razones que afectaron la realización en tiempo y forma de las actividades y entregables previstos.

Fase	Iteración	Semana	Disciplina	Entregable	Responsable	Entrega
Inicial	Iteración 1	1	Modelado del Negocio	Evaluación de la Organización Objetivo	Arquitecto	Si
Inicial	Iteración 1	1	Modelado del Negocio	Modelo de Caso de Uso del Negocio	Arquitecto	Si
Inicial	Iteración 2	3	Modelado del Negocio	Evaluación de la Organización Objetivo	Arquitecto	No
Inicial	Iteración 2	3	Modelado del Negocio	Modelo de Caso de Uso del Negocio	Arquitecto	Si
Inicial	Iteración 2	3	Diseño	Modelo de Servicios	Arquitecto	No
Elaboración	Iteración 1	5	Modelado del Negocio	Modelo de Caso de Uso del Negocio	Arquitecto	No
Elaboración	Iteración 2	6	Diseño	Modelo de Servicios	Arquitecto	Si

Tabla 4: Planificación de Entregables

Fases	Inicial			Elaboracion			Construccion			Transicion	
Iteracciones	Iter.1		Iter.2	Iter.1		Iter.2	Iter.1		Iter.2	Iter.1	
semanas	1	2	3	4	5	6	7	8	9	10	11
Real	Inicial			Elaboracion			Construccion			Trans.	
12/05						Corte 12/06					

Tabla 5: Duración prevista y real de las Fases e iteraciones de la prueba

Principales problemas:

Los problemas principales se debieron a no tener completa la documentación correspondiente a la definición de la Arquitectura inicial y por lo tanto, el diseño de los Casos de Uso para comenzar la implementación en la Fase de Elaboración, y la definición del Alcance del proyecto, que no pudo ser acordado con el Cliente en los tiempos previstos.

Hubo una falta de compromisos con las otras áreas de la dirección debido a que como aun no es oficial el proyecto existen prioridades en estas áreas de servicio.

5.5 Resultados por fase e iteración

En esta sección se describen los resultados obtenidos por Fase e iteración en cuanto a versiones de los entregables de la Extensión SOA realizados y entregados, más los principales entregables relacionados del proceso base, y cantidad de Casos de Uso del Negocio, Casos de Uso del Sistema, Casos de Uso relevantes a la Arquitectura identificados y servicios definidos.

En la Fase Inicial se entregó una única versión del documento de Evaluación de la Organización Objetivo en el cual se identificaron los aspectos principales de la Organización en que se realizaría el desarrollo. Se entregaron también dos versiones del Modelo de Casos de Uso del Negocio con la identificación de los procesos del negocio

realizada y la especificación de los flujos correspondientes. A partir de este Modelo se realizaron y entregaron también una versión del Modelo de Casos de Uso del Sistema, y una versión de la Descripción de la Arquitectura de Software con la definición inicial de la Arquitectura.

En la Fase de Elaboración no se entregaron nuevas versiones de los entregables correspondientes al modelado del Negocio, Se entregó la primera versión del Modelo de Servicios, cuya versión final sería entregada recién en la última fase del proyecto.

Entregable	Cantidad de versiones	Fase Entrega	Iteración	Corresponde
Evaluación de la Organización Objetivo	1	Inicial	1	Extensión SOA
Modelo de Caso de Uso del Negocio	2	Inicial	1 y 2	Extensión SOA
Modelo de Caso de Uso de Sistema	1	Inicial	1 y 2	Proceso base
Alcance del Sistema	1	Inicial	1 y 2	Proceso base
Descripción de la Arquitectura de Software	1	Inicial	2	Proceso base
Modelo de Servicios	2	Elaboración	2	Extensión SOA

Tabla 6: Cantidad versiones y entregables principales en cada Fase e iteración

5.6 Resultados por Disciplina, actividad y entregable

En esta sección se presentan los resultados obtenidos en las Disciplinas Modelado del Negocio, Diseño e Implementación en las cuales la Extensión SOA agrega las principales actividades y entregables para el desarrollo orientado a servicios

Disciplina Modelado del Negocio

En la Disciplina Modelado del Negocio se realizaron las dos actividades previstas por la metodología: Evaluar la Organización Objetivo y Identificar los procesos del Negocio, obteniendo los entregables correspondientes.

En el entregable Evaluación de la Organización Objetivo se identificaron aspectos del proyecto COMEX y de los sistemas y plataforma SOA existentes que sirvieron de contexto al desarrollo. En el entregable Modelo de Casos de Uso del Negocio se identificaron ocho procesos principales luego de dos versiones del entregable, los que se describieron como Casos de Uso del Negocio identificando Actores que participan, flujo principal y flujos alternativos.

Los Casos de Uso del Negocio fueron descritos en forma textual y en forma gráfica mediante Diagramas de Actividad.

Las plantillas se presentan en los Anexos.

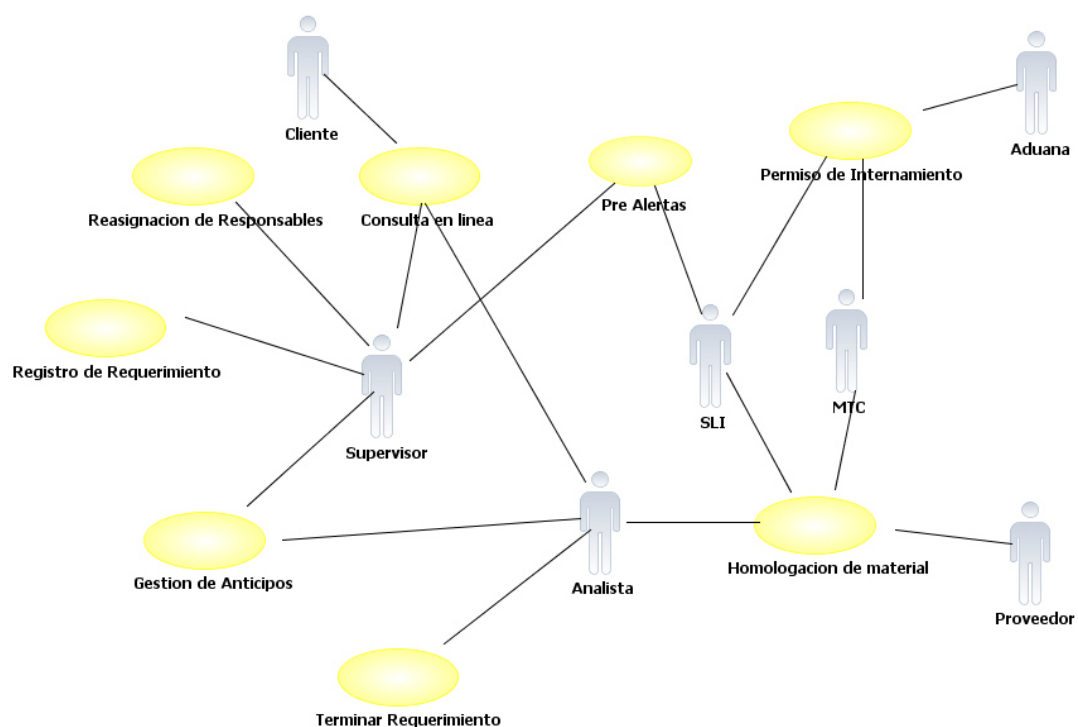


Figura 34: Diagrama de Casos de Uso del Negocio en la Prueba SOA

Si bien los Casos de Uso del Negocio identificados deberían corresponder todos a procesos del Negocio, se observa que en algunos casos éstos no presentan la granularidad correspondiente a un proceso del Negocio completo por, ejemplo Pre Alertas, sino más bien a un paso en algún otro proceso macro, lo que sería equivalente a un Caso de Uso del Sistema en todo caso. Esta es una de las dificultades encontradas en

el seguimiento del enfoque, la identificación de los procesos con la granularidad correcta, que permita especificar los procesos del Negocio en forma completa, para luego traducirlos en Casos de Uso del Sistema de menor granularidad que los realicen mediante pasos sucesivos en la ejecución del Sistema.

De los Casos de Uso del Negocio definidos el que está asociado a un principal proceso de la Organización es el Registro de Requerimiento, el cual se compone de pasos: Registrar Orden de Compra, asignación de Responsable, Asignación de Operador Logístico, Validar Orden en SAP y Enviar Correo.

Caso de uso de Negocio	Descripción
Registro de Requerimiento	El Supervisor recibe un requerimiento, que será validado su existencia en el sistema SAP, para luego ser asignado a un analista responsable quien hará el seguimiento de la importación y a un Operador Logístico que será el encargado de realizar la importación de la mercancía.

Tabla 7: Ejemplo de descripción del Caso de Uso del Negocio Registro de Requerimiento

Disciplina de Requerimientos

Si bien la Disciplina de Requerimientos y las actividades y entregables definidas corresponden al proceso base, la realización del Modelo de Casos de Uso del Sistema es parte importante de la metodología de la Extensión SOA, ya que a partir de los Casos de Uso del Negocio identificados más los requerimientos especificados se derivan los Casos de Uso del Sistema que luego se utilizan como entrada para la Definición de la Arquitectura y también de los servicios asociados.

La cantidad de Casos de Uso del Sistema definidos es de 20 hasta la fecha del corte, por lo que no se mostrarán en su totalidad, sino que se presentará un subconjunto.

El Caso de Uso del Negocio Registro de Requerimiento se desglosa en varios Casos de Uso de Sistema para realizar el proceso del negocio correspondiente: Asignar Analista Responsable, Asignar Operador Responsable, Registrar Orden de Compra, Consultar Orden de Compra SAP y Enviar Correo.

El flujo principal del Caso de Uso del Negocio comienza cuando un Supervisor recibe un requerimiento de importación, para el cual este requerimiento es asignado a un

Analista Responsable, luego es asignado a un Operador Logístico para proseguir a realizar el registro de la Orden de Compra en el sistema, el registro de la Orden de Compra en el sistema COMEX debe realizar una previa validación de su existencia en el sistema SAP si todo es correcto debe enviar notificaciones vía correo electrónico al analista seleccionado y al operador seleccionado.

Caso de Uso del Sistema	Descripción	Importancia
Asignar Analista Responsable	El Supervisor una vez que recibe un requerimiento de importación, procede a asignarle un Analista Responsable de hacer el seguimiento de la importación, es obligatoria la asignación de responsable.	Parte del CU de Negocio Registro de Requerimiento.
Asignar Operador Logístico	El Supervisor una vez ya asigno al Analista responsable de la Orden de Compra, procede a asignarle un Operador Logístico que será el encargado de realizar la importación física de la mercancía, es obligatoria la asignación de Operador Logístico.	Parte del CU de Negocio Registro de Requerimiento.
Registrar Orden de Compra	El supervisor registra la Orden de Compra en el sistema COMEX.	Relevante a la Arquitectura – parte del CU Registro de Requerimiento.
Consultar Orden de Compra SAP	El siguiente CU es invocado por el CU de Sistema Registrar Orden de Compra para realizar una validación previa en SAP.	Relevante a la Arquitectura – parte del CU Registro de Requerimiento.
Enviar Correo	En este caso de uso se maneja el envío de mensajes vía e-mail, esto se da cuando se registra la Orden de Compra en el sistema COMEX.	Relevante a la Arquitectura – parte del CU Registro de Requerimiento

Consulta de embarques	Se podrán ver los embarques de un orden de compra desde la base que están siendo registrados por el operador en sus sistemas.	Relevante a la Arquitectura
Registro de Anticipos	El supervisor registra los anticipos desde el sistema COMEX a SAP	Relevante a la Arquitectura

Tabla 8: Descripción de los principales Casos de Uso del Sistema en la prueba SOA

Disciplina de Diseño

En la Disciplina de Diseño se realizaron las actividades definidas para el diseño orientado a servicios: Identificar y categorizar servicios, Especificar servicios, Investigar servicios existentes, Asignar servicios a componentes y Definir orquestación de servicios. Las tres primeras actividades se realizan conjuntamente con la actividad de Definición de la Arquitectura del proceso base.

En dicha actividad, a partir de los Casos de Uso del Sistema identificados como relevantes a la Arquitectura que se presentan en la vista de Casos de Uso del Documento de la Arquitectura, se realiza la descomposición del software en subsistemas en la vista Lógica, más la identificación de procesos en la vista de Procesos, el mapeo a la implementación de dichos elementos de software en la vista de Implementación y el despliegue del software en el hardware en la vista de Deployment.

La método SOA agrega el Modelo de servicios donde se incluye la información correspondiente a los servicios identificados, categorizados y especificados, del cual se extrae la vista de Servicios que se incluye en el Documento de la Arquitectura, donde se muestra la identificación de servicios realizada, para luego presentar también la asignación de servicios a componentes en la vista de Implementación, y su despliegue en la vista de Deployment. Los servicios fueron identificados y categorizados siguiendo la guía de clasificación propuesta (Ver Anexos), en servicios centrados en procesos para la definición de servicios que orquestan otros servicios y corresponden a procesos del Negocio, servicios intermediarios que proveen funcionalidades más complejas utilizando servicios básicos del negocio y de acceso a datos.

Servicio	Categorización	Propósito
GestionOrdenCompra	Básico del Negocio	Encapsula todas las funcionalidades relacionadas a las Órdenes de Compra.
GestionAnalista	Básico del Negocio	Maneja información relativa al analista.
GestionSeguridad	Básico del Negocio	Proporciona mecanismos de seguridad que necesita el sistema COMEX
GestionAnticipos	Básico del Negocio	Maneja información relativa los anticipos.

Tabla 9: Ejemplo de Identificación y categorización de servicios en la prueba SOA

Servicio	Funcionalidades provistas	Categorización	Caso de uso Asociado	Sub Sistema Asociado
GestionOrdenCompra	. Registrar Orden de Compra . Asignar Analista responsable . Asignar Operador Logístico . Listar Orden de Compra . Listar Orden de Analista . Listar Orden de Operador . Cerrar Orden de Compra . Listar embarques de orden . Listar Posiciones de orden	Servicio básico de negocio	. Registro de Requerimiento . Listar Orden de Compra. . Ver Orden de Compra . Reasignar. Responsable . Listar Posiciones . Listar embarques	Gestión de Órdenes de Compra

Tabla 10: Especificación del servicio Gestión de Orden de Compra en la prueba SOA

Método	AsignarResponsableOC			
Parámetros	Analista	Char	04	Código del analista
	Orden de Compra	Char	10	Numero de Orden de Compra
Valor de retorno	1/0	Char	O2	1/0
Excepciones	BDEXcepcion			
Descripción	Se registra en el sistema un nuevo responsable de la orden de compra.			
Pre Condiciones	La orden de Compra de existir en el sistema COMEX			
Post Condiciones	Se envía una notificación vía correo electrónico			

Tabla 11: Contrato funcional del método AsignarResponsableOC

Capítulo 6: Conclusiones y trabajos futuros

6.1 Conclusiones sobre el trabajo

- Para este trabajo de tesina se realizó una investigación y conceptualización del enfoque de desarrollo Service Oriented Architecture (SOA) para el desarrollo de aplicaciones distribuidas con orientación a servicios, identificando las principales características que debe cumplir un producto SOA y su relación con otros enfoques como Business Process Management (BPM) para el tratamiento de procesos del Negocio.
- Se definió un método a seguir para el desarrollo de aplicaciones SOA, que se incorporó un proceso de adaptación del RUP para luego incorporarlo en el marco metodológico de trabajo Megon que sigue la empresa telefónica, que representa un paso en el camino para la adopción de dicho paradigma en forma disciplinada, basada en la conceptualización y estandarización de las prácticas asociadas con el enfoque y no a una herramienta o interpretación en particular maximizando el valor de la comprensión y aplicación del paradigma al desarrollo de software. La definición del método propuesto contiene Disciplinas, Actividades, Entregables y Roles para realizar el desarrollo, así como plantillas para los entregables y guías para los aspectos más importantes.
- El método propuesto incluye nuevos elementos a utilizar en el desarrollo como son los procesos del Negocio modelados como Casos de Uso del Negocio, y el Modelo de Servicios donde se identifican, definen, especifican y reutilizan los servicios que hacen a la Organización en general y al sistema en desarrollo en particular.
- Para la aplicación del método propuesto, se definió un conjunto núcleo de elementos a incorporar en el proceso base adaptación del RUP, en lo que se denominó Extensión del proceso RUP Adaptado para desarrollo SOA.

6.2 Trabajos Futuros

- Como trabajo a futuro han quedado varias líneas abiertas que permiten la continuación de la investigación en el área, en primer lugar la evaluación de la

suite de herramientas de IBM para modelado del Negocio en forma gráfica y con BPEL y su ejecución en el motor de procesos que se provee, esto debido a que IBM es una de los principales proveedores de Software en la empresa tgestionana.

- Se podría investigar también la integración de los conceptos en la metodología SOA propuesta con procesos ágiles como eXtreme Programing.
- Otra línea abierta de trabajo interesante y que está siendo tratada por la comunidad del software en general, es la conjunción del enfoque SOA con los enfoques presentados en el Capítulo 2, Business Process Management (BPM).
- Para la conjunción con BPM, si bien en el método propuesto existen actividades definidas que tienen en cuenta la utilización de ese enfoque, será necesario investigar herramientas que permitan realizar el modelado de los procesos del Negocio, tanto en forma gráfica para definir los diagramas de flujo y elementos asociados, como en BPEL para permitir la ejecución de procesos en un motor de ejecución de procesos. Existen herramientas que proveen esas tres características y se cuenta con una lista de herramientas de este estilo recomendadas por la OMG, pero cuya evaluación excedía el alcance definido para este trabajo.
- Otro aspecto a tener en cuenta es la conjunción de BPM-SOA-MDA (Model Driven Architecture) es como armonizar los enfoques y el uso de herramientas que los soportan, para obtener una suite de trabajo integral para BPM-SOA-MDA, donde el ideal sería una herramienta BPM que permita el modelado de procesos del negocio en forma gráfica, la asignación de servicios a estos procesos, tanto a nivel del proceso entero en servicios centrados en procesos, como en las invocaciones a servicios en el flujo de los mismos, la definición de los componentes que implementarán estos servicios diseñados con enfoque SOA, y la generación por parte de una herramienta de MDA de los componentes definidos, que puedan ser invocados en la ejecución del flujo definido en BPEL por el motor de ejecución de procesos de la herramienta BPMS.

6.3 Recomendaciones

- Se recomienda que luego de terminar todo el proceso de adaptación del RUP para desarrollos SOA, el método sea ajustado y mejorado según los resultados

obtenidos en la prueba de la misma y agregando elementos al conjunto núcleo definido inicialmente.

- Para iniciar en SOA o en cualquier otro paradigma una empresa debe tener una base metodológica disciplinaria que permita guiar a la empresa al logro de su objetivo, caso contrario podría ocasionar un costo de propiedad y la pérdida de la agilidad y competitividad que debía aportar la adopción de SOA.
- Los proyectos de sistemas deben ahora ser enfocados desde una perspectiva organizacional donde la participación de los actores vaya desde la alta gerencia hasta los desarrolladores.
- Una vez que las empresas adopten el uso de SOA es necesario un esquema de administración especial para seguir adoptando el uso de estándares y principios de SOA, para lo cual se hace necesario un proceso de SOA Governance.

Capítulo 7: REFERENCIAS BIBLIOGRÁFICAS

1. [BPMI] Business Process Management Initiative, www.bpmi.org/ [consultado 01 de mayo de 2008].
2. [BPMN06] Business Process Modeling Notation (BPMN) Specification, Object Management Group (OMG) Final Adopted Specification, febrero 2006, dtc/06-02-01, BPMI 2004 <http://www.omg.org/docs/dtc/06-02-01.pdf> [consultado 30 de abril de 2008]
3. [Endr04] Endrei Mark, Ang Jenny, Arsanjani Ali, Chua Sook, Comte Phillpe, Krogdahi Pal, Luo Min, Newling Tony, Patterns: Service-oriented Architecture and Web Services, IBM Redbook, SG24-6303-00, 2004, www.redbooks.ibm.com/redbooks/pdfs/sg246303.pdf [consultado 31 de marzo de 2008]
4. [Erlt05] Erl Thomas: Service-Oriented Architecture: Concepts, Technology, and Design, Prentice Hall, 1st. edition, 2005.
5. [Kraf05] Krafzig, D. Banke, K. Slama, D. : Enterprise SOA, Service Oriented Architecture Best Practices, Prentice Hall, 2005.
6. [Cast08] Castillo Felix, Fanny, Metodologia, estrategia, técnicas y herramientas de desarrollo, [http://sisbib.unmsm.edu.pe/bibvirtualdata/tesis/Basic/castillo ff/cap03.pdf](http://sisbib.unmsm.edu.pe/bibvirtualdata/tesis/Basic/castillo_ff/cap03.pdf) [consultado 09 de junio 2008]
7. [OASIS] Organization for the Advancement of Structured Information Standards (OASIS), www.oasis-open.org [consultado 31 de marzo de 2008].
8. [OMG] Object Management Group (OMG), www.omg.org/ [consultado 28 de marzo de 2008].
9. [Owen03] Owen, M., Raj, J. : BPMN and Business Process Management, Introduction to the New Business Process Modeling Standard, Popkin Software, 2003, <http://www.popkin.com> [consultado 31 de abril de 2008]
10. [RUP08] IBM - Rational Unified Process (RUP), www-306.ibm.com/software/awdtools/rup/ [consultado 31 de marzo de 2008]

11. [Shfp03] H.Smith, P.Fingar, Business Process Management: The third wave, Meghan-Kieffer Press, 2003.
12. [Stev04] Stevens M., McGovern J., Tyagi S., Mathew S., Java Web Services Architecture, Morgan Kaufmann Publishers, The Morgan Kaufmann Series in Data Management Systems, 2003.
13. [SWEB04] Software Body of Knowledge (SWEBOK), version 2004 www.swebok.org/ironman/pdf/SWEBOK_Guide_2004.pdf, [consultado 31 de enero 2008].
14. [W3C] Worl Wide Web Consortium (W3C), www.w3.org/ [consultado 17 de abril de 2008].
15. [Yansha] Yansha department store embraces supplier collaboration to streamline processes
<ftp://ftp.software.ibm.com/software/solutions/pdfs/ODC03019-USEN-00.pdf>

Capítulo 8: Anexos

Anexo A: Plantillas de entregables de la Extensión SOA

8.1 Plantillas de la Disciplina Modelado del Negocio

Plantilla del entregable Evaluación de la Organización Objetivo

[Nombre del proyecto]

Evaluación de la Organización Objetivo

Control de Cambios y Evolución del documento				
Versión <99>	Fecha de Cambio <DD/MM/AA>	Modificado por <Persona que crea o modifica el documento>	Págs. <Número de página modificada>	Resumen del cambio <Describir el cambio, indicar sección o párrafo en la que realizó la modificación>

1 Contexto del Negocio

[En esta sección se debe proveer una breve introducción del dominio del Negocio en que la Organización trabaja.]

2 Factores externos

[En esta sección se describen los factores externos que influyen en la Organización.]

2.1 Clientes

[Aquí se presenta una lista de los clientes de la Organización y sus expectativas sobre los productos. Se debería incluir un resumen de las investigaciones realizadas para comprender la demanda de los clientes en el Negocio.]

2.2 Competidores

[Aquí se presenta una lista de los competidores, si los hay, en el Negocio de la Organización.]

2.3 Otros involucrados

[Aquí se presenta una lista de otros involucrados en el Negocio de la Organización, como proveedores y socios de Negocio.]

3 Factores internos

[En esta sección se presentan los factores internos que inciden en la Organización y su Negocio.]

3.1 Procesos del Negocio

[Aquí se presenta una breve descripción de los procesos del Negocio actuales, principalmente los que están relacionados con el desarrollo a realizar y otros con los cuales existan importantes interrelaciones a tener en cuenta.]

3.2 Herramientas de soporte

[Aquí se presenta una lista de las herramientas de soporte que son actualmente utilizadas en la Organización.]

3.3 Organización interna

[Aquí se describe brevemente la organización interna de la Organización, que roles y equipos tiene actualmente y cuál es su orden jerárquico.]

3.4 Competencias, habilidades y actitudes

[Aquí se presenta un inventario de las competencias, habilidades y actitudes de los individuos en la Organización.]

3.5 Capacidad de cambio en la Organización

[Aquí se presenta una lista de la capacidad de cambio percibida en la Organización.]

4 Conclusiones de la evaluación

[En esta sección se presentan las conclusiones de la evaluación realizada sobre la Organización Objetivo, presentando una lista de las áreas de mayores problemas y oportunidades, sin importar la categoría a la que pertenecen.]

4.1 Áreas problemáticas

[Aquí se presenta un resumen del análisis de las actividades en los procesos del Negocio existentes actualmente.]

4.2 Nuevas tecnologías aplicables

[Aquí se presenta un resumen de las soluciones disponibles que son aplicables a la Organización.]

Plantilla del entregable Modelo de Casos de Uso del Negocio

[Nombre del proyecto]

Modelo de Casos de Uso del Negocio Versión [1.0]

Control de Cambios y Evolución del documento				
Versión <99>	Fecha de Cambio <DD/MM/AA>	Modificado por <Persona que crea o modifica el documento>	Págs. <Número de página modificada>	Resumen del cambio <Describir el cambio, indicar sección o párrafo en la que realizó la modificación>

Índice

1. ACTORES	114
1.1. [ACTOR 1]	114
1.2. [ACTOR 2]	114
2. CASOS DE USO	114
2.1. DIAGRAMAS DE CASOS DE USO	114
2.2. [CASO DE USO 1]	114
2.2.1. DESCRIPCIÓN	114
2.2.3. FLUJO DE EVENTOS PRINCIPAL	114
2.2.4. FLUJOS DE EVENTOS ALTERNATIVOS	114
2.2.6. REQUERIMIENTOS ESPECIALES	115
2.3. [CASO DE USO 2]	115

1. Actores del Negocio

[En esta sección se debe describir a cada uno de los actores del Negocio que participan en los Casos de Uso del Negocio, un actor del Negocio es un usuario del Negocio o cualquier entidad que interactúa con el Negocio.]

1.1. [Actor del Negocio 1]

[Descripción del Actor 1 del Negocio]

1.2. [Actor del Negocio 2]

[Descripción del Actor 2 del Negocio]

2. Casos de Uso del Negocio

2.1. Diagramas de Casos De Uso

[Aquí se presentan los Diagramas de Casos de Uso del Negocio, para mostrar la interacción entre los Actores y los Casos de Uso del Negocio.]

2.2. [Caso de Uso del Negocio 1]

2.2.1. Descripción

[Explicar brevemente el propósito del caso de uso del Negocio]

2.2.2. Flujo de eventos principal

[En esta sección se realiza una descripción textual del flujo del Negocio que representa el Caso de Uso. El flujo debe describir que hace el Negocio para proveer de valor a un actor del Negocio, pero no como hace el Negocio para resolver sus problemas. Las alternativas simples se pueden presentar dentro del texto del Caso de Uso. Si el flujo alternativo es más complejo, use una sección separada para describirlo.]

2.2.3. Flujos de eventos alternativos

1.1.1.1. [Flujo alternativo 1]

[En esta sección se describen las alternativas más complejas a las cuales se hacía referencia en el Flujo de eventos principal. Estos flujos alternativos representan la conducta alternativa normalmente debida a excepciones que ocurren en el flujo principal. Ellos pueden ser necesarios para describir los eventos asociados con la conducta alternativa. Cuando finaliza el flujo alternativo, se retoman los eventos del flujo de eventos principal a menos que se establezca otra cosa.]

[Sub-flujo alternativo 1]

[Los flujos alternativos pueden dividirse en subsecciones si esto aporta claridad]

[Sub-flujo alternativo 2]

...

1.1.1.2. [Flujo alternativo 2]

...

2.2.4. Diagrama de Actividad

[En esta sección incluye un Diagrama de Actividad que modele en forma gráfica los flujos del Caso de Uso del Negocio descritos previamente. Es importante que se muestren los distintos condicionales que existen en el proceso que determinan

que se siga un camino u otro en la ejecución del proceso, y cuales son todas las opciones de terminación para los flujos modelados.]

2.2.5. Requerimientos especiales

[En esta sección se especifican los requerimientos especiales que son específicos a un caso de uso del Negocio, que son las características del caso de uso del Negocio no cubiertas por los flujos descritos.]

- [Requerimiento especial 1]
- ...

2.3. [Caso de Uso del Negocio 2]

...

3. Reglas del Negocio

[En esta sección se especifican las Reglas del Negocio que se deben tener en cuenta en la ejecución de los distintos procesos definidos. Las Reglas del Negocio son tipos de requerimientos sobre como el Negocio, incluyendo sus herramientas, debe funcionar.

Pueden ser leyes y regulaciones impuestas al Negocio entre otras, y pueden ser clasificadas de distinta forma según el Negocio, por ejemplo pueden estar agrupadas por dominio, usuario o grupo de productos, aunque una clasificación más formal las define como Restricciones y Derivaciones.

Las Reglas del Negocio deben ser expresadas con un nivel de formalismo que permita su automatización, una forma puede ser utilizando el Object Constraint Language (OCL) especificado en UML. De todas formas es útil mostrar las Reglas de Negocio como notas de texto en los elementos de los distintos diagramas y modelos, por ejemplo en los Diagramas de Actividad.]

3.1. [Regla del Negocio 1]

...

8.2 Plantillas de la Disciplina Diseño Extensión SOA

Plantilla del entregable Modelo de Servicios

[Nombre del proyecto]

Modelo de Servicios Versión [1.0]

Control de Cambios y Evolución del documento				
Versión <99>	Fecha de Cambio <DD/MM/AA>	Modificado por <Persona que crea o modifica el documento>	Págs. <Número de página modificada>	Resumen del cambio <Describir el cambio, indicar sección o párrafo en la que realizó la modificación>

Índice

1. IDENTIFICACIÓN Y CATEGORIZACIÓN DE SERVICIOS	118
1.1. [SERVICIO1]	118
1.1.1. <i>Funcionalidades provistas</i>	118
1.1.2. <i>Categorización</i>	118
1.1.3. <i>Caso/s de Uso asociado/s</i>	118
1.1.4. <i>Subsistema de Diseño asociado</i>	118
1.2. [SERVICIO2]	118
2. ESPECIFICACIÓN DE SERVICIOS	118
2.1. [SERVICIO1]	118
2.1.1. <i>Contrato funcional</i>	118
2.1.1.1. [Método1]	118
2.1.1.2. [Método2]	119
2.2. CONTRATO FUNCIONAL	119
3. SERVICIOS EXISTENTES	119
3.1. [SERVICIO1]	119
3.1.1. <i>Funcionalidades provistas</i>	119
3.1.2. <i>Categorización</i>	119
3.1.3. <i>Contrato funcional</i>	119
3.1.3.1. [Método1]	119
3.1.3.2. [Método2]	120
3.2. [SERVICIO2]	120

1. Identificación y categorización de servicios

[En esta sección se listan los servicios identificados indicando el tipo de servicio correspondiente. Los servicios pueden clasificarse como básicos, intermediarios, centrados en procesos y públicos empresariales. Se debe describir para cada servicio, en lenguaje natural de forma clara y sin ambigüedades, la justificación de porqué se define ese servicio, que funcionalidades provee indicando a que Caso de Uso corresponden, y que tipo de servicio es, indicando también que subsistema debe proveer este servicio.]

1.1. [Servicio1]

1.1.1. Funcionalidades provistas

[Se describe en lenguaje natural las funcionalidades generales provistas por el servicio1 identificado.]

1.1.2. Categorización

[Se indica que tipo de servicio es el servicio1 identificado, de acuerdo a la clasificación de servicios indicada en el documento GuíaServicios.pdf provisto en la sección Material.]

1.1.3. Caso/s de Uso asociado/s

[Se identifican los Casos de Uso que definen las funcionalidades que provee el servicio1 identificado.]

1.1.4. Subsistema de Diseño asociado

[Se identifica el Subsistema de Diseño que provee el servicio1 identificado.]

1.2. [Servicio2]

.....

2. Especificación de servicios

[Para cada servicio identificado se realiza su contrato funcional, especificando para cada interface definida los métodos que deben ser implementados para proveer el servicio acordado para la interface.]

2.1. [Servicio1]

2.1.1. Contrato funcional

[Para cada operación definida en la interface que compone el contrato funcional, se debe especificar la información que se indica a continuación.]

2.1.1.1. [Método1]

Método	[nombre del método]		
Parámetros	[nombre del parámetro]	[tipo del parámetro]	[descripción del parámetro]
Valor de Retorno	[nombre del valor retornado]	[tipo del valor retornado]	[descripción del valor retornado]
Excepciones	[Excepción 1 que levanta el método]		
	[Excepción 2 que levanta el método]		
Descripción	[Breve descripción en lenguaje natural de lo que realiza el		

	método]
Precondiciones	[Breve descripción en lenguaje natural de lo que se asume al invocar al método]
Postcondiciones	[Breve descripción en lenguaje natural de lo que se asume una vez ejecutado el método]

2.1.1.2. [Método2]

.....

2.2. Contrato funcional

.....

3. Servicios existentes

[En esta sección se listan los servicios existentes en la Organización identificados, tanto básicos como de otros niveles, que sirvan para reutilizar en la aplicación en desarrollo. No todos los servicios existentes en la Organización serán reutilizados en la aplicación en desarrollo, los que no se usen no se listarán en esta sección. Puede suceder que se deban implementar servicios intermediarios para reutilizar otros servicios que no se adapten totalmente a los requerimientos de la aplicación lo cual será preferible a implementar nuevamente el servicio desde cero. Los servicios básicos de acceso a datos, autenticación, seguridad, sesión, etc. deberán ser reutilizados por todos los servicios definidos en niveles superiores de la SOA.]

3.1. [Servicio1]

3.1.1. Funcionalidades provistas

[Se describe en lenguaje natural las funcionalidades generales provistas por el servicio1 existente identificado.]

3.1.2. Categorización

[Se indica que tipo de servicio es el servicio1 existente identificado, de acuerdo a la clasificación de servicios indicada en el documento GuíaServicios.pdf provisto en la sección Material.]

3.1.3. Contrato funcional

[Para cada operación definida en la interface que compone el contrato funcional, se debe especificar la información que se indica a continuación.]

3.1.3.1. [Método1]

Método	[nombre del método]		
Parámetros	[nombre del parámetro]	[tipo del parámetro]	[descripción del parámetro]
Valor de Retorno	[nombre del valor retornado]	[tipo del valor retornado]	[descripción del valor retornado]
Excepciones	[Excepción 1 que levanta el método]		
	[Excepción 2 que levanta el método]		
Descripción	[Breve descripción en lenguaje natural de lo que realiza el método]		

Precondiciones	[Breve descripción en lenguaje natural de lo que se asume al invocar al método]
Postcondiciones	[Breve descripción en lenguaje natural de lo que se asume una vez ejecutado el método]

3.1.3.2. [Método2]

.....

3.2. [Servicio2]

.....

Anexo B: Guía para la clasificación de servicios en SOA

Introducción:

El foco de una SOA está en la infraestructura funcional y sus servicios del negocio, no en la infraestructura técnica y sus servicios técnicos. La entidad básica es la función o proceso del negocio.

Los servicios se clasifican en distintos tipos con determinadas características que tienen distinto significado desde el punto de vista del diseño, implementación y gestión del proyecto. Estas características son significativamente distintas en cuanto a reusabilidad, mantenibilidad, escalabilidad y performance de los servicios.

La primera división que se hace en una SOA es en dos partes bien diferenciadas: application frontend y backend. Las application frontends no son servicios pero si son elementos activos de la SOA, inician todos los procesos del negocio y reciben los resultados, típicos ejemplos son GUI y procesos batch. En el backend si hay servicios que se clasifican en cuatro tipos de servicios distintos que se describen a continuación: básicos, intermediarios, centrados en procesos y empresariales públicos.

Descripción:

A continuación se describe la clasificación de servicios indicando las características que tiene cada tipo de servicios.

SERVICIOS BASICOS:

Los servicios básicos son la base de SOA, son servidores puros y no mantienen el estado conversacional de la sesión. Se dividen en servicios centrados en los datos y centrados en la lógica.

Servicios centrados en los datos:

Su propósito es manejar los datos persistentes, almacenamiento, recuperación, mecanismos de lockeo y gestión transaccional. Se comportan similares a la capa de acceso a datos de una aplicación tradicional. La diferencia es que la capa de acceso a datos maneja los datos de la aplicación entera, mientras que un servicio centrado en los datos trata con una sola entidad del negocio. Cualquier otro servicio que requiera acceso a estos datos debe usar la interface del servicio, por lo que una aplicación podría requerir entonces varios servicios centrados en los datos coordinados. El problema con este enfoque es que en general en la Base de Datos se tienen tablas que se comunican y se necesitan para manejar los datos de la aplicación.

Servicios centrados en la lógica:

Estos servicios proveen encapsulamiento para cálculos complejos o reglas del Negocio, tradicionalmente encapsulados en bibliotecas y frameworks del negocio. Por

ejemplo un motor de productos de seguros que encapsula el conocimiento, términos y condiciones de los productos de una compañía de seguros. Esta funcionalidad podría operarse desde central, ser integrada con sistemas de IT internos a la empresa de los distintos departamentos en la misma, y con socios de negocio fuera de la empresa. Pueden utilizarse también distintas facades que provean las distintas vistas para los distintos tipos de usuarios (ej. agencias independientes vs. depto. de reclamos).

SERVICIOS INTERMEDIARIOS:

Son servicios sin estado que hacen de puente entre las inconsistencias técnicas o discrepancias conceptuales en el diseño. Son tanto clientes como servidores en la SOA, mediando entre los distintos elementos que deben funcionar juntos, pero son distintos que los servicios técnicos de infraestructura ya que éstos proveen una API técnica mientras que los intermediarios proveen una API orientada al Negocio. Se dividen a su vez en technology gateways, adapters, facades y functionality-adding.

Technology gateways:

Estos servicios hacen de puente entre discrepancias tecnológicas, incorporando dos o más tecnologías para comunicación o codificación de datos. Hacen de proxys para sus servicios de Negocio y representan la funcionalidad de los servicios que están por debajo en un ambiente que es distinto tecnológicamente que el ambiente de ejecución del servicio de Negocio original. En la figura se ve la Aplicación 1 en la tecnología A que accede al Servicio Básico en la tecnología B, a través del technical gateway que traduce la tecnología B en la A para permitir el acceso al Servicio Básico.

Adapters:

Es un tipo especial de servicio intermediario que mapea las firmas y formatos de de mensaje de un servicio a los requerimientos de un cliente. Por ejemplo, si se tiene dos servicios para el mismo concepto de distintas compañías, con un adapter para cada uno en cada sentido de los pedidos, se mapean los requerimientos de la aplicación que usa un servicio en el otro y viceversa. En la figura se puede ver esta situación, donde se agregan los adapters A1 y A2 para permitir el acceso a los servicios básicos S1 y S2.

Facades:

Su propósito es proveer una vista distinta (y posiblemente agregada) de uno o más servicios existentes, por lo que también pueden actuar como technology gateways y/o adapters. Como el patrón de Gamma, una facade puede utilizarse para proveer una vista específica de un conjunto de servicios que se encuentran por debajo, que en general son servicios básicos. Puede actuar entonces como capa para simplificar el acceso a servicios de un proyecto, aunque este uso puede generar una capa monolítica de acceso a los distintos servicios lo que dificultará el reuso de los servicios que están por debajo por parte de otros proyectos. En el ejemplo se puede ver como el servicio facade oculta de las aplicaciones que lo acceden los distintos servicios que proveen las funcionalidades requeridas.

Functionality - adding:

Este tipo de servicio se usa cuando se quiere agregar funcionalidad a un servicio existente sin cambiar el servicio mismo. En este caso se crea un servicio que provea la misma funcionalidad del servicio original y agregue las nuevas características requeridas. Entre los escenarios de uso para este tipo de servicio se pueden mencionar el de utilización de servicios de terceras partes sin código disponible, como primer paso en la reimplementación de un servicio existente, servicio original está en desarrollo por una parte del equipo u otro proyecto en la empresa.

SERVICIOS CENTRADOS EN PROCESOS

Estos servicios encapsulan el conocimiento de los procesos del Negocio de la Organización, controlan y mantienen el estado del proceso en ejecución. Desde el punto de vista técnico son la clase de servicios más sofisticada. Actúan tanto de clientes como de servidores en una SOA, manteniendo el estado del proceso para sus clientes. Una de las principales ventajas que presentan estos servicios es que separan la lógica de procesos, asignando la lógica del Negocio núcleo en los servicios básicos, el control de diálogo en la aplicación de frontend y la lógica de los procesos en estos servicios centrados en procesos. Estos son mayormente específicos del proyecto definiendo los procesos del Negocio y su control, en base a orquestación de los servicios existentes.

Es en este tipo de servicios donde aparecen los BPMS (Business Process Management System) y la utilización de lenguajes de modelado de procesos como BPML (Business Process Modeling Language) y BPEL4WS (Business Process Execution Language for Web Services) para orquestación de los servicios utilizados para llevar adelante el proceso definido. Sin embargo los servicios centrados en procesos podrían no existir siendo suplantados por una capa de control de procesos en el application frontend que encapsula la definición del proceso, aunque es deseable que si estén presentes y modelados en un BPMS. En las figuras se ven los dos casos planteados, con y sin servicios centrados en procesos. En el primer caso el Servicio centrado en procesos encapsula la orquestación de los servicios necesarios para realizar el proceso.

SERVICIOS EMPRESARIALES PUBLICOS

Los tipos de servicio anteriores son solamente para uso dentro de los límites de una empresa en particular. Los servicios empresariales públicos son servicios que una empresa ofrece a socios y clientes para su consumo. Por ejemplo, proveer servicios para que los clientes hagan seguimiento de sus envíos en la empresa que los hace, proveer funcionalidad para enviar mensajes SMS de forma que las empresas puedan incorporarla a sus sistemas propios. Estos servicios tienen requerimientos específicos de interface, desacoplamiento, seguridad y facturación de uso, ya que las empresas deben acordar claramente como se realiza el uso de los mismos.