

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 3



TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO INFORMÁTICO

TÍTULO: Propuesta de una guía para adoptar una
Arquitectura Orientada a Servicios en la UCI.

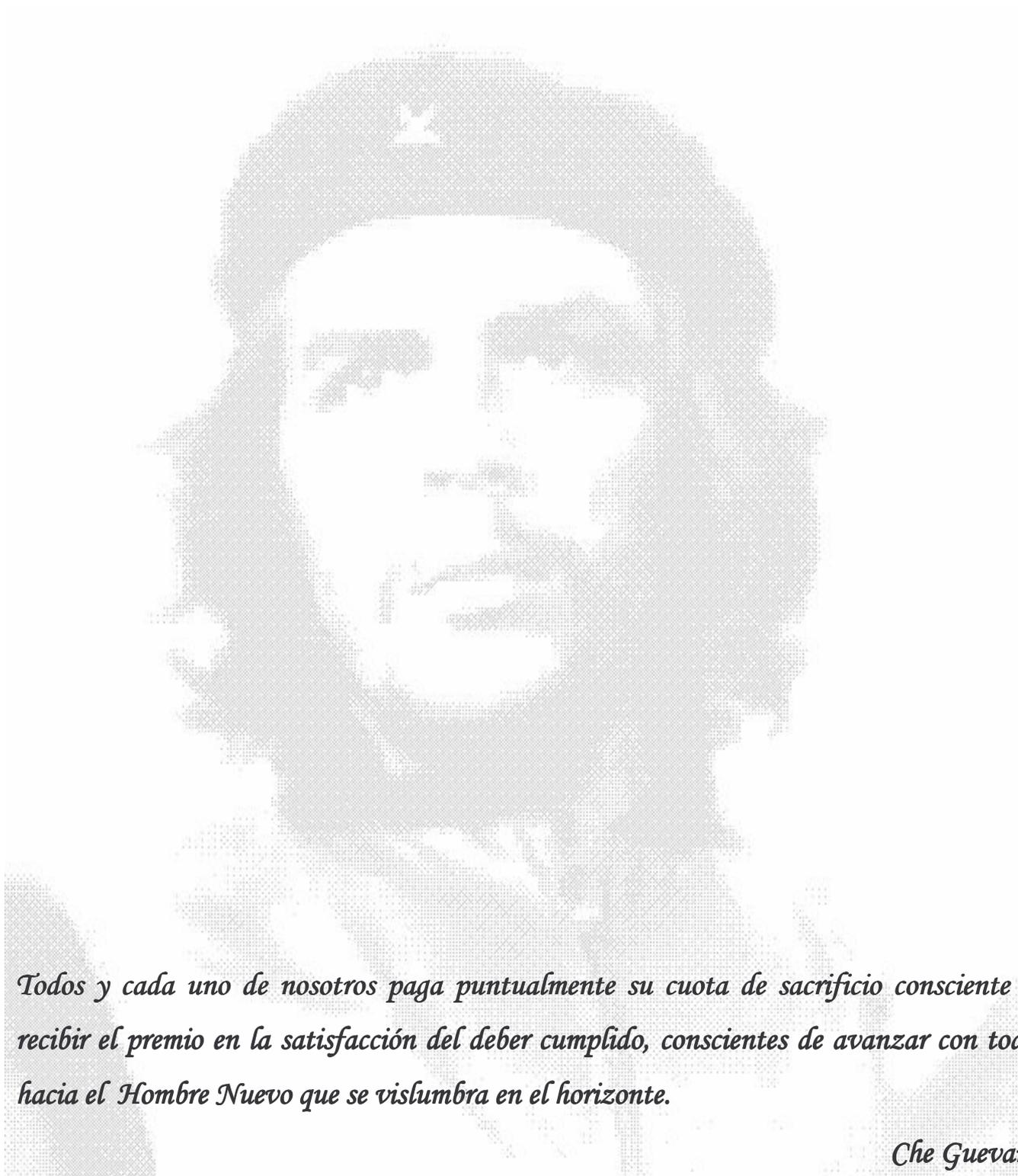
AUTOR(ES): Annerys Sánchez Lamar.

Kenia Estela Lamoth Malo de Molina.

TUTOR: Ing. Geykel Raúl Moreno Ceballos.

CONSULTANTE: Ing. Manuel Alejandro Gil Martín.

16 de Junio de 2008



Todos y cada uno de nosotros paga puntualmente su cuota de sacrificio consciente de recibir el premio en la satisfacción del deber cumplido, conscientes de avanzar con todos hacia el Hombre Nuevo que se vislumbra en el horizonte.

Che Guevara.

DECLARACIÓN DE AUTORÍA

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

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

Kenia Estela Lamothe Malo de Molina

Annerys Sánchez Lamar

Ing, Geykel Raúl Moreno Ceballos

DATOS DE CONTACTO

Síntesis del tutor Ing. Geykel Raúl Moreno Ceballos

Profesión: Ing. Informático

Categoría Docente: Adiestrado

Años de graduado: 2

Síntesis del consultante Ing. Manuel Alejandro Gil Martín

Profesión: Ing. Informático

Categoría docente: Adiestrado

Años de graduado: 3

AGRADECIMIENTOS

*Agradecemos primeramente a la Revolución
y a nuestro Comandante por habernos dado la posibilidad
de estudiar y superarnos.*

*A la Universidad de las Ciencias Informáticas
por formarnos y educarnos durante estos años.*

*Agradecemos a Manuel Alejandro Gil Martín
por toda la ayuda que nos prestó,
por todo el tiempo que nos dedicó, por ser ante todo un amigo incondicional,
porque sin él no hubiéramos podido hacer realidad este trabajo.*

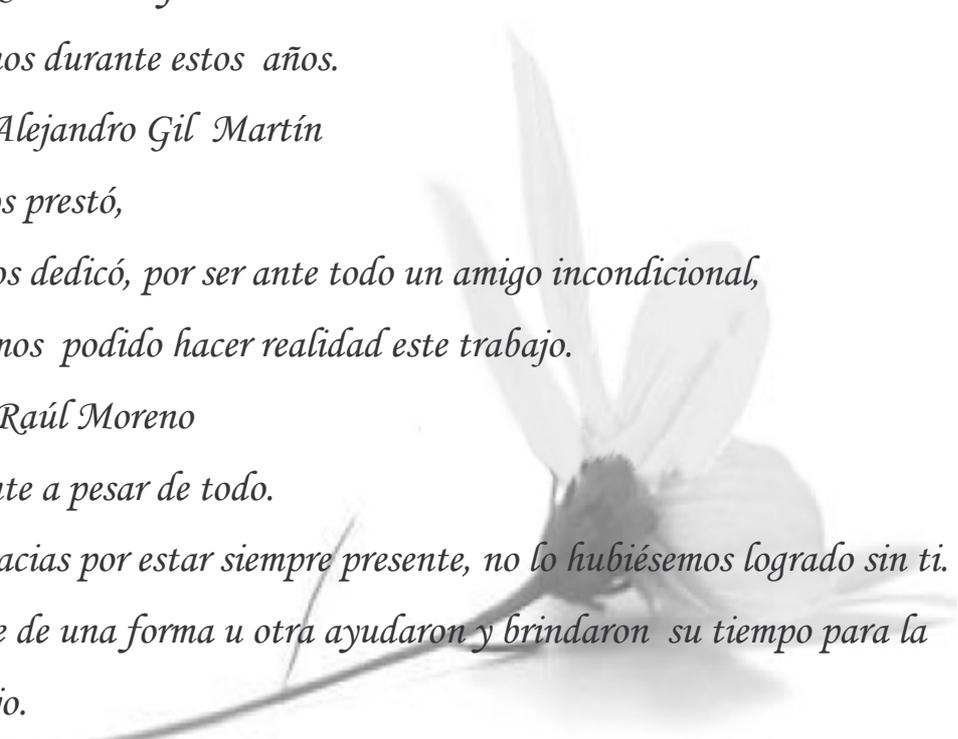
*A nuestro tutor Geykel Raúl Moreno
por estar siempre presente a pesar de todo.*

A Wilkie J. Zaldívar, gracias por estar siempre presente, no lo hubiésemos logrado sin ti.

*A todas las personas que de una forma u otra ayudaron y brindaron su tiempo para la
realización de este trabajo.*

Muchas gracias por todo.

Annerys y Kenia.



DEDICATORIA

*Quiero dedicar este trabajo de diploma a la persona que más quiero en el mundo, a quien le debo todo lo que soy y seré a partir de ahora,
no tengo palabras para agradecerte todo lo que haz hecho por mí y solo puedo decirte que te quiero muchísimo,*

esta tesis es para mi mamá Mercedes de la Caridad Malo de Molina Gainza.

También quiero dedicarle esta tesis a mi familia,

a mis abuelos papi y mami,

a mis hermanas Hortensia y Lisandra,, a mi padrastro Julio,

a todos gracias por apoyarme y confiar en mí..

A mi amiga y compañera de tesis Annerys,

gracias por toda la ayuda que me brindaste durante la tesis,

gracias por estar ahí siempre que lo necesité.

A mi novio Alexander por compartir conmigo estos años en la universidad,

por haberme ayudado a pasar por momentos muy difíciles durante la carrera

y por estar siempre a mi lado.

A mis amigas de la UCI.

Kenia

DEDICATORIA

*Dedico esta tesis primeramente a mis abuelos Romelia y Danilo,
por ser mis mejores guías ,por apoyarme en todo,
sin ustedes nunca hubiese logrado nada.*

*A mi madre, por confiar siempre en mí, por ayudarme a ser quien soy, por ser mi mejor
confidente, por estar siempre a mi lado.*

A mi padre, por sus consejos, su cariño, su atención y su amor.

A pipo, por quererme como una hija.

A mi hermano, por estar siempre unidos en todo.

*A Magda, donde quiera que estés, te llevo siempre en mi corazón y te doy las gracias por
haber sido como una madre para mí, por ayudarme en todo,
por haber sido tan especial como eras; nunca te olvidaré.*

*A Celia, mi amiga del alma, por todos los momentos buenos
y malos que hemos pasado juntas,
porque nunca nos separaron las distancias, por estar siempre ahí.*

*A Wilkie, por ser mi compañero, mi pareja, mi amor, por aguantarme, por ayudarme en
todo, por ser tan especial, porque te quiero.*

*A Kenia, por ser mi compañera de tesis, amiga incondicional durante cinco años, por
aguantar mi mala cara, por las noches sin dormir.*

A Jesús y Rita, sin ustedes no lo hubiera podido lograr.

A mis amigas de la UCI.

Annerys

RESUMEN

Con el fin conocer cómo se puede adoptar una Arquitectura Orientada a Servicios (SOA) en la Universidad de las Ciencias Informáticas (UCI) fue realizada la presente investigación. El trabajo analiza los principales conceptos relacionados con el objeto de estudio para una mejor comprensión por parte del lector. Explica cuáles son las características fundamentales de una SOA, profundizando en las ventajas que reporta la implantación de este tipo de arquitectura en la UCI.

La investigación incluye un estudio de los enfoques SOA que brindan las empresas líderes en el mercado mundial de software, específicamente las metodologías y buenas prácticas que proponen para el logro de una correcta orientación a los servicios y culmina con una propuesta de una guía para la adopción de una SOA, la cual está basada en las condiciones actuales que presenta la UCI.

Palabras claves: Arquitectura Orientada a Servicios, guía.

INDICE

INTRODUCCIÓN.....	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	6
1.1. Introducción.	6
1.2. Breve historia de la arquitectura de software.	6
1.3. Conceptualización General.	9
1.3.1 ¿Qué es la Arquitectura de Software?	9
1.4. Arquitectura Orientada a Servicios.	10
1.4.1. Antecedentes.	10
1.4.2. ¿Qué es SOA?	11
1.4.3. Elementos que conforman una SOA.....	12
1.4.3.1. Consumidores de servicios.	12
1.4.3.2. Servicios.....	12
1.4.3.2.1. Clasificación de los servicios.....	13
1.4.3.3. UDDI.....	14
1.4.3.4. ESB	16
1.4.3.5. BPM y SOA.	19
1.4.5. Beneficios de SOA.	24
1.4.6. SOA y Servicios Web.....	26
1.5. Metodologías de desarrollo.	28
1.5.1. ¿Qué es una metodología?.....	28
1.5.2. Rational Unified Process (RUP).....	29
1.5.3. Extreme Programing (XP)	31
1.5.4. Microsoft Solution Framework (MSF).....	32
1.6. Conclusiones Parciales.	33
CAPÍTULO 2: SOLUCIÓN PROPUESTA	34
2.1. Introducción.	34
2.2. Buenas prácticas desarrolladas por las empresas líderes para SOA.....	34
2.2.1. Software Associates.....	34
2.2.2. Microsoft.....	37

2.2.3.	IBM.....	39
2.2.4.	Software AG.....	43
2.3.	Metodología RUP para SOA.....	45
2.3.1.	Extensión al proceso base para desarrollo SOA.....	46
2.3.1.1.	Disciplina Modelado del Negocio.....	46
2.3.1.2.	Disciplina de Diseño.....	48
2.3.1.3.	Disciplina Implementación.....	50
2.4.	Resultado de las entrevistas.....	51
2.4.1.	Técnica para la obtención de los datos.....	51
2.4.2.	Procesamiento de la información.....	52
2.5.	Solución Propuesta.....	54
2.5.1.	Necesidad de adoptar una SOA.....	54
2.5.2.	Necesidad de una guía.....	55
2.5.3.	Propuesta de una guía para la adopción de SOA.....	56
2.5.4.	Guía para la adopción de SOA.....	56
2.5.4.1.	1ra Etapa: Inicio.....	57
2.5.4.2.	2da Etapa: Composición y despliegue.....	59
2.5.4.5.	3ra Etapa: Gobierno.....	62
2.6.	Conclusiones Parciales.....	65
CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA.....		66
3.1.	Introducción.....	66
3.2.	Método de Evaluación de Expertos.....	66
3.2.1.	Antecedentes históricos.....	66
3.2.2.	Características del método Delphi.....	66
3.3.	Fases del método Delphi.....	67
3.3.1.	Elaboración del objetivo.....	67
3.3.2.	Selección de los especialistas.....	67
3.3.3.	Elaboración del cuestionario.....	67
3.3.4.	Resultados de la evaluación.....	68
3.4.	Conclusiones.....	71
CONCLUSIONES.....		72

RECOMENDACIONES.....	73
BIBLIOGRAFÍA.....	74
BIBLIOGRAFÍA CONSULTADA.....	77
ANEXOS.....	78
GLOSARIO.....	84

Índice de Figuras

Fig 1.1 Representación de un servicio y sus componentes.

Fig 1.2 Representación de la interacción de un repositorio de servicios con los consumidores.

Fig 1.3 Representación de la integración mediante un ESB.

Fig 1.4 Representación de conexiones punto a punto y conexiones mediante un ESB.

Fig 1.5 Relación de los niveles de Madurez de SOA con los de CMMI de la versión escalonada.

Fig 2.1 Propuesta de Software Associates.

Fig 2.2 Propuesta SOA de Microsoft.

Fig 2.3 Prácticas SOA de IBM.

Índice de Gráficas

Fig 2.1 Representación en % del criterio sobre el nivel de madurez en que se encuentra la UCI.

Fig 2.2 Relación en % del criterio sobre la existencia de procesos descritos en la UCI.

Fig 2.3 Relación en % del criterio sobre la integración de las áreas de trabajo en la UCI.

Fig 3.1 Tabla: Pasos propuestos para la adopción de una SOA.

Fig 3.2 Distribución radial de la importancia dada a los pasos para adoptar una SOA.

INTRODUCCIÓN

El desarrollo de la computación ha propiciado su integración en los diferentes sectores de la sociedad y a pesar de que en los inicios no se contaba con una metodología coherente y robusta para desarrollar los programas, surge la demanda de software con el propósito de resolver innumerables problemas como el procesamiento de datos, la automatización de procesos complejos, entre otros.

Sin duda, los primeros software que fueron desarrollados no poseían mucha complejidad pues se hacían sin ninguna planificación y solo con el objetivo de resolver problemas sencillos. La gran mayoría de los programas se realizaban para determinadas empresas o para una persona específica, en caso de que estos fallaran se depuraba el error sin una documentación que llevara constancia del desarrollo del programa lo cual le daba al software un entorno personalizado.

Con el tiempo crece la demanda de software para todo tipo de instituciones, organizaciones y sectores, de esta forma aumenta la complejidad de las operaciones que deben realizar y dado al gran avance de la informática, se hace necesario comenzar a pensar en la búsqueda de técnicas para mejorar la calidad y el funcionamiento de los mismos.

Surgen así nuevos paradigmas y conceptos que se toman en cuenta por su grado de importancia, siendo uno de ellos el de Arquitectura de Software, el cual establece los fundamentos para crear aplicaciones mejor estructuradas y pensadas, brindando conceptos y aportando elementos a tener en cuenta para el buen desarrollo del software.

Con la utilización del software como un negocio surge la competencia y con esta los cambios continuos en las demandas y los requisitos de los usuarios creando la necesidad de tener aplicaciones listas para los clientes y cumplir sus expectativas respondiendo a los cambios continuos sin necesidad de desarrollar un nuevo software o reescribir totalmente el existente, lo que da lugar a una evolución de la arquitectura.

Las necesidades actuales que tiene toda organización para lograr sus objetivos demandan la construcción de complejos sistemas que requieren la combinación de diferentes tecnologías y plataformas de hardware o software, esto exige que los profesionales presten atención al diseño de una arquitectura bajo la cual este soportado el funcionamiento de sus sistemas.

Cada vez más las organizaciones dependen de su infraestructura de tecnologías de la información (IT, del inglés Information Technology) para alcanzar sus objetivos. Pero en un entorno competitivo como el actual aprovechar las oportunidades de negocio exige moverse con rapidez. Las empresas necesitan interconectar los procesos, personas e información tanto en la propia institución como para atravesar fronteras. La falta de integración entre los componentes de la IT-sistemas, aplicaciones y datos- hace difícil obtener una respuesta rápida y efectiva ante los cambios que afectan de forma natural los negocios. Esta inflexibilidad genera costes, reduce la capacidad de respuesta ante los clientes y afecta negativamente la productividad de los empleados. En general, una deficiente integración es uno de los problemas más importantes a los que debe enfrentarse toda organización para mantener su competitividad y garantizar su crecimiento.

La Arquitectura Orientada a Servicios (SOA, del inglés Service Oriented Architecture) supone una estrategia general de los elementos de la IT donde los sistemas distribuidos y aplicaciones complejas se pueden transformar en una red de recursos integrados, simplificados y sumamente flexibles. SOA es un paradigma o estilo de arquitectura que permite la creación o los cambios de negocio desde las perspectivas de las IT de una forma fácil a través de la composición de nuevos procesos utilizando las funcionalidades de negocio que están contenidas en las infraestructuras de aplicaciones actuales o futuras. Las compañías e instituciones que adopten SOA como arquitectura obtienen muchos beneficios pues mejoran en los tiempos de realización de cambios en los procesos, aumentan la calidad en los servicios, reducen los costos de mantenimiento y de integración, entre otros.

La Universidad de las Ciencias Informáticas (UCI) no queda al margen en esta cuestión, pues también busca mejoras en la calidad de sus aplicaciones, a la vez que aumenta la demanda

de software en conjunto con los requerimientos de los clientes; esto implica transformaciones en los procesos de negocio haciéndose necesario una mejor orientación al proceso.

Analizando el planteamiento anterior la **situación problemática** que vemos en la UCI es la siguiente:

La Universidad de las Ciencias Informáticas desde su surgimiento hasta la actualidad presenta muchos problemas en los sistemas, estimulado por el continuo crecimiento de su matrícula docente así como de profesores y trabajadores vinculadas a ella, este incremento provocó la necesidad de realizar tecnologías de forma rápida sin el debido estudio profundo de cada una de ellas para resolver los problemas inmediatos que existían. Los sistemas actuales que se encuentran hoy en la universidad trabajan en base a los mismos datos y sin embargo no se encuentran acoplados, por otra parte existen sistemas que guardan estrecha relación uno con el otro, pero están fabricados con diferentes tecnologías, trayendo como consecuencia que no funcionen con la rapidez necesaria y que los datos no estén sincronizados.

A partir de la situación problemática descrita el **problema científico** queda planteado de la siguiente manera:

- No existe una guía en la Universidad de las Ciencias Informáticas que posea los elementos necesarios a tener en cuenta al adoptar una Arquitectura Orientada a Servicios.

Después de haberse elaborado el problema científico se puede plantear como **objeto de estudio**:

- Arquitectura Orientada a Servicios.

Dentro del objeto de estudio se definirá como **campo de acción**:

- Metodologías y buenas prácticas desarrolladas por las empresas líderes para la adopción de una Arquitectura Orientada a Servicios.

Al conocer la situación problemática y el problema científico, el **objetivo general** que se ha trazado es:

- Proponer una guía que contribuya a adoptar una Arquitectura Orientada a Servicios en la UCI.

Las **tareas investigativas** trazadas para alcanzar este objetivo son las siguientes:

- Elaboración del marco teórico de la investigación.
- Determinar las metodologías propuestas o desarrolladas por las empresas líderes.
- Determinar los lineamientos más importantes para la adopción de una SOA.
- Realización de entrevistas para caracterizar de manera general a SOA en la UCI.
- Proponer una guía para adoptar una Arquitectura Orientada a Servicios.
- Investigación sobre el método Delphi para la aceptación y validación de la guía propuesta.
- Realización de cuestionarios para que los expertos evalúen la guía propuesta.

Posibles resultados.

- ◆ Obtener una guía para adoptar una Arquitectura Orientada a Servicios en la UCI.

Para el desarrollo de la investigación del presente trabajo se propone utilizar dos métodos de la investigación: El método teórico y el método empírico.

Dentro del **método teórico** se utilizará el **Histórico- Lógico** mediante el estudio de la teoría que permitirá llegar a las conclusiones a partir de los conocimientos que se adquieran durante la investigación y para analizar la trayectoria por la que atraviesan los elementos que se tratan en el marco teórico, manejándose la evolución de dichos elementos y sus conexiones históricas fundamentales. En este método se utilizarán principalmente la búsqueda y las consultas bibliográficas.

El **método empírico** se utilizará mediante la realización de **entrevistas**, que se harán con el fin de obtener conocimientos acerca de las metodologías existentes para el desarrollo de

software y en especial para SOA. Además para medir la validez de la solución del problema se empleará como método la entrevista de expertos.

La tesis está estructurada por tres capítulos y estos a su vez por epígrafes. A continuación se hace una breve reseña sobre el contenido que se trata en cada uno de ellos.

El **capítulo 1** expone la conceptualización general asociada al problema, una breve historia de la arquitectura de software y sus antecedentes. Se definen los principales conceptos relacionados con SOA.

El **capítulo 2** muestra por qué en el trabajo se le presta mayor atención a las metodologías y buenas prácticas desarrolladas por las empresas líderes. Además se hace una selección de las principales empresas que han adoptado una SOA y han obtenido éxito; se eligen los elementos comunes que estas empresas siguen para adoptar SOA y se hace la presentación de la propuesta que da solución al problema científico, o sea, una guía que establece los elementos generales de las metodologías y buenas prácticas desarrolladas para SOA.

El **capítulo 3** finalmente muestra la validación de la propuesta presentada para dar solución al problema.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Introducción.

Este capítulo está enfocado a ofrecer un acercamiento a SOA y brindar una panorámica del tema, la cual engloba los principales conceptos relacionados a este tipo de arquitectura, los elementos que la conforman y los beneficios que aportaría la adopción de este tipo de arquitectura para el soporte de la infraestructura tecnológica de una organización. También se brinda una pequeña descripción de metodologías de desarrollo de software seleccionadas de acuerdo a su popularidad y usabilidad a nivel mundial.

1.2. Breve historia de la arquitectura de software.

Hasta el momento no se ha escrito una historia aceptable de la Arquitectura de Software. Fueron Mary Shaw o David Garlan los que escuetamente reseñaran la historia a principios de la década de los 90 y desde ese entonces sus párrafos han sido reutilizados una y otra vez en la literatura, pero por esos motivos no se deja de mencionar, puesto que el objetivo de este estudio es señalar las supervivencias y las re-matizaciones que ha sufrido la AS (Arquitectura de Software) contemporánea.

Las AS remontan su surgimiento a partir de la década de 1960, sin embargo su historia no ha sido tan continua como la de la ingeniería del software[1]. Después de las tempranas inspiraciones de Edsger Dijkstra, de David Parnas y de Fred Brooks, la AS quedó en estado de vida latente durante unos cuantos años, hasta comenzar su expansión explosiva con los manifiestos de Dewayne Perry de AT&T Bell Laboratories de New Jersey y Alexander Wolf de la Universidad de Colorado [2]. Puede decirse que Perry y Wolf fundaron la disciplina, y su llamamiento fue respondido en primera instancia por los miembros de lo que podría llamarse la escuela estructuralista de Carnegie Mellon: David Garlan, Mary Shaw, Paul Clements, Robert Allen.

Lo cierto es que cada vez que se narra la historia de la arquitectura de software (o de la ingeniería de software, según el caso), se reconoce que en un principio, hacia 1968, Edsger Dijkstra, de La Universidad Tecnológica de Eindhoven en Holanda y Premio Turing 1972,

propuso que se establezca una estructuración correcta de los sistemas de software antes de lanzarse a programar, escribiendo código de cualquier manera [3]. Dijkstra sostenía que las ciencias de la computación eran una rama aplicada de las matemáticas y que se deberían seguir pasos más formales para descomponer problemas mayores. Estas ideas fueron las bases para que otros como Niklaus Wirth [4] y DeRemer y Kron [5] tuvieran ideas que poco a poco fueron decantando entre los ingenieros y arquitectos después.

Fue en la conferencia de la NATO de 1969 donde P. I Sharp formuló estas sorprendentes apreciaciones:

“Pienso que tenemos algo, aparte de la ingeniería de software: algo de lo que hemos hablado muy poco pero que deberíamos poner sobre el tapete y concentrar la atención en ello. Es la cuestión de la arquitectura de software. La arquitectura es diferente de la ingeniería. Como ejemplo de lo que quiero decir, echemos una mirada a OS/360. Partes de OS/360 están extremadamente bien codificadas. Partes de OS, si vamos al detalle, han utilizado técnicas que hemos acordado constituyen buena práctica de programación. La razón de que OS sea un amontonamiento amorfo de programas es que no tuvo arquitecto. Su diseño fue delegado a series de grupos de ingenieros, cada uno de los cuales inventó su propia arquitectura. Y cuando esos pedazos se clavaron todos juntos no produjeron una tersa y bella pieza de software” [6].

Posteriormente en la década de los años 70 existieron varios precursores como David Parnas, el cual demuestra que los criterios seleccionados en la descomposición de un sistema impactan en la estructura de los programas y propuso diversos principios de diseño que debían seguirse a fin de obtener una estructura adecuada. Luego introdujo el concepto de ocultamiento de la información, unos de los principios fundamentales del diseño de software aún en la actualidad.

No fue hasta la década de los años 80 que los métodos de desarrollo de software estructurados hasta ese momento resultaron no escalar suficientemente, lo que dio lugar a un nuevo paradigma: la programación orientada a objetos.

El primer estudio en que aparece la expresión “arquitectura de software” en el sentido en que hoy lo conocemos es sin duda el de Perry y Wolf ocurrió tan tarde como en 1992, aunque el trabajo se fue gestando desde 1989.

Dando cumplimiento a las profecías de Perry y Wolf, la década de 1990 fue sin duda la de la consolidación y diseminación de la AS en una escala sin precedentes. Las contribuciones más importantes surgieron en torno del instituto de ingeniería de la información de la Universidad Carnegie Mellon. En la misma década, demasiado pródiga en acontecimientos, surge también la programación basada en componentes, que en su momento de mayor impacto impulsó a algunos arquitectos mayores, como Paul Clements [7].

Un segundo gran tema de la época fue el surgimiento de los patrones, cristalizada en dos textos fundamentales, el de la Banda de los Cuatro en 1995 y la serie POSA desde 1996 [8].

De cualquier forma, la AS de este período realizó su trabajo de homogeneización de la terminología, desarrolló la tipificación de los estilos arquitectónicos y elaboró lenguajes de descripción de arquitectura.

Uno de los acontecimientos arquitectónicos más importantes del año 2000 fue la hoy célebre tesis de Roy Fielding que presentó el modelo REST, el cual establece definitivamente el tema de las tecnologías de Internet y los modelos orientados a servicios y recursos en el centro de las preocupaciones de la disciplina [9]. En el mismo año se publica la versión definitiva de la recomendación IEEE Std 1471, que procura homogeneizar y ordenar la nomenclatura de descripción arquitectónica y homologa los estilos como un modelo fundamental de representación conceptual.

Sin dudas la AS es hoy en día un conjunto inmenso y heterogéneo de áreas de la investigación teórica y formulación práctica. La AS comenzó siendo una abstracción descriptiva puntual que en los primeros años no investigó de manera sistemática ni las relaciones que la vinculaban con los requerimientos previos, ni los pasos metodológicos a dar luego para comenzar a componer el diseño. Pero esa sincronicidad estructuralista no pudo

sostenerse. Por el contrario, daría la impresión que, a medida que pasa el tiempo, la AS tiende a redefinir todos y cada uno de los aspectos de la disciplina madre, la ingeniería de software, sólo que a un mayor nivel de abstracción y agregando una nueva dimensión reflexiva en lo que concierne a la fundamentación formal del proceso.

1.3. Conceptualización General.

1.3.1 ¿Qué es la Arquitectura de Software?

No es novedad que ninguna definición de la AS es respaldada unánimemente por la totalidad de los arquitectos. El número de definiciones circulantes alcanza un orden de tres dígitos, amenazando llegar a cuatro. De hecho, existen grandes compilaciones de definiciones alternativas o contrapuestas, como la colección que se encuentra en el SEI (SEI, del inglés Software Engineering Institute), a la que cada quien puede agregar la suya. [10]

Una definición reconocida es la de Clements [11]: La AS es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones.

En una definición tal vez demasiado amplia, David Garlan [12] establece que la AS constituye un puente entre el requerimiento y el código, ocupando el lugar que en los gráficos antiguos se reservaba para el diseño.

La definición "oficial" de AS se ha acordado que sea la que brinda el documento de IEEE Std 1471-2000,[13] adoptada también por Microsoft, que reza así:

"La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución."

De los conceptos antes expuestos podemos decir que la AS es la parte de la ingeniería de software que se encarga de la descripción de la estructura de un sistema como una serie de componentes, con el objetivo de organizar los distintos subsistemas adecuadamente que conforman el sistema y permitir la integración de diversos grupos de desarrollo en un mismo proyecto.

1.4. Arquitectura Orientada a Servicios.

1.4.1. Antecedentes.

En las últimas décadas los departamentos de IT de las empresas han construido una infraestructura que actualmente soporta en gran medida la operación de sus empresas y sus clientes. El resultado de este proceso, ha sido la creación y mantenimiento de un número considerable de aplicaciones al interior de las empresas, cada una responsable de sus propias tareas.

Los negocios exigen crear aplicaciones cada vez más complejas, con menos tiempo y presupuesto que antes. En muchos casos crear estas aplicaciones requiere de funcionalidades ya antes implementadas como parte de otros sistemas. En este punto los arquitectos de software se pueden enfrentar a dos opciones:

- ◆ **Tratar de reutilizar la funcionalidad ya implementada en otros sistemas.** Una labor difícil de realizar, debido a que estos no fueron diseñados para integrarse o sobre plataformas y/o tecnologías incompatibles entre ellas.

- ◆ **Re-implementar la funcionalidad requerida ("reinventar la rueda").** Aunque implica más tiempo de desarrollo, es en la mayoría de los casos la más fácil y segura.

Aunque no sea la más acertada a largo plazo, la segunda opción es la más escogida. Esto trae como resultado:

- Funcionalidad replicada en varias aplicaciones.
- Dificultad de migración de los sistemas internos, al haber múltiples conexiones desde sistemas que dependen de estos para su funcionamiento.

- Al no haber una estrategia de integración de aplicaciones, se generan múltiples puntos de falla, que pueden detener la operación de todos los sistemas muy fácilmente.
- Un modelo así, por lo general no escala muy bien.

El inconveniente final es una pobre respuesta al cambio. Las aplicaciones siguen siendo concebidas desde un principio como islas independientes.

1.4.2. ¿Qué es SOA?

En los últimos tiempos se ha hablado mucho de SOA y cada vez más las empresas están concientizadas de la necesidad de incorporar SOA a su infraestructura de tecnologías de información. Sin embargo muchos expertos en SOA tienen definiciones ligeramente diferentes. Algunos conceptos que a continuación se muestran ayudan a comprender SOA.

- ◆ El concepto SOA hace referencia a un enfoque de arquitectura cuyo objetivo es la creación de sistemas a partir de servicios autónomos. Con SOA, la integración pasa a ser una reflexión previa más que una idea posterior. Probablemente, la solución final esté formada por servicios desarrollados en distintos lenguajes de programación y se aloje en plataformas diferentes con numerosos modelos de seguridad y procesos empresariales[14].

- ◆ SOA: Es una estrategia en la cual las aplicaciones hacen uso de los servicios disponibles en la red. Mediante SOA se pueden desarrollar aplicaciones que utilicen servicios o que se comporten como un servicio para que otros accedan a ellas o ambos. [15]

- ◆ SOA es una arquitectura de aplicación en la cual todas las funciones se definen como servicios independientes con interfaces invocables bien definidas, que pueden ser llamadas en secuencias definidas para formar procesos de negocio.[16]

De estos conceptos se infiere que SOA es una arquitectura que fomenta la creación de aplicaciones que se adaptan fácilmente a los inevitables y frecuentes cambios del negocio. SOA es tanto un marco de trabajo como un marco de implementación para el desarrollo de los procesos de negocio de una organización, es utilizada con fines organizativos. Propone

desglosar las funcionalidades principales de las aplicaciones en servicios autónomos y luego orquestarlos para formar nuevas aplicaciones de negocio.

1.4.3. Elementos que conforman una SOA.

1.4.3.1. Consumidores de servicios.

Son aquellos elementos de la arquitectura SOA que son capaces de descubrir servicios a través de un repositorio o hacen llamadas a los repositorios de acuerdo al contrato y a través de una interfaz definida a tal efecto.

1.4.3.2. Servicios.

Los servicios representan grupos lógicos de operaciones relacionadas con algún concepto del negocio. 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. Es la realización técnica que cumple con el contrato definido, en forma de componentes, programas, datos de configuración, bases de datos. 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. Las funcionalidades del servicio se exponen en la interfaz del servicio a clientes conectados en la red (ver figura 1.1). Existen distintos tipos de servicios con determinadas características que tienen distinto significado en SOA, por ejemplo los procesos del negocio se realizan en servicios orientados a procesos que se componen de secuencias definidas de invocaciones a servicios [17].



1.1 Representación de un servicio y sus componentes.

Un servicio funciona como una aplicación independiente, teniendo sus propias reglas de negocio, datos, procedimientos de administración y operación. Expone toda su funcionalidad utilizando una interfaz basada en mensajes. La comunicación hacia y desde el servicio, es realizada utilizando mensajes y no llamadas a métodos. Estos mensajes deben contener o referenciar toda la información necesaria para ser entendidos.

1.4.3.2.1. Clasificación de los servicios.

Los servicios en SOA presentan varias clasificaciones, entre las cuales se encuentran los servicios básicos, intermediarios, de procesos y públicos.[18]

♦ **Servicios básicos:** Los servicios básicos son los fundamentos de SOA. Son proveedores puros y no mantienen un estado conversacional. Se dividen en servicios centrados en los datos y centrados en la lógica, donde los primeros tienen como propósito manejar los datos persistentes, almacenamiento, recuperación y gestión transaccional, y los segundos proveen encapsulamiento para cálculos complejos o reglas de negocio, tradicionalmente encapsulados en bibliotecas o marcos de trabajo.

♦ **Servicios intermediarios:** 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.

◆ **Servicios de proceso:** 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 desde el punto de vista técnico son la clase de servicios más sofisticada. Una de las principales ventajas que presentan estos servicios es que separan la lógica de los procesos, definiendo los procesos de negocio y su control, en base a orquestación de servicios existentes. Pueden residir en herramientas BPM.

◆ **Servicios públicos:** 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.

1.4.3.3. UDDI

UDDI (UDDI, del inglés Universal Description, Discovery and Integration) son las siglas del catálogo de negocios de Internet. UDDI es un registro público diseñado para almacenar de forma estructurada información sobre empresas y los servicios que estas ofrecen. A través de UDDI, se puede publicar y descubrir información de una empresa y de sus servicios [19].

Un 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 y temporal del proyecto que los creó. 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, tarifas de uso, restricciones técnicas, aspectos de seguridad y niveles de disponibilidad del servicio. Debe notarse que el enfoque está dado en repositorios dentro de los límites de una organización, repositorios que son usados para integración de servicios cruzando fronteras entre organizaciones, típicamente tienen otros requerimientos - en particular, los que se hacen públicos en Internet [20].

Lo más importante de una UDDI (ver figura 1.2) es que contiene información sobre las interfaces técnicas de los servicios de una empresa. A través de un conjunto de llamadas basadas en SOAP, se puede interactuar con el UDDI tanto en tiempo de diseño como de ejecución para descubrir datos técnicos de los servicios que permitan invocarlos y utilizarlos. De este modo, UDDI sirve como infraestructura para una colección de software basado en servicios Web. Los dos escenarios más comunes para UDDI dentro de una organización son la reutilización de código y la configuración dinámica de aplicaciones.

♦ **Reutilización de código:** En tiempo de diseño, los desarrolladores pueden buscar en una UDDI los servicios Web o recursos de programación para reutilizarlos en aplicaciones nuevas.

♦ **Configuración dinámica de aplicaciones:** En tiempo de ejecución, una aplicación consulta el UDDI para descubrir la información actualizada de enlace al servicio que necesita, para conectarse después directamente a ese servicio.



1.2 Representación de la interacción de un repositorio de servicios con los consumidores y servidores.

▪ **Ventajas que brinda la utilización de UDDI**

Posibilita compartir, buscar y reutilizar servicios Web propios o de otros desarrolladores, así como recursos programables.

- Contribuye al aumento de la productividad en menor tiempo, ya que permite utilizar recursos ya elaborados.
- Disminuye el costo de recursos y desarrollo de aplicaciones.
- Contribuye al aumento de la confiabilidad de las aplicaciones, así como mejora el proceso de administración de estas.

1.4.3.4. ESB

Cuando se habla de SOA, se hace necesario referirse al concepto del Bus de Servicios Empresariales (conocido por las siglas ESB del inglés Enterprise Service Bus), esta tecnología permite integrar sistemas a nivel de servicios, cuando se discute sobre tecnología para implementar SOA en una empresa la primera herramienta que se debe incorporar es el “ESB”.

Según Krafzig [20], 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, auditoría, entre otros.

En pocas palabras se puede decir que un ESB es el elemento que conecta los servicios con sus consumidores(ver figura 1.3) y proporciona:[21, 22]

- ◆ **Conectividad:** El propósito principal de un bus de servicios es interconectar a los participantes de una arquitectura SOA.

- ◆ **Soporte a la heterogeneidad de tecnologías:** Debe ser capaz de conectar a participantes basados en distintos lenguajes de programación, sistemas operativos, entornos de ejecución y protocolos de comunicación.

- ◆ **Soporte a la heterogeneidad de paradigmas de comunicación:** Debe ser capaz de mantener distintos modos de comunicación (por ejemplo comunicaciones síncronas y asíncronas).



1.3 Representación de la integración mediante un ESB.

Un ESB se puede ver entonces como una solución de integración distribuida, basada en los mensajes y en estándares abiertos. A medida que los departamentos de IT se centran cada vez más en el diseño de SOA para reducir los costes de desarrollo y para aumentar la agilidad del negocio, los ESB se están convirtiendo en un primer paso clave para el establecimiento de una SOA empresarial. Constituyen los cimientos de una SOA y pueden complementarse con capacidades de productividad adicionales, como la orquestación de servicios y los registros.

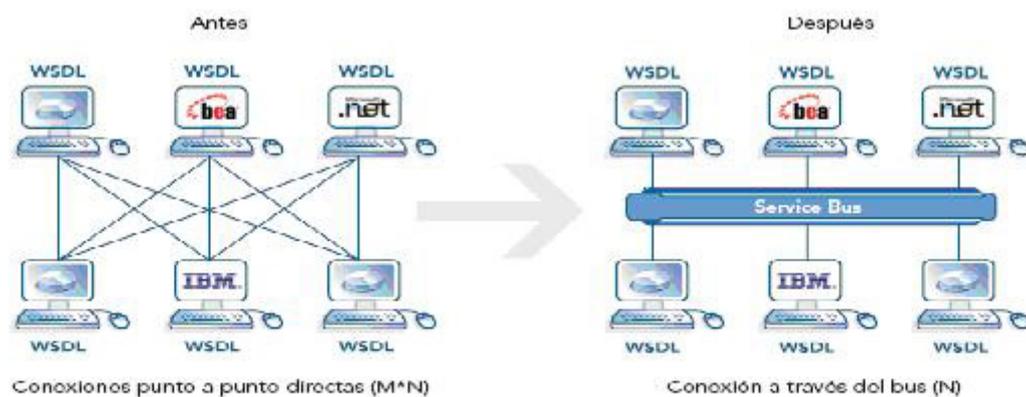
- **Requisitos a escala empresarial de un ESB**

Los requisitos a escala empresarial de un ESB son funciones esenciales de conocer si se desea llegar a obtener una satisfactoria integración de una SOA, pues estos elementos resuelven los problemas mayoritarios a los que se enfrentan los clientes y los proveedores de servicios en un entorno de SOA. Algunos de los más importantes requisitos a escala empresarial de un ESB son:[22]

- **Mensajería distribuida:** El núcleo del ESB lo constituye una aplicación de middleware orientada al mensaje. Este núcleo proporciona un método de transporte fiable y distribuido que emplea un mecanismo de almacenamiento y reenvío gracias al cual se garantiza la entrega de los mensajes incluso en caso de anomalías en la red.

- **Transparencia de las ubicaciones:** Con la mediación entre servicios, un servicio cliente que invoque al proveedor de servicio solo necesita saber que el servicio existe; el cliente no necesita saber dónde se está ejecutando el servicio. El ESB localiza el servicio cuando se invoca.

- **Transparencia de transporte:** En los enfoques tradicionales de la integración punto a punto todos los componentes y objetos están muy estrechamente acoplados (ver figura 1.4). En SOA, los servicios están repartidos por todo el entorno de IT y su acoplamiento es menos estricto, gracias a la transparencia de las ubicaciones. Además de apoyarse en la transparencia de las ubicaciones para conectar clientes y proveedores de servicios, el ESB también proporciona protocolo de transporte físico para hacer posible la comunicación entre servicios utilizando transportes diferentes.



1.4 Representación de conexiones punto a punto y conexiones mediante un ESB.

- **Soporte multiprotocolo:** ESB necesita ser capaz de soportar muchos tipos de sistemas de transporte para integrar sistemas dispares y gestionar el transporte de comunicaciones complejas eficazmente.

- **Calidad de Servicio:** En las aplicaciones empresariales, la calidad de servicio hace referencia, fundamentalmente, a su fiabilidad. La entrega de los mensajes y la fiabilidad de los servicios de invocación son funciones de misión crítica de cualquier sistema. Un ESB, por otro

lado, puede proporcionar un servicio de alta fiabilidad garantizando la entrega del mensaje de principio a fin que va más allá de la fiabilidad que puede ofrecer transportes, de igual manera, los métodos empleados para alcanzar un alto nivel de calidad de servicio deben satisfacer los estándares existentes.

- **Transformación:** Si bien la tarea de un ESB es dirigir mensajes de un servicio al siguiente, hay ocasiones en que el formato de los datos de un servicio no satisface los requisitos del siguiente servicio. Por ese motivo, el ESB debe ser capaz de transformar los datos de un formato a otro.

1.4.3.5. BPM y SOA.

La administración de procesos de negocio (BPM, del inglés Business Process Management) es el entendimiento, gestión e innovación de procesos bajo estándares internacionales, alineados con la estrategia de negocio para asegurar la efectividad del proceso y crear valor a la cadena productiva de la empresa y su sector. Constituye un nuevo paradigma para abordar procesos de mejoramiento que aumenta la eficiencia y facilita integración entre diferentes compañías. Se lleva a la práctica integrando la estrategia, los procesos y la tecnología. Constituye una alternativa para la búsqueda de ventajas competitivas, para afrontar escenarios de competencia en mercados nacionales e internacionales. Se perfila como una nueva línea de pensamiento que atiende necesidades tangibles de las empresas y abre nuevos nichos de mercado para nuevas empresas que se enfoquen en temas de gestión empresarial.

El concepto de BPM está también muy ligado a SOA. BPM es una disciplina de gestión que combina una visión centrada en procesos y de integración de funcionalidades que pretende mejorar la efectividad de las organizaciones. Una solución BPM dispone de los medios necesarios para la realización efectiva de estos procesos así como las funcionalidades necesarias para que los gestores de las empresas puedan controlar y modificar los flujos de trabajo tanto manuales como automáticos.[21]

Aunque BPM puede considerarse como una entidad al margen de las iniciativas SOA, la capacidad para definir nuevos procesos de negocio de forma flexible y rápida es mucho mayor si los recursos de los sistemas de IT se exponen en la forma de orientación a servicios.

1.4.4. Niveles de Madurez de una SOA.

Existe un tema muy interesante y que cada día gana más popularidad en el mundo y es el tema de un modelo de madurez para SOA. El modelo de madurez para SOA aparte de ser un término de los que muchos platican, es una aproximación arquitectónica para resolver los problemas de reusabilidad, flexibilidad y visibilidad de los sistemas de software que soportan las operaciones de las organizaciones o compañías. El modelo de madurez para SOA tiene una relación directa con cinco niveles del CMMI (Integración del Modelo de Madurez de las Capacidades). CMMI constituye un marco de referencia de la capacidad de las organizaciones de desarrollo del software en el desempeño de sus diferentes procesos, proporcionando una base para la evaluación de madurez de las mismas y una guía para implementar una estrategia de mejora continua de dichos procesos.

La versión escalonada del CMMI es una aproximación que usa un conjunto predefinido de áreas de procesos para definir un camino para la mejora de una organización, esto hace que tenga una relación directa con el modelo de madurez para SOA, que a su vez consta de cinco niveles de madurez pero aterrizado en cuestiones básicas de la arquitectura. [23]

Los cinco niveles que propone el CMMI en la versión escalonada (ver figura 1.5) son:

◆ Nivel 1: Inicial

Procesos impredecibles, pobremente controlados y reactivos. Pocas actividades están definidas y el éxito de los proyectos depende del esfuerzo individual. Hay carencia de procedimientos formales, estimaciones de costos, planes del proyecto y mecanismos de administración para asegurar que los procedimientos se siguen.

◆ Nivel 2: Gestionado

Procesos caracterizados en proyectos y acciones reactivas con frecuencia. Son establecidas las actividades básicas para la administración de proyectos de software para el seguimiento

de costos, programación y funcionalidad. El éxito está en repetir prácticas que hicieron posible el éxito de proyectos anteriores, por lo tanto hay fortalezas cuando se desarrollan procesos similares, y gran riesgo cuando se enfrentan nuevos desafíos.

◆ **Nivel 3: Definido**

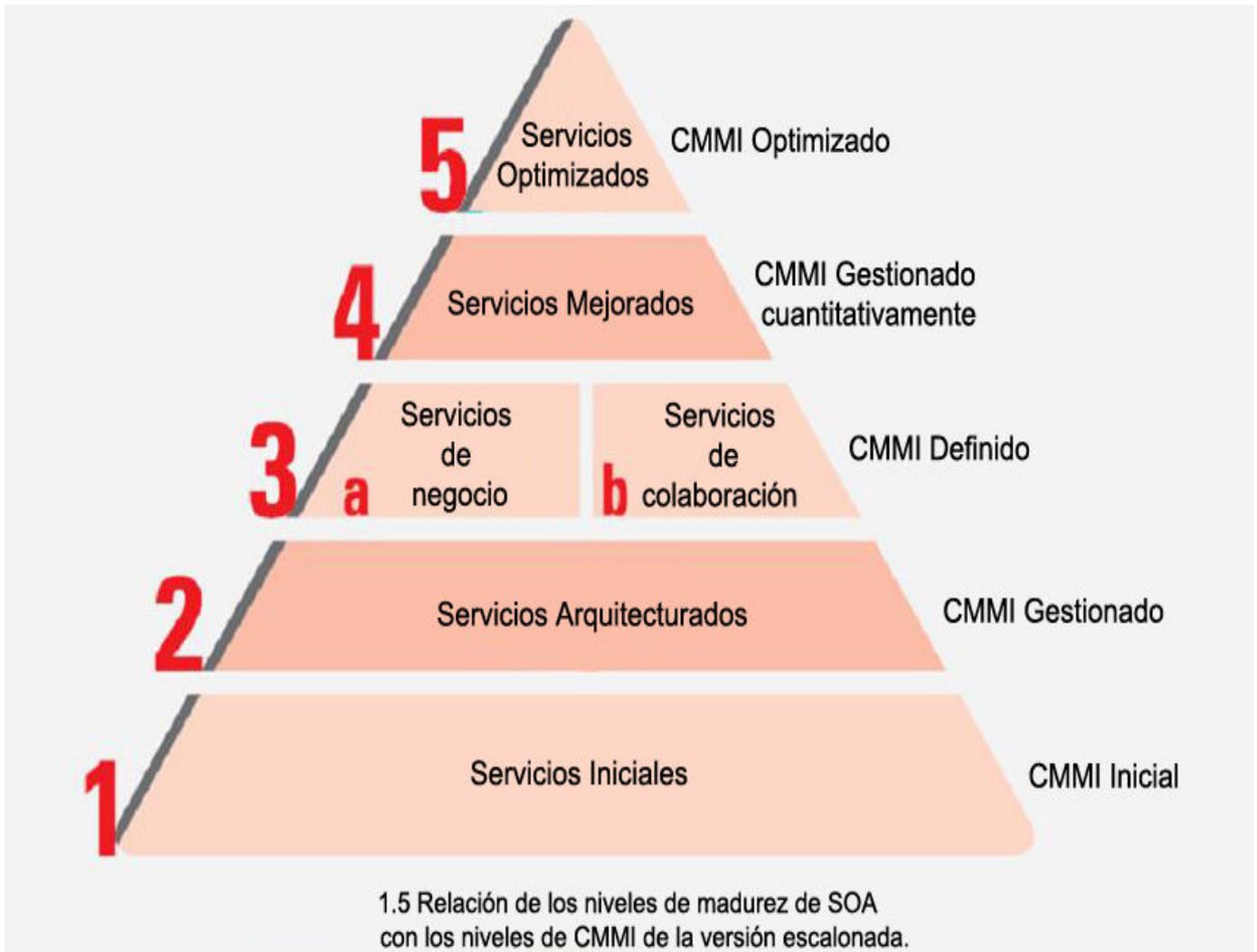
Procesos caracterizados en la Organización y con acciones proactivas. Las actividades del proceso de software para la administración e ingeniería están documentadas, estandarizadas e integradas en un proceso de software estándar para la Organización.

◆ **Nivel 4: Gestionado cuantitativamente**

Los procesos son medidos y controlados. Se registran medidas detalladas de las actividades del Proceso y calidad del Producto. El proceso de software y el producto son entendidos cuantitativamente y controlados.

◆ **Nivel 5: Optimizado**

Enfoque continuo en la mejora de procesos. Existe una mejora continua de las actividades, las que se logran a través de la regeneración con las áreas de procesos y también a partir de ideas innovadoras y tecnología. La recolección de datos es automatizada y usada para identificar elementos más débiles del proceso. Se hace un análisis riguroso de causas y prevención de defectos.



El modelo de madurez para SOA también abarca cinco niveles (ver figura 1.5) los cuales son:

Nivel 1: Servicios SOA iniciales.

Se indican fases de exploración y adopción de SOA, se logra que en los proyectos de desarrollo se integre SOA como parte de la arquitectura, en este nivel existen servicios creados para necesidades inmediatas, no para servicios de negocio.

- **Propósito:** Introducir funcionalidad y nueva tecnología.
- **Alcance:** Proyecto piloto, investigación y experimentación, proyectos pequeños.
- **Tecnología:** Estándares, integración de entidades heredadas.
- **Organizacional:** Desarrolladores aprenden a construir servicios.

◆ Nivel 2: Servicios SOA arquitecturados.

En este nivel pasa a tener un rol fundamental la mediación de los servicios, entrando en acción las características de un ESB y contándose con los registros de servicios como lo es UDDI. En este nivel tiene lugar la integración a bases de datos.

- **Propósito:** Reducción de costos y control de IT.
- **Alcance:** Múltiples aplicaciones integradas dentro de la organización.
- **Tecnología:** Mensajería confiable, mediación heterogeneidad y seguridad.
- **Organizacional:** Grupo de arquitectos provee el liderazgo.

◆ Nivel 3(a): Servicios de negocio SOA / Nivel 3(b): Servicios de colaboración SOA.

En este nivel el negocio comienza a tomar una alta importancia para el usuario. Para que se llegue a estar en este nivel, ya se debe contar con el desarrollo de servicios de negocio, existiendo la colaboración de los servicios para lograr procesos que den mayor sentido a la organización. Entra a tomar sentido el concepto de BPM.

- **Propósito (a):** Agilidad del negocio.
- **Alcance(a):** Procesos de negocio a través de empresas o unidades de negocio distintas.
- **Tecnología(a):** Rehúso, facilidad de modificación y alta disponibilidad.
- **Organizacional(a):** Asociación de IT con el negocio, el ciclo de vida de SOA toma el control.

- **Propósito (b):** Colaboración con asociados del negocio y mercado en general.
- **Alcance (b):** Abarca múltiples organizaciones internas o externas.
- **Tecnología (b):** Servicios externos, seguridad a través de la empresa, traducción.
- **Organizacional (b):** Asociaciones entre organizaciones.

◆ Nivel 4: Servicios SOA de métricas de negocio.

El objetivo fundamental de este nivel es lograr el monitoreo de los procesos. En este nivel se pueden tener métricas de los procesos de negocio basados en los servicios:

- **Propósito:** Encontrar medidas de desempeño del negocio.
- **Alcance:** Unidades de negocio y/o a través de la empresa.

- **Tecnología:** Tableros ejecutivos, motores de reglas de negocio.
- **Organizacional:** Evaluación y respuesta en procesos de negocio en ejecución.

◆ **Nivel 5: Servicios de negocios de SOA optimizados.**

Cuando se logra tener métricas, viene entonces la optimización de los procesos de negocio, donde comienza a adquirir importancia la política del mejoramiento continuo.

- **Propósito:** Optimización del negocio, reaccionar y responder automáticamente.
- **Alcance:** Unidades de negocio y/o a través de la empresa.
- **Tecnología:** Automatización basada en eventos para la automatización.
- **Organizacional:** Cultura de mejora continua en las organizaciones.

1.4.5. Beneficios de SOA.

Los beneficios de SOA para una organización se plasman a dos niveles distintos: al del usuario corporativo y a nivel de la organización de IT.

Desde el punto de vista de la empresa, SOA permite el desarrollo de una nueva generación de aplicaciones dinámicas que resuelven una gran cantidad de problemas de alto nivel, fundamentales para el crecimiento y la competitividad. Las soluciones SOA permiten entre otras cosas:[21]

◆ **Mejorar la toma de decisiones.** Al integrar el acceso a los servicios e información de negocio dentro de un conjunto de aplicaciones dinámicas compuestas, los directivos disponen de más información y de mejor calidad. Las personas, procesos y sistemas que abarcan múltiples departamentos pueden introducirse de forma más directa en una panorámica unificada, lo que permite conocer mejor los balances de costes y beneficios que se producen en las operaciones de negocio que se realizan a diario. Y al disponer de mejor información en un tiempo menor, las organizaciones pueden reaccionar de manera más ágil y rápida cuando surgen problemas o cambios.

◆ **Posibilidad de reaccionar rápidamente a los cambios que se produzcan en el mercado.** Dado que SOA conecta a proveedores y clientes de forma directa y transparente, agilizando toda transacción y optimizando los resultados finales, también se acelera el servicio

al cliente, anticipándose la empresa a sus necesidades y reaccionando ante los cambios, siempre imprevisibles, que surjan en la demanda.

- ◆ **Definir los procesos y modelos de negocio entre compañías.** Gracias a la reutilización de procesos de negocio, las compañías cuentan con un sistema que fomenta la actitud colaborativa y disminuye al máximo el coste total de propiedad. Las empresas pueden contar con recursos externos, sin incrementar sus inversiones, ni adquirir tecnología complementaria a la ya existente.

- ◆ **Mejorar la productividad de los empleados.** Un acceso óptimo a los sistemas y la información y la posibilidad de mejorar los procesos permiten a las empresas aumentar la productividad individual de los empleados. Estos pueden dedicar sus energías a los procesos importantes, los que generan valor añadido y a actividades de colaboración, semiestructuradas, en vez de aceptar las limitaciones y restricciones impuestas por los sistemas de IT rígidos y monolíticos.

- ◆ **Potenciar las relaciones con clientes y proveedores.** Las ventajas de SOA trascienden las fronteras de la organización. Los beneficios que ofrece SOA trascienden los límites de la propia organización. Los procesos de fusión y compra de empresas se hacen más rentables al ser más sencilla la integración de sistemas y aplicaciones diferentes. La integración con socios comerciales y la optimización de los procesos de la cadena de suministro son, bajo esta perspectiva, objetivos perfectamente asequibles. Desde el punto de vista de los departamentos de IT, la orientación a servicios supone un marco conceptual mediante el cual se puede simplificar la creación y mantenimiento de sistemas y aplicaciones integradas, y una fórmula para alinear los recursos de IT con el modelo de negocio y las necesidades y dinámicas de cambio que le afectan.

- ◆ **Aplicaciones más productivas y flexibles.** La estrategia de orientación a servicios permite a IT conseguir una mayor productividad de los recursos de IT existentes –como pueden ser las aplicaciones y sistemas ya instalados e incluso los más antiguos- y obtener mayor valor de ellos de cara a la organización sin necesidad de aplicar soluciones de

integración desarrolladas ex profeso para este fin. La orientación a servicios permite además el desarrollo de una nueva generación de aplicaciones compuestas que ofrecen capacidades avanzadas y multifuncionales para la organización con independencia de las plataformas y lenguajes de programación que soportan los procesos de base.

- ◆ **Desarrollo de aplicaciones más rápido y económico.** El diseño de servicios basado en estándares facilita la creación de un repositorio de servicios reutilizables que se pueden combinar en servicios de mayor nivel y aplicaciones compuestas en respuesta a nuevas necesidades de la empresa. Con ello se reduce el coste del desarrollo de soluciones y de los ciclos de prueba, se eliminan redundancias y se consigue su puesta en valor en menos tiempo.

- ◆ **Aplicaciones más seguras y manejables.** Las soluciones orientadas a servicios proporcionan una infraestructura común para desarrollar servicios seguros, predecibles y gestionables. Conforme van evolucionando las necesidades de negocio, SOA facilita la posibilidad de añadir nuevos servicios y funcionalidades para gestionar los procesos de negocio críticos. Se accede a los servicios y no a las aplicaciones, y gracias a ello la arquitectura orientada a servicios optimiza las inversiones realizadas en IT potenciando la capacidad de introducir nuevas capacidades y mejoras. Y además, puesto que se utilizan mecanismos de autenticación y autorización robustos en todos los servicios –y puesto que los servicios existen de forma independiente unos de otros y no se interfieren entre ellos- la estrategia de SOA permite dotarse de un nivel de seguridad superior.

1.4.6. SOA y Servicios Web.

La arquitectura SOA provee un marco de diseño para que aplicaciones independientes puedan integrarse a través de la red y de esta forma se pueda acceder a sus funcionalidades. Lo cierto es que un diseño basada en SOA no necesariamente exige la implantación de servicios Web. Sin embargo, la forma más habitual de implementar SOA es a través de los servicios Web, pues de esta forma se pueden descomponer estas aplicaciones en servicios e implementar la funcionalidad de forma modular.

Los servicios Web son aplicaciones que utilizan estándares para el transporte, codificación y protocolo de intercambio de información, permiten la comunicación entre sistemas de cualquier plataforma y se utilizan en gran variedad de escenarios de integración tanto dentro de las organizaciones como con socios de negocio.[21]

Otra definición relativa a servicio Web es la que lo describe como una entidad computacional accesible a través de Internet (mediante la utilización de estándares y protocolos específicos) que permite solicitar un determinado servicio. [24]

Los servicios Web se basan en un conjunto de estándares de comunicación entre los que figuran:

- ◆ **WSDL** (Web Service Description Language): Descripción en (XML) de un servicio a nivel de comunicación, operaciones, mensajes (API) y protocolos utilizados.

- ◆ **SOAP** (Simple Object Access Protocol): Protocolo basado en intercambio de mensajes XML. La especificación permite realizar el intercambio a través de múltiples protocolos de transporte como por ejemplo, HTTP o SNMP.

Existen más especificaciones, a las que se denomina genéricamente como la Arquitectura Web Services, en la que se definen distintas funcionalidades para el descubrimiento de servicios Web, gestión de eventos, seguridad y fiabilidad en el intercambio de mensajes y transacciones.

La utilización de Web Services en una SOA provee muchas ventajas, entre las que figuran la interoperabilidad, la reusabilidad y la formalización.

- **Interoperabilidad:** Porque las descripciones basadas en lenguajes estándar (WSDL) permiten que la utilización de servicios Web sea independientemente de plataformas, protocolos y lenguajes de programación.

- **Reusabilidad:** Porque al tratarse de componentes de software con una interfaz estándar (interoperable) y públicamente accesible, la capacidad de reutilización de su funcionalidad es máxima.

- **Formalización:** Porque las descripciones de los servicios Web son formales, es decir, interpretables, no solo por humanos, sino también por entidades software.

1.5. Metodologías de desarrollo.

Todo desarrollo de software es riesgoso y difícil de controlar, pero si no llevamos una metodología de por medio, lo que obtenemos es clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos.

1.5.1. ¿Qué es una metodología?

La palabra metodología proviene del griego meta, “más allá”, odos que significa “camino” y logos, que significa “estudio”, de lo cual se puede inferir sin ninguna dificultad que la metodología se refiere a los métodos de investigación que se siguen para alcanzar una gama de objetivos en una ciencia. Pero cuando se habla sobre metodología no es tan sencillo como parece, puesto que se presenta como una ciencia amplia, compleja y transdisciplinaria, con un objeto bien definido, los métodos, así como normas y principios y una estructura bien definida. [25].

Se encuentran igualmente otras definiciones como la que nos brinda Kaplan cuando afirma que: "La palabra metodología es utilizada para referirse a una disciplina y a su contenido. Por metodología yo quiero decir el estudio --descripción, explicación y justificación-- de métodos, y no los métodos en sí mismos". [26]

Según la Real Academia de la lengua española se puede concluir que metodología se refiere al conjunto de métodos que se siguen en una investigación científica o en una exposición doctrinal. [27]

En otras palabras la metodología se entenderá como la parte del proceso de investigación (Método Científico), que sigue la instrucción y permite sistematizar los métodos y las técnicas

necesarias para llevarla a cabo, la cual dependerá de los postulados que el investigador considere válidos, pues será a través de la acción metodológica como recolecte, ordene y analice la realidad.

El proceso de desarrollo de un software no escapa del uso de metodologías para lograr un buen resultado final, porque todo desarrollo de software es riesgoso y difícil de controlar, pero si no llevamos una metodología de por medio, lo que se obtiene son clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos.

No existe una metodología de software universal. Cada equipo de desarrollo escoge la metodología según las características de su proyecto, por lo que es importante determinar el alcance del proyecto antes de escoger la metodología que se va a usar en el desarrollo del mismo. A continuación se mencionan algunas metodologías existentes, abordando en cada una de ellas sus aspectos principales.

1.5.2. Rational Unified Process (RUP).

La metodología RUP, llamada así por sus siglas en inglés Rational Unified Process, divide en 4 fases el desarrollo del software:[28, 29]

- **Inicio:** El objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración:** En esta fase el objetivo es definir la arquitectura del sistema.
- **Construcción:** En esta etapa el objetivo es llegar a obtener un producto listo para su utilización.
- **Transición:** El objetivo es llegar a obtener el despliegue del proyecto.

Flujos de Trabajo.

- ◆ **Modelamiento del negocio:** Describe los procesos de negocio, identificando quiénes participan y las actividades que requieren automatización.

- ◆ **Requerimientos:** Define qué es lo que el sistema debe hacer, para lo cual se identifican las funcionalidades requeridas y las restricciones que se imponen.

- ◆ **Análisis y diseño:** Describe cómo el sistema será realizado a partir de la funcionalidad prevista y las restricciones impuestas (requerimientos), por lo que indica con precisión lo que se debe programar.

- ◆ **Implementación:** Define cómo se organizan las clases y objetos en componentes, cuáles nodos se utilizarán, la ubicación en ellos de los componentes y la estructura de capas de la aplicación.

- ◆ **Prueba (Testeo):** Busca los defectos a lo largo del ciclo de vida.

- ◆ **Instalación:** Produce release del producto y realiza actividades (empaquete, instalación, asistencia a usuarios, etc.) para entregar el software a los usuarios finales.

- ◆ **Administración del proyecto:** Involucra actividades con las que se busca producir un producto que satisfaga las necesidades de los clientes.

- ◆ **Administración de configuración y cambios:** Describe cómo controlar los elementos producidos por todos los integrantes del equipo de proyecto en cuanto a: utilización/actualización concurrente de elementos, control de versiones, etc.

- ◆ **Ambiente:** Contiene actividades que describen los procesos y herramientas que soportarán el equipo de trabajo del proyecto; así como el procedimiento para implementar el proceso en una organización.

El ciclo de vida de RUP tiene tres características fundamentales:

Guiado por casos de uso.

Los casos de uso reflejan las necesidades de los futuros usuarios, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. Los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso.

Centrado en la arquitectura.

La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comenzando por los casos de uso relevantes desde el punto de vista de la arquitectura (casos de uso arquitectónicamente significativo). El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.

Iterativo e incremental.

RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Es práctico dividir el trabajo en partes más pequeñas o miniproyectos. Cada miniproyecto es una iteración que resulta en un incremento. Cada iteración se realiza de forma planificada.

1.5.3. Extreme Programming (XP)

Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo y corto equipo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.[28-30]

La metodología se basa en:

- ◆ **Pruebas Unitarias:** Se basa en las pruebas realizadas a los principales procesos, de tal manera que podamos hacer pruebas de las fallas que pudieran ocurrir.
- ◆ **Refabricación:** Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- ◆ **Programación en pares:** Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un

proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. [31]

Lo fundamental en este tipo de metodología es:

- La comunicación entre los usuarios y los desarrolladores.
- La simplicidad al desarrollar y codificar los módulos del sistema.
- La retroalimentación concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

1.5.4. Microsoft Solution Framework (MSF)

Es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.[28]

MSF tiene las siguientes características:

- ◆ **Adaptable:** Es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.

- ◆ **Escalable:** Puede organizar equipos tan pequeños entre tres o cuatro personas, así como también, proyectos que requieren cincuenta personas o más.

- ◆ **Flexible:** Es utilizada en el ambiente de desarrollo de cualquier cliente.

- ◆ **Tecnología Agnóstica:** Porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

MSF se compone de varios modelos encargados de planificar las diferentes partes implicadas en el desarrollo de un proyecto: Modelo de Arquitectura del Proyecto, Modelo de Equipo,

Modelo de Proceso, Modelo de Gestión del Riesgo, Modelo de Diseño de Proceso y finalmente el modelo de Aplicación.

1.6. Conclusiones Parciales.

En este capítulo se presentó toda la información bibliográfica o el fundamento teórico referente al tema de SOA, se analizaron las condiciones específicas que rodean el problema que se plantea, se profundizó en algunos conceptos necesarios para la comprensión de este trabajo.

CAPÍTULO 2: SOLUCIÓN PROPUESTA

2.1. Introducción.

Hasta el momento se ha hablado de SOA y los elementos y conceptos que involucra pero no se han mencionado las metodologías y distintos enfoques que existen mundialmente para lograr su adopción exitosa; por tales motivos en el presente capítulo se pueden encontrar las metodologías y buenas prácticas para SOA que proporcionan varias de las empresas líderes en el mercado del software a nivel mundial, así como el estudio de la situación actual que presenta la UCI respecto a SOA, lo cual se tuvo en cuenta para la elaboración de la guía que constituye el objetivo principal de nuestro trabajo. Como conclusión al proceso de estudio y diagnóstico, el capítulo finaliza con la propuesta de una guía

2.2. Buenas prácticas desarrolladas por las empresas líderes para SOA.

El concepto de SOA domina ya el panorama actual de las IT, cada vez más empresas toman conciencia de que SOA representa una ventaja competitiva importante y son muchos ya los que ven SOA como la arquitectura del futuro.

Hoy en día existen muchas empresas que brindan servicios SOA, otras planean adoptar esta arquitectura y otras muchas han fracasado en el intento de realizar proyectos SOA exitosos, pero lo que no da lugar a dudas es que SOA abre caminos para crear valores en los sistemas de información.

En la actualidad existen varias empresas promotoras de SOA a nivel internacional de las cuales se ofrecen de forma resumida las características de sus propuestas de trabajo, ellas son: IBM, Microsoft, Software AG y Software Associates.

2.2.1. Software Associates.

Software Associates Corporation es una empresa líder en consultoría e implantación de soluciones SOA y BPM. [32]

El ciclo de vida SOA que propone Software Associates se define en base a las políticas generales de la empresa donde se tienen en cuenta las distintas fases por las que pasa un proyecto y las versiones por las que deben pasar los distintos componentes SOA, las validaciones exigidas, las autorizaciones necesarias, sus correspondientes aceptaciones, su catalogación a nivel superior, entre otras más. [33]

Lo primero que Software Associates hace cuando comienza a realizar un proyecto SOA es definir su alcance y elegir soluciones que brinden resultados a corto plazo, de esta forma logra que existan menos probabilidades de complicarse tecnológicamente. Consideran también que cuando se trata de implantar una SOA es necesario que se vinculen la mayor cantidad de expertos posibles en el tema, lo cual es un elemento de suma importancia para que se marche por buen camino, paralelamente a esto comienza a emplear un plan estratégico para lograr una óptima orientación a los servicios. Esta estrategia y planificación de SOA abarca a grandes rasgos cuatro etapas relevantes las cuales ayudan a que se consigan los objetivos empresariales que dieron lugar a pensar en su adopción.



2.1 Propuesta SOA de Software Associates

Adaptación Metodológica

Para comenzar a desarrollar esta etapa Software Associates realiza primeramente un estudio profundo de la naturaleza del proyecto, esto consiste en comenzar a identificar los servicios dentro del área en la que se comenzará a trabajar, que a su vez se convertirá en el área piloto

para iniciar el proyecto SOA. Posteriormente selecciona un marco metodológico de forma tal que puedan adaptar la metodología seleccionada a las necesidades específicas de su negocio.

Definición de la arquitectura SOA

Análogamente a la etapa anterior y luego de haber establecido los distintos objetivos empresariales que se persiguen, se procede a ensamblar los elementos mínimos indispensables de una SOA con el fin de garantizar el éxito; esto implica conectar los primeros servicios y seleccionar o desarrollar un registro o un repositorio. En esta fase ya se logra tomar conciencia del crecimiento del proyecto y se logra comenzar a visualizar el futuro que representa SOA para una empresa.

Definición del proyecto SOA

Para llegar a esta fase se hace necesario que el proyecto SOA haya logrado un grado profundo de madurez dentro de la organización, ya se deben haber alcanzados los objetivos mínimos y SOA dentro de la empresa adquiere un valor fundamental. Es en esta etapa que se comienzan a tener en cuenta los planes de seguridad y se construye completamente la infraestructura de mensajería, donde entra a ocupar un rol fundamental el ESB y el estudio de las herramientas que ofrece el mercado así como el desarrollo del gobierno de los servicios. Incluye también la revisión de los estándares y la atención especial a la evolución del mercado SOA.

Plan de implantación.

En esta etapa se confecciona un plan para la implantación total de SOA en la organización, puesto que cuando se llega a esta fase se cuenta con un proyecto seguro, confiable y flexible, el cual puede comenzar a soportar la infraestructura tecnológica de la empresa. Este plan se realiza por iteraciones y se realizan pruebas para la mejora continua de SOA en la organización.

2.2.2. Microsoft.

Microsoft es una empresa muy conocida mundialmente y fundada en el año 1975. Esta empresa, que se ha mantenido en la cabecera en cuanto a las IT y sin dudas presta gran importancia a lo referente a SOA, describe este tipo de arquitectura como una estrategia general de organización de los elementos de las tecnologías de la información, en un compendio de aplicaciones complejas y sistemas distribuidos que se pueden transformar en una red de recursos integrados y flexibles. Por lo que un proyecto SOA bien ejecutado permitiría alinear los recursos tecnológicos con los objetivos del negocio, ganando así un mayor grado de integración con clientes y proveedores y ayudaría a las empresas a optimizar sus procesos internos y sus flujos de información para mejorar la productividad de sus empleados, dando como resultado final un aumento notable de la agilidad de la organización.

Se puede decir que no todos los esfuerzos de orientación a servicios han resultado satisfactorios para Microsoft. Los proyectos SOA tienen un éxito limitado cuando adoptan un modelo ascendente, implantar SOA por el gusto de tener SOA sin tener una referencia clara de los objetivos del negocio en el que debe aplicarse es un proyecto sin principios organizativos. Por otra parte un modelo descendente exige inversiones de tiempo y recursos muy grandes que cuando se obtengan los resultados en un futuro ya estos no reflejarán las necesidades del negocio.

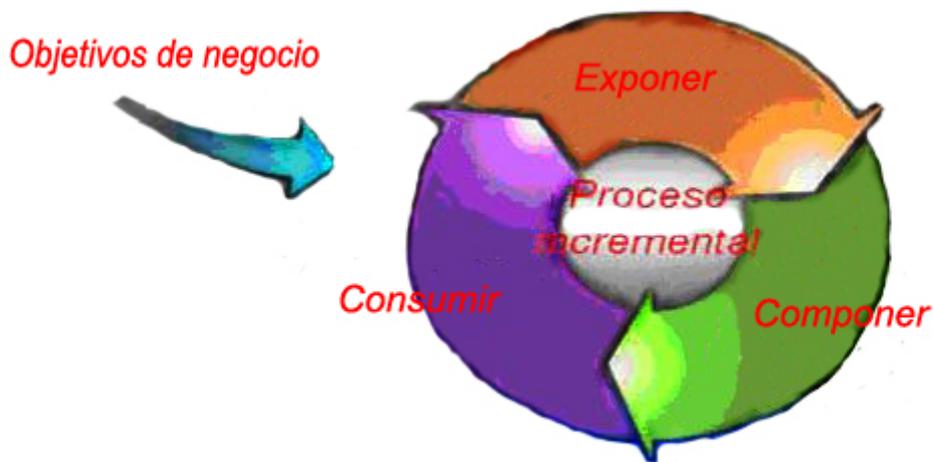
Microsoft por su parte aboga por lo que llama una metodología “middle out”. En esta metodología los esfuerzos de un proyecto SOA estarían dirigidos hacia una visión estratégica y las necesidades del negocio y se completarían mediante proyectos SOA incrementales e iterativos diseñados de forma que cumplan con los objetivos del negocio, dando respuesta a una necesidad concreta a la a vez.

◆ Middle-out.

Middle-out es una estrategia de pasos progresivos, en la cual lo primero que se tiene en cuenta son las condiciones principales de la empresa y la visión estratégica de esta para establecer con claridad las prioridades del proyecto, teniendo como principal ventaja una

rápida puesta en vigor y obtención de resultados en todos sus pasos incrementales e iterativos.

Desde el punto de vista más técnico middle-out se puede resumir en tres pasos: exposición, composición y consumo. Una vez que las principales condiciones del negocio están bien definidas se puede comenzar el proceso de implementación. Partiendo de la visión priorizada y claramente definida, cada proyecto de implementación es un paso progresivo de crear (exposición) nuevos servicios, agregar (composición) estos servicios dentro de procesos mas amplios, y hacer que las salidas estén a disposición de los usuarios (consumo) dentro de la empresa.[21]



2.2 Propuesta SOA de Microsoft

Exposición.

La fase de exposición de esta metodología SOA se centra en generar los servicios necesarios a partir de las aplicaciones y datos disponibles. La forma más común de implementación de tecnologías son los servicios Web. Mientras los desarrolladores deliberan qué exponer, sus decisiones se tomarán a partir de necesidades bien definidas y priorizadas del negocio.

En esta fase se debe de tener en cuenta la forma en que se implementan los servicios, enfocarse principalmente en qué servicios crear a partir de los datos y aplicaciones subyacentes.

Composición.

Una vez que son expuestos los servicios individuales, estos deben ser integrados en un proceso de negocio multifuncional. El objetivo de esta fase es permitir la mayor flexibilidad y agilidad del negocio, admitiendo la adicción, cambio y eliminación de procesos sin entrar en conflictos con las aplicaciones y sistemas de información subyacentes.

Consumo.

Esta fase es el paso final para la construcción de una solución SOA, después de crear una nueva aplicación o proceso de negocio, la funcionalidad resultante se pone a disposición de los usuarios finales o de otros sistemas. Estas aplicaciones basadas en tecnologías Web son las que conducen a la productividad del negocio. Los usuarios podrán consumir los servicios compuestos utilizando distintos medios, como puede ser portales Web, clientes avanzados y dispositivos móviles.

2.2.3. IBM.

International Business Machines o IBM (conocida coloquialmente como el Gigante Azul) es una empresa que fabrica y comercializa herramientas, programas y servicios relacionados con la informática.

Para esta prestigiosa compañía, SOA ocupa un importante rol, según se puede encontrar en declaraciones de sus ejecutivos más importantes, los que alegan que:

“La arquitectura SOA desempeña un papel estratégico en cualquier compañía, porque permite colaborar e innovar, identificar nuevas oportunidades de negocio y poner las necesidades de los clientes en primer plano” [34]

“La administración orientada a servicios, o SOA, es una forma de ver los procesos del negocio como un conjunto de servicios vinculados. Constituye un enfoque que se vale de los estándares abiertos para que las operaciones de la empresa sean más eficientes, eficaces y fluidas. Cuando los procesos de negocio se asientan sobre una base SOA, una empresa puede lograr que sus datos y aplicaciones de software interoperen mejor entre las distintas unidades de la empresa y también con terceros externos. Este enfoque apalanca los recursos

existentes a fin de contribuir a mejorar la productividad, reaccionar rápidamente a las condiciones cambiantes del mercado y aprovechar las oportunidades que se presentan.” [35]

IBM comienza evaluando la combinación de aplicaciones existentes en una empresa u organización a fin de determinar si la transformación es necesaria, incluyendo mantener, renovar, reemplazar o eliminar aplicaciones. Lleva a cabo un completo inventario de sus aplicaciones y recomienda ajustes en la combinación, sobre la base de criterios de negocios estratégicos, financieros, funcionales y técnicos. Esta importante empresa desarrolladora identifica disparadores, tales como adquisiciones, fusiones y consolidaciones de sistemas, que podrían afectar el presupuesto con que cuenta una empresa para desarrollar una SOA y verifica su alineación con los objetivos corporativos.

Luego de esto IBM incluye la evaluación, la visión de plan y el plan de transición, para ayudar a las empresas en cómo pueden desarrollar una SOA; esto incluye:

- **La evaluación de SOA:** Brinda una visión precisa a las empresas interesadas de cómo sus negocios pueden beneficiarse con SOA.
- **Definición de la visión de SOA:** Constituye una descripción sobre cómo SOA puede ayudar a resolver los problemas de negocio específicos de una empresa.
- **Plan de transición de SOA:** Se lleva a cabo para ayudar a que se establezca cómo la organización o empresa implicada realizará la transición a SOA.
- **Definición de un grupo de expertos:** Define un grupo de personas que poseen las aptitudes correctas para constituir un grupo experto interno de SOA.
- **Organización y dirección de SOA:** Se definen las normas relacionadas con la SOA que se comenzará a realizar en la empresa u organización que se encuentra implicada.
- **Plan de IT:** Se incluyen la planificación de la inversión y la priorización de las actividades de las IT.
- ◆ **El ciclo de vida de SOA**

Los desarrolladores de SOA de IBM han manifestado que cuando piensan en este tipo de arquitectura lo hacen en término de un ciclo de vida o desarrollo y luego de realizar los primeros pasos para comenzar a desarrollar una SOA en determinada empresa continúan con

la fase de modelado, obteniendo los requerimientos y diseñando los procesos de negocio. Luego que los procesos sean optimizados, se implementan ensamblándoles nuevos o existentes servicios para formar dichos procesos de negocio; más tarde, estos se despliegan en un ambiente de servicios integrados y altamente seguros y luego que los procesos del negocio son desplegados, se administran y monitorean desde perspectivas del negocio y de las IT [36]. El ciclo de vida se retroalimenta de la información obtenida de esta fase de administración para mantener un continuo mejoramiento en los procesos. Bajo todas estas etapas del ciclo de vida está la fase de gobernación y procesos, que proporciona la guía para todo el proyecto SOA.



2.3 Práctica SOA de IBM.

Modelado

Se comienza la fase de modelado reuniendo y analizando requerimientos del negocio que entonces se utilizarán para modelar, simular y optimizar los procesos del negocio. Los procesos resultantes se utilizan para diseñar servicios de software asociados y niveles de

servicios que soporten estos procesos. Durante esta fase, se usa un modelo para establecer un entendimiento común entre el negocio y las IT de los procesos del negocio, los objetivos y las salidas, así como para asegurarse que la aplicación resultante cumpla con los requerimientos del negocio definidos. Este modelo provee la línea base donde se medirá el rendimiento del negocio.

Ensamblado

Durante la fase de ensamblado, se crean los servicios a partir de bienes existentes, como por ejemplo sistemas financieros, aplicaciones y otras soluciones que corran en el negocio. En muchos casos, una librería de servicios existentes se puede utilizar para encontrar servicios que ya existen en la organización. Si no existe esta funcionalidad, se puede crear y probar un servicio que entregue la funcionalidad requerida para un proceso en particular del negocio. Luego que los servicios requeridos estén disponibles, son orquestados para implementar un proceso del negocio.

Despliegue

En esta fase se configura y escala el ambiente de ejecución donde se lograrán los niveles de servicios requeridos por los procesos del negocio. Después que esté configurado un proceso se puede desplegar a un ambiente de servicios robusto, escalable y altamente seguro. Este ambiente se optimiza para que pueda ejecutar confiadamente misiones críticas de los procesos del negocio a la vez que provee flexibilidad para las actualizaciones dinámicas en respuesta al cambio de los requerimientos. Este acercamiento o solución orientada a servicios reduce el costo y la complejidad asociados al mantenimiento de numerosas integraciones punto a punto.

Administración.

La fase de administración implica establecer y mantener la disponibilidad y tiempo de respuesta de los servicios, así como administrar los bienes de servicios subyacentes. Se pueden monitorear los indicadores claves de rendimiento KPI (KPI, del inglés Key Performance Indicators) en tiempo real para obtener la información necesaria para prevenir, aislar, diagnosticar y arreglar problemas, pues es fundamental comprender el rendimiento en

tiempo real de los procesos del negocio, de manera que se establezca una mejora continua. Esta fase incluye además la administración y mantenimiento del control de versiones sobre los servicios que conforman los procesos del negocio.

Gobierno y procesos.

Esta fase es crítica para el éxito de cualquier proyecto SOA. Para asegurar este éxito, se puede crear un centro de excelencia dentro del negocio para implementar las políticas del gobierno y seguir comprobados estándares internacionales de control de información y tecnologías relacionadas. Implementar fuertes políticas de gobierno puede dar resultado a un exitoso proyecto de SOA, incluso tiene el potencial de obtener mayores ganancias e incrementar los valores de las acciones empresariales.

2.2.4. Software AG.

Software AG es una compañía alemana líder mundial en soluciones de IT y con gran renombre por sus confiables y eficaces soluciones de software para misión crítica. Ofrece además el poder para crear, administrar y gobernar de manera rápida y rentable nuevos procesos de negocio dentro de una SOA, brindando el control que se necesita para alcanzar la visión de negocios necesaria.[37]

Esta empresa presta mucha atención a la investigación y desarrollo de muchos líderes en IT alrededor del mundo, para ello creó el Centro de Competencia en SOA, una división que tiene como fin ayudar a las organizaciones a entender el valor de SOA en sus ambientes exclusivos, crear e instrumentar una eficaz estrategia SOA y medir los beneficios. Básicamente este centro ayuda a:

- Comprender el valor de SOA para un negocio en particular.
- Desarrollar e implementar una estrategia SOA.
- Evaluar los beneficios a partir de los objetivos empresariales preestablecidos.
- Reducir los costes y el tiempo necesario para la implementación de una SOA.

Software AG considera que cuando una organización piensa adoptar un proyecto SOA se necesitan considerar varios aspectos relevantes para el éxito del proyecto, uno de estos

aspectos sería establecer una visión del proyecto SOA, de gran relevancia; dado que se trata de definir y establecer la línea de trabajo desde el comienzo, es decir, puntualiza de forma clara lo que será la SOA y el valor que ofrecerá. Otro aspecto sería establecer una metodología para SOA porque se hace muy necesario desde el comienzo de un proyecto SOA el uso de una metodología clara y consistente.

Descubriendo SOA.

La fase de descubrimiento es una fase que tiene poca duración y posee como principal objetivo proporcionar un análisis rápido de los objetivos que se desean alcanzar de los sistemas que con que cuenta la empresa en la que se aspira a realizar la SOA. Es en esta etapa cuando se determina si es factible el desarrollo de una SOA, es decir, se decide si en verdad esta empresa requiere un proyecto SOA.

Evaluación de la preparación.

En esta fase Software AG comienza a explorar de forma más profunda como SOA puede lograr los retos específicos de la empresa, tanto en el contexto de los objetivos de la organización como en la infraestructura de IT. Ya se identifica la existencia real del problema que lleva a la implementación de una SOA y se determina de forma general el impacto que impondrá la adopción de esta arquitectura en el futuro de la organización, es también donde se inicia una propuesta de solución SOA.

Ciclo de vida de los servicios de SOA.

Esta fase contempla la planificación y diseño del proyecto hasta la implementación y evaluación de acuerdo con factores de éxito identificados, también se desarrollan y establecen directrices de servicios y se diseñan y construyen de forma detallada datos específicos para el establecimiento de este tipo de arquitectura. Es aquí donde se ponen en práctica los servicios en el proyecto y se despliegan soluciones basadas en los datos que se obtuvieron durante las fases anteriores.

Evaluación de madurez.

Software AG plantea que se considera de gran importancia aprovechar al máximo todas las ventajas que ofrece la utilización de la SOA que se viene creando y desarrollando continuamente, por esta razón, en esta etapa se hace una revisión la cual está dirigida a los éxitos y planes de SOA. Esta revisión se debe diseñar para aprovechar todos los logros alcanzados, identificar oportunidades de mejora y para diseñar un plan enfocado a obtener aún mejores resultados en el futuro.

Optimización.

Al final de cada etapa se evalúa el éxito del proyecto con un informe del logro y se identifican huecos y áreas para el refinamiento.

2.3. Metodología RUP para SOA.

Una metodología para el desarrollo orientado a servicios no requiere necesariamente un nuevo proceso de desarrollo, sino que se puede construir sobre el proceso o enfoque de desarrollo que se siga en la Organización, agregando actividades y artefactos específicos para el desarrollo orientado a servicios, y por lo tanto en principio cualquier proceso podría servir de base. Sin embargo, dada la naturaleza compleja y cambiante de los requerimientos del negocio en el contexto de las Organizaciones que podrían incorporar el enfoque SOA, es recomendable que se siga un enfoque iterativo incremental como forma de lidiar con estos cambios.

Procesos como el Rational Unified Process (RUP), que es clasificado como “proceso pesado”, y eXtremme Programming (XP), que es clasificado como “proceso liviano” o ágil, presentan ambos como base un enfoque de desarrollo iterativo incremental, que resulta apropiado para un desarrollo orientado a servicios.

El proceso de desarrollo base es principalmente una adaptación del RUP que mantiene las características más importantes del mismo mencionadas en el capítulo 1: 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.

En la dimensión del tiempo se definen las cuatro fases: Inicial, Elaboración, Construcción y Transición, en las tres primeras fases se definen dos iteraciones y en la última fase una iteración.

En la dimensión de las Disciplinas se definen las disciplinas tradicionales del desarrollo de software: requerimientos, diseño, implementación y verificación, más las disciplinas de soporte: gestión del proyecto, de la calidad, de la configuración e implantación. [18]

2.3.1. Extensión al proceso base para desarrollo 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. Un aspecto principal en una SOA es el modelado de los procesos de la Organización como forma de unir los aspectos del negocio con las soluciones informáticas provistas, teniendo en cuenta que los servicios orientados a procesos constituirán la traducción de los procesos del negocio a los procesos que se modelan en las aplicaciones de la Organización.

Como parte central de la metodología definida se incorporan 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. Luego en la Disciplina de Implementación se agrega la actividad de implementación de servicios según fueron definidos en la etapa de diseño.[18]

2.3.1.1. Disciplina Modelado del Negocio.

El propósito de la disciplina Modelado del Negocio en general es asegurar que clientes, usuarios finales, desarrolladores y otros involucrados posean un conocimiento común de la Organización objetivo, derivar los requerimientos del sistema de software que son necesarios

para apoyar a la Organización y comprender como el sistema de software a implantar se inscribe en la misma.

El esfuerzo de modelado del Negocio puede tener distintos alcances dependiendo del contexto y las necesidades, incluyendo por ejemplo un esfuerzo mayor de reingeniería del Negocio.

A continuación se describen las actividades propuestas en la Disciplina Modelado del Negocio, se presenta la actividad Evaluar la Organización Objetivo, la actividad Identificar los procesos del Negocio, presentando en cada caso el Objetivo de la actividad y su descripción.

◆ **Evaluar la Organización Objetivo.**

Objetivo: Esta actividad tiene como objetivo involucrar al equipo de proyecto con la Organización para la cual se está realizando el desarrollo, interiorizando a los participantes sobre aspectos como: el área, funcionamiento, empleados, etc. de la Organización.

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, actitudes de las personas, clientes, competencia, desafíos tecnológicos, problemas y áreas de mejora, identificando claramente los involucrados en el esfuerzo de modelado del negocio.

◆ **Identificar los procesos del Negocio.**

Objetivo: Esta actividad tiene como 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 Casos de Uso del Negocio con los flujos correspondientes y se modelan con Diagramas de Actividad.

Descripción: Esta actividad está compuesta de varias sub-actividades que en conjunto permiten identificar los Procesos del Negocio, describiendo los actores participantes y el flujo de ejecución de los mismos en la Organización. Para esto se deben encontrar los actores y

Casos de Uso del Negocio, las reglas del Negocio y los términos que se manejan en el Negocio. Es importante definir los límites del Negocio a ser modelado y quién y qué interactúa con el Negocio, describir los procesos del Negocio y plasmarlos en diagramas de actividad en el Modelo de Casos de Uso del Negocio.

El propósito de la disciplina de Diseño en la metodología SOA definida agrega los siguientes objetivos a los determinados en el proceso base: identificar y catalogar los servicios necesarios para cumplir con los procesos identificados de la Organización en los Casos de Uso del Negocio definidos, especificar estos servicios especificando las interfaces y sus operaciones, así como los componentes que los implementarán y la reutilización de otros servicios y componentes dentro de la Organización. Un aspecto importante en 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.

2.3.1.2. Disciplina de Diseño

A continuación se describen las actividades propuestas en la Disciplina Diseño, se presenta la actividad Identificar y categorizar servicios, especificar servicios, investigar servicios existentes, asignar servicios a componentes, y definir orquestación de servicios.

◆ Identificar y categorizar servicios.

Objetivo: Esta actividad tiene como objetivo identificar los servicios necesarios para realizar los procesos del negocio, clasificándolos según el tipo de servicio: básicos, intermediarios, centrados en procesos y públicos empresariales.

Descripción: Teniendo en cuenta la Descripción de la Arquitectura en sus vistas de Casos de Uso del Sistema y 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, por ejemplo clientes. De la misma forma se identifica claramente en que subsistema se realiza cada Caso de Uso del Sistema definido, comenzando con los Casos de Uso

identificados como relevantes para 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 correspondientes a los mismos. Se brinda una guía para clasificar los servicios en básicos, intermediarios, centrados en procesos y públicos empresariales, como apoyo para realizar la identificación de los mismos.

◆ **Especificar servicios.**

Objetivo: Esta actividad tiene como objetivo especificar los servicios identificados, definiendo los contratos de servicio para cada uno incluyendo las interfaces que brindará y sus operaciones, parámetros, etc.

Descripción: Para cada servicio identificado se realiza el contrato funcional especificando para cada interfaz definida los métodos que deben ser implementados para proveer el servicio acordado para la interfaz.

◆ **Investigar servicios existentes.**

Objetivo: Esta actividad tiene como 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 más reutilizados son los servicios básicos.

Descripción: Una vez que los servicios necesarios han sido identificados y especificados, se deben investigar los servicios con que se cuenta en la Organización, tanto básicos como de otros niveles, para reutilizar en la aplicación en desarrollo. 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 como los de acceso a datos, autenticación, seguridad, sesión, etc. deberán ser reutilizados por todos los servicios definidos en niveles superiores de la SOA.

◆ **Asignar servicios a componentes.**

Objetivo: Esta actividad tiene como objetivo definir los componentes que deberán ser implementados para proveer los servicios especificados, esta correspondencia no tiene porque ser con los subsistemas identificados previamente. Asimismo se debe especificar para cada componente los servicios que provee.

Descripción: Para asignar servicios a componentes se debe responder la pregunta de quién (cual componente) 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 reglas del Negocio y servicios que implementa, así como otros componentes que utiliza.

◆ **Definir orquestación de servicios.**

Objetivo: Esta actividad tiene como 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 utilizará preferiblemente una herramienta BPMS.

2.3.1.3. Disciplina Implementación.

El propósito de la disciplina de Implementación en la metodología SOA propuesta agrega los siguientes objetivos a los definidos en el proceso base: implementar los componentes proveedores de servicios según lo establecido por la asignación de servicios a componentes. A continuación se presenta la actividad Implementar servicios incluyendo su objetivo y descripción.

◆ **Implementar servicios.**

Objetivo: Esta actividad tiene como objetivo implementar los 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, con ligamiento en tiempo de desarrollo o de ejecución).

Descripción: En esta actividad se deben 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 interfaz los métodos definidos respetando los parámetros (nombre, tipo) especificados, valor retornado (nombre, tipo) y la utilización de otros servicios para la realización de las operaciones definidas, según lo establecido en la especificación de las interfaces y en los componentes definidos para la implementación de servicios.

2.4. Resultado de las entrevistas.

2.4.1. Técnica para la obtención de los datos.

Para la obtención y procesamiento de los datos se usó la técnica de la entrevista. Esta técnica consiste en obtener los datos necesarios a través de un diálogo entre dos personas, el investigador; el cual pasa a jugar el rol de entrevistador y el entrevistado, quien generalmente es una persona entendida en la materia de la investigación que se realiza. [38]

La entrevista es una técnica muy antigua e importante y con el paso del tiempo ha sido indispensable su utilización porque permite obtener datos que de otro modo serían muy difíciles de conseguir.

Una entrevista posee objetivos primordiales, sin los cuales carece de sentido su realización, el primero y más importante es obtener información sobre el objeto de estudio, se debe tener en cuenta que se realiza para describir con objetividad situaciones o fenómenos, contribuyendo de esta manera a que se logren hallazgos y que al final favorezca al proceso de planteamiento de una o varias soluciones del problema en cuestión [39].

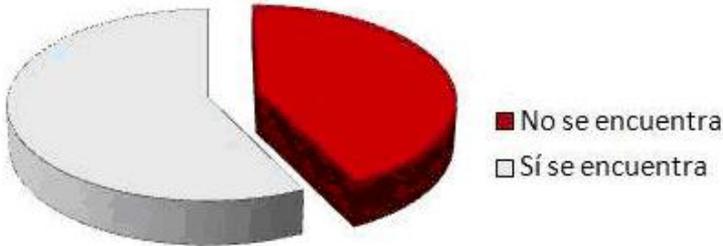
2.4.2. Procesamiento de la información.

La entrevista (Anexo I) se aplicó a siete personas. Para la selección del personal a entrevistar se tuvieron en cuenta los cargos que desempeñan en la universidad, los años vinculados a la UCI, así como su participación en eventos científicos, se tuvo en cuenta el nivel de experiencia y conocimientos sobre SOA.

El proceso de la realización de la entrevista arrojó como resultado que la UCI no se encuentra aún del todo preparada para crear y adoptar una SOA que soporte su infraestructura tecnológica. El 100 % de los entrevistados reconoció que el nivel de conocimientos de SOA por parte del personal vinculado a proyectos productivos, profesores y estudiantes de la UCI es muy bajo, en la universidad no se cuenta con especialistas en SOA y la gran mayoría de las personas que usan el concepto poseen una idea errónea de lo que ciertamente representa SOA.

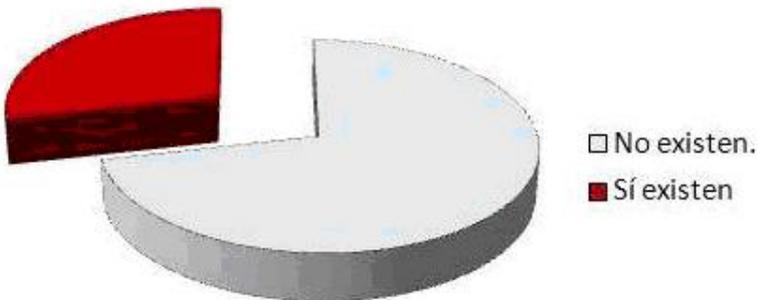
El 71.4 % del total de los entrevistados sugirió la idea que se debe comenzar a introducir los conceptos relacionados con SOA en la docencia, pues este tipo de arquitectura impone un cambio en la cultura y los hábitos de los desarrolladores de software, no solo meramente tecnológico, sino también un cambio de mentalidad que pone en claro un enfoque hacia el negocio y los objetivos empresariales. Los entrevistados alegaron que vincular SOA en la docencia fomentaría a que estudiantes y profesores se relacionen con los temas que abarca este tipo de arquitectura, lo cual propiciaría que el proceso del cambio cultural que impone SOA se acate con menos trauma en todos los niveles.

El 42.8% de total de los entrevistados concuerdan que la universidad no se encuentra todavía en el primer nivel de madurez de una SOA, afirmando que hasta el momento el personal se está capacitando y familiarizando con los temas y conceptos que incluye mientras el 57.2% del total de los entrevistados coinciden que en la UCI SOA se encuentra en el inicio del primer nivel de madurez (ver gráfica 2.1).



2.1 Representación en % del criterio sobre el nivel de madurez en que se encuentra la UCI

El 71.5% de los consultados plantean que en la UCI no se ha realizado aún un análisis y revisión de los procesos para identificarlos y describirlos, mientras que el otro 28.5% señalan que existen varios procesos descritos, aunque estos son muy pocos (ver gráfica 2.2).

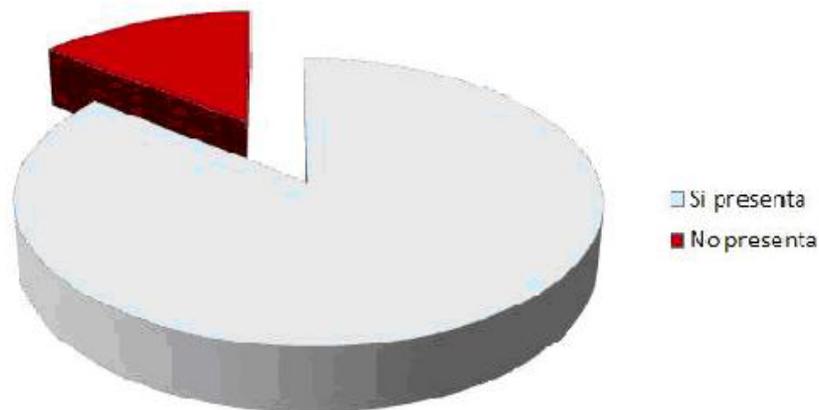


2.2 Relación en % del criterio sobre la existencia de procesos descritos en la UCI.

El 100% de los entrevistados coinciden que en la UCI hay servicios definidos, aunque son muy pocos con respecto a todos los servicios que se deben definir.

El 14.3% del total encuentra que la UCI no presenta problemas en la integración en las diferentes áreas de trabajo, mientras que el resto alegan que en la universidad no se encuentran integradas todas las fuerzas necesarias para comenzar a pensar en una SOA, dado a que muchas de las áreas no tienen públicos sus flujos de trabajo y no se ha logrado un mecanismo factible para la integración, ejemplo de ello lo constituyen dos áreas

fundamentales de la UCI como la docente y la de producción, que aún no logran integrarse de forma eficiente para resolver los problemas de la universidad (ver gráfica 2.3).



2.3 Relación en % del criterio sobre la integración de las áreas de trabajo en la UCI.

El 100 % del total de los entrevistados concuerdan en que es de vital importancia comenzar a trabajar en las bases de la adopción de una SOA para resolver los problemas de integración, flexibilidad, reusabilidad de los sistemas, alegando que la UCI se encuentra en una situación favorable para comenzar a pensar en una SOA que soporte la Infraestructura Tecnológica y que el proceso de implantación de SOA serviría de apoyo para fidelizar clientes a partir de los resultados que se logren.

2.5. Solución Propuesta.

2.5.1. Necesidad de adoptar una SOA.

Las empresas con el transcurso del tiempo han tenido la necesidad de buscar mejoras en las comunicaciones y en la calidad de vida para poder subsistir en un mercado competitivo como el actual, esto ha traído como resultado que cada vez se haga más evidente la orientación a los servicios y no al proceso.

Los enfoques tradicionales de integración a largo plazo no resultan la solución más viable para organizaciones complejas como la UCI, sin embargo es históricamente la solución más empleada, la cual solo ostenta un futuro donde la reutilización y la escalabilidad de los sistemas es limitado, sin lograr una gestión centralizada y visibilidad en los procesos del negocio. De forma opuesta SOA ofrece un marco flexible de aplicaciones que rompe con el concepto monolítico anterior de evolucionar a componentes distribuidos para que las organizaciones resuelvan sus problemas de integración, obtengan mejoras en los procesos de contacto con el cliente, reduzcan el tiempo de lanzamiento de productos, aprovechen la infraestructura existente, los sistemas, el conocimiento adquirido y de esta forma puedan minimizar los costes de integración, desarrollo y operacionales.

Los cambiantes mercados, las presiones competitivas, la urgente necesidad de evolucionar, la obtención de mayor flexibilidad, entre otros muchos factores, hace que surja en la UCI la imperiosa urgencia de emigrar hacia nuevos desarrollos tecnológicos, siendo la única manera con que se pueda dar respuesta rápida a los clientes y resolver la caótica situación en que la UCI se encuentra sumergida. Con SOA se logran resolver estos problemas, lo que hace que constituya la mejor solución para lograr integración y responder a todas estas necesidades que han surgido y a otras nuevas que surgirán con el paso del tiempo.

2.5.2. Necesidad de una guía.

Cuando se habla de adoptar una SOA no se trata solamente de realizar algunas modificaciones y extensiones a los sistemas actuales. Para lograr y obtener ventajas de las posibilidades de SOA es necesario que se considere y se planee todo el ciclo completo. Este tipo de arquitectura debe implementarse de forma que persiga una línea de evolución en la que se debe tener en cuenta el estado actual y el futuro de la empresa, ofreciendo la posibilidad de una implementación gradual y progresiva que integre personas, procesos y datos. Un enfoque SOA para determinada empresa implica muchos cambios y supone esfuerzo continuo lo que indica que se debe planificar y ejecutar el proyecto siguiendo una línea de trabajo que oriente su perfeccionamiento y desarrollo. SOA involucra tiempo, tecnologías de punta, nuevos conceptos, personal capacitado. Desarrollar un proyecto de tal

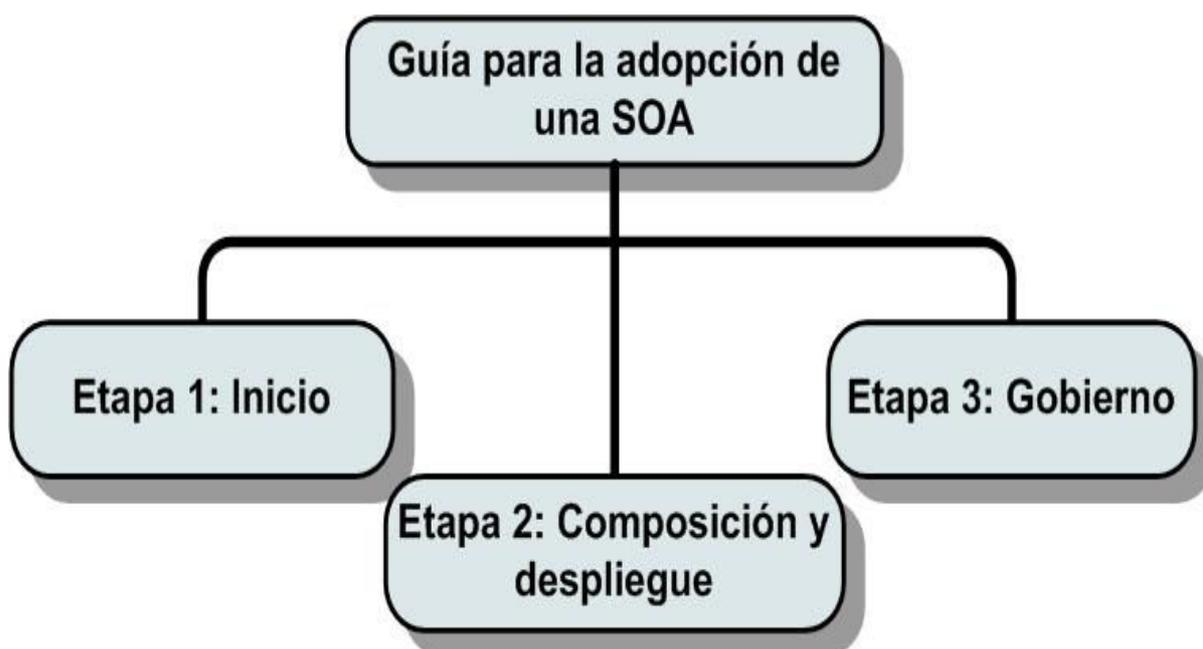
magnitud necesita una guía que oriente su desarrollo de forma correcta y organizada y cree soluciones a través de pasos discretos, continuos e inteligentes.

2.5.3. Propuesta de una guía para la adopción de SOA.

En este epígrafe se propone una guía compuesta por pasos fundamentales que se deben tener en cuenta para la adopción de SOA en la UCI. Esta propuesta tiene como objetivo asesorar y supervisar la construcción y desarrollo de este tipo de arquitectura desde principio a fin, siguiendo una línea de trabajo profunda basada en las condiciones actuales que presenta la universidad. Con ello no se pretende dar resultados mágicos de la noche a la mañana pero sí que oriente cómo trabajar durante el proceso de adopción de SOA.

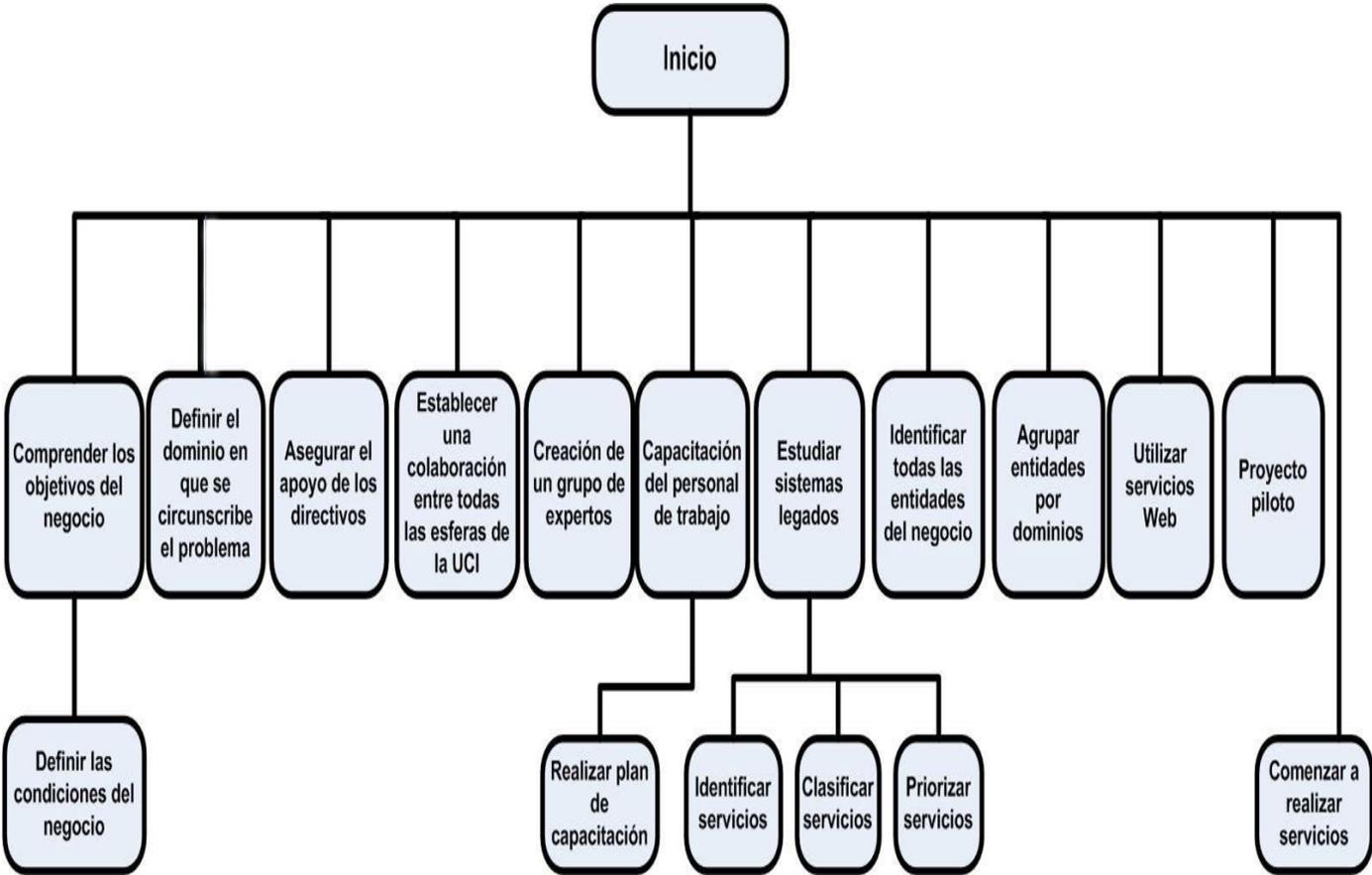
2.5.4. Guía para la adopción de SOA.

La guía propuesta plantea lograr soluciones en pasos discretos, iterativos e incrementales que abarquen todo el ciclo de vida de desarrollo de un proyecto SOA. Esta guía vincula las condiciones actuales que presenta la UCI y está compuesta por tres etapas fundamentales las cuales poseen los pasos básicos que se deben tener en cuenta para el logro y mantenimiento de una SOA.



2.5.4.1. 1ra Etapa: Inicio.

La etapa de inicio es muy importante para lograr la adopción exitosa de SOA, es aquí donde se deben identificar con claridad los problemas que conlleven a la empresa, en este caso la UCI; a adoptar un proyecto SOA y comenzar a pensar en el alcance del mismo. A partir de esta etapa tendrán lugar los primeros cambios organizacionales, especialmente en las estructuras administrativas y posteriormente involucrará a todo el personal, que servirá de apoyo e interactuará con SOA, el cual se irá capacitando. En esta temprana etapa es donde debe lograrse una colaboración a escala empresarial, es decir, directivos y equipos administrativos deben integrarse y comenzar a trabajar en conjunto para alcanzar el éxito.



1. Comprender los objetivos del negocio: Identificar los problemas más prioritarios que tiene la UCI. Cuanto más precisa sea esta información más fácilmente se podrá delimitar el alcance y la dirección que debe tener el proyecto.

◆ **Definir las condiciones del negocio:** Definir primeramente qué es lo que se quiere lograr y después especificar cómo.

2. Definir el dominio en el que se circunscribe el problema: Definir el ámbito en el que se aplicará SOA dentro de la UCI. Además de definir el ámbito, tenemos que pensar en qué SOA se implementa mejor si empezamos poco a poco, es decir, vamos en lugar de atacar el dominio completo vamos atacando pequeñas partes del mismo.

3. Asegurar el apoyo de los directivos: Convencer a los directivos de la UCI la importancia que representa para un desarrollo de SOA contar con su máximo apoyo, es primordial mostrarles los éxitos y fracasos de otras empresas que han intentado adoptar SOA para tomar experiencia y disminuir posibles errores.

4. Establecer una colaboración entre todas las esferas de la UCI: Implica vincular a todas las esferas tecnológicas que posee la UCI con el objetivo de adoptar SOA, es decir alinear a toda a todas las esferas para que trabajen y piensen de un nuevo modo.

5. Creación de un grupo de expertos: Seleccionar el personal de la UCI que posea mayor conocimiento en SOA para vincularlos al desarrollo del proyecto SOA y estimularlos a transmitir sus conocimientos a otras personas.

6. Capacitación del personal de trabajo: Adiestrar el personal que trabajará en el proyecto SOA, dado que actualmente la UCI no cuenta con suficiente personal capacitado en el tema.

◆ **Realizar plan de capacitación:** Elaborar un plan detallado que contenga los cursos y conferencias sobre SOA, el cual vincule además sus horarios, y el personal encargado de impartirlas.

7. Estudiar sistemas legados. Consiste en analizar y revisar los sistemas actuales de la UCI para adquirir de ellos los servicios que se puedan reutilizar y habilitarlos para implementar la SOA. Este paso incluye:

- ◆ **Identificar servicios de los sistemas de legado:** Cuáles son los servicios que podrían servir para el proyecto SOA.
- ◆ **Clasificar servicios de los sistemas de legado:** Agrupar los servicios de acuerdo a su clasificación.
- ◆ **Priorizar servicios de los sistemas de legado:** Identificar los servicios más importantes y priorizarlos. Los servicios priorizados serían los primeros en construirse.

8. Identificar todas las entidades del negocio: Las entidades del negocio de una aplicación representan entidades reales y suelen ser sustantivos, algunos ejemplos de entidades serían Persona, Nómina, Factura, Depósito, etc. Se deben identificar todas las entidades del negocio para integrarlas a la solución SOA.

9. Agrupar las entidades por dominios: Clasificar las entidades del negocio y congregadas de acuerdo a características comunes.

10. Utilizar Web services: Establecer la comunicación entre los servicios a través de la red.

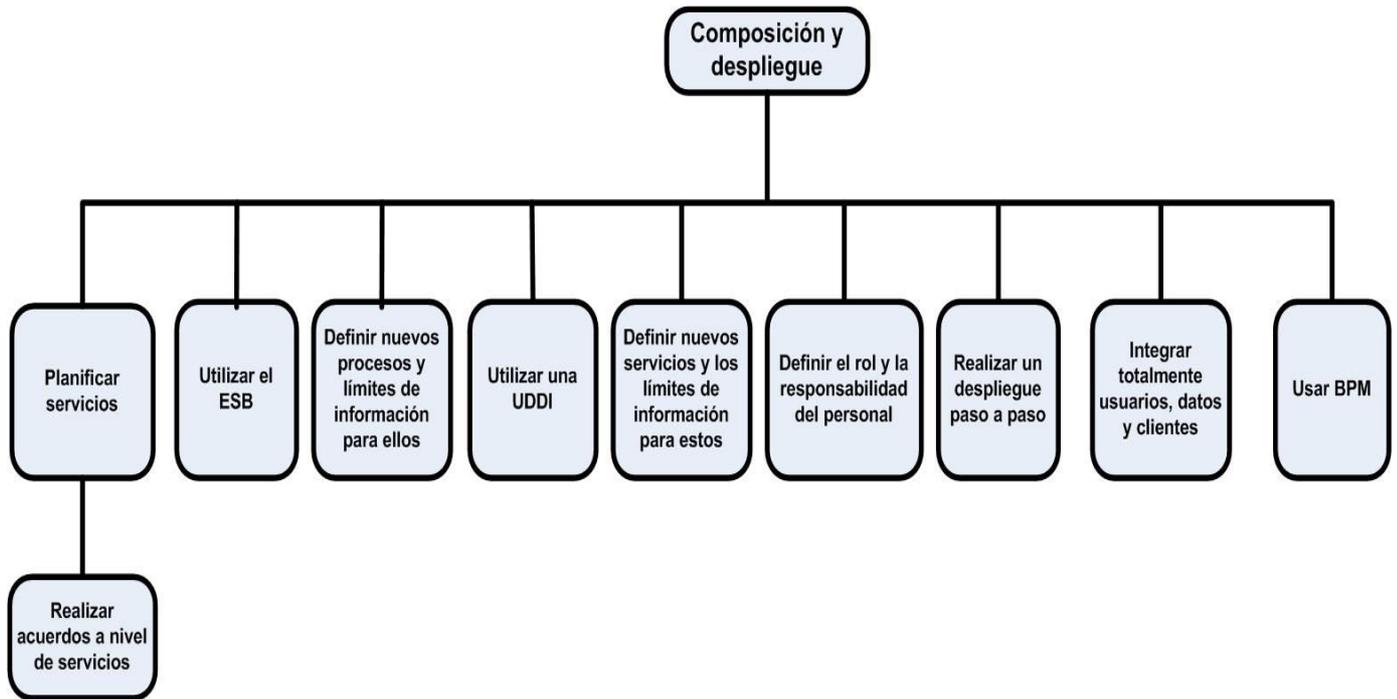
11. Proyecto piloto: Consiste en la creación de un equipo que comience a construir los servicios inmediatos para la adopción de SOA aplicándosele técnicas para la evaluación del proyecto; esto permitiría minimizar riesgos de fallos en la implementación final.

12. Comenzar a realizar servicios: Iniciar las implementaciones de los servicios inmediatos con que contará SOA.

2.5.4.2. 2da Etapa: Composición y despliegue.

Antes de comenzar a trabajar en esta etapa se debe hacer una revisión de los objetivos que se lograron cumplir en la primera etapa y si los resultados son factibles entonces se comienza a trabajar en la composición de los elementos fundamentales que engloba este tipo de arquitectura.

La etapa de composición y despliegue tiene como objetivo fundamental lograr que comience a funcionar SOA en la UCI y que los servicios comiencen a ser consumidos por los usuarios. Es una etapa muy importante, pues es donde se comienzan a obtener resultados tangibles que indicarán la medida de lo bien que se han desarrollado el proyecto SOA.



1. Planificar los servicios: Alinear los servicios a los intereses de la organización e incluye:

◆ **Realizar los acuerdos a nivel de servicios:** Un acuerdo a nivel de servicio (SLA) es un paso fundamental a tener en cuenta, pues tratan de mantener y garantizar la calidad de un servicio brindado a un cliente. Para que se tenga una mejor idea, se brindan a continuación algunos de los elementos a incluir en un SLA.[40]

- Horario del servicio.
- Tiempo de respuesta.
- Responsabilidades y obligaciones del área de soporte.
- Tiempo de vigencia del acuerdo SLA.
- Responsabilidad y obligaciones de los usuarios.

2. Utilizar un ESB: Este paso es muy importante ya que los ESB ofrecen los medios de conectividad necesarios para integrar y orquestar los servicios en SOA. Trabajando conjuntamente SOA y un ESB pueden ayudar a reducir la cantidad de interfaces complejas en la UCI.

3. Definir nuevos procesos y los límites de la información para estos: Tratar de diseñar nuevos procesos de negocio que automaticen la interacción entre servicios, así como los flujos de información. No se trata tanto de crear nuevas funcionalidades, sino de orquestar servicios y flujos de información existentes.

4. Utilizar una UDDI: Comenzar a utilizar una UDDI para almacenar los servicios que se han desarrollado y compartir información sobre SOA que pueda ser requerida por otras áreas o departamentos de la UCI.

5. Definir nuevos servicios y los límites de información para estos: Continuar definiendo servicios que le seguirán agregando valor a SOA.

6. Definir el rol y la responsabilidad del personal: Asignar, a cada integrante en el desarrollo del proyecto SOA, el papel que desempeñará; esto permite que se pueda lograr un mayor nivel de responsabilidad y compromiso con los intereses de la UCI.

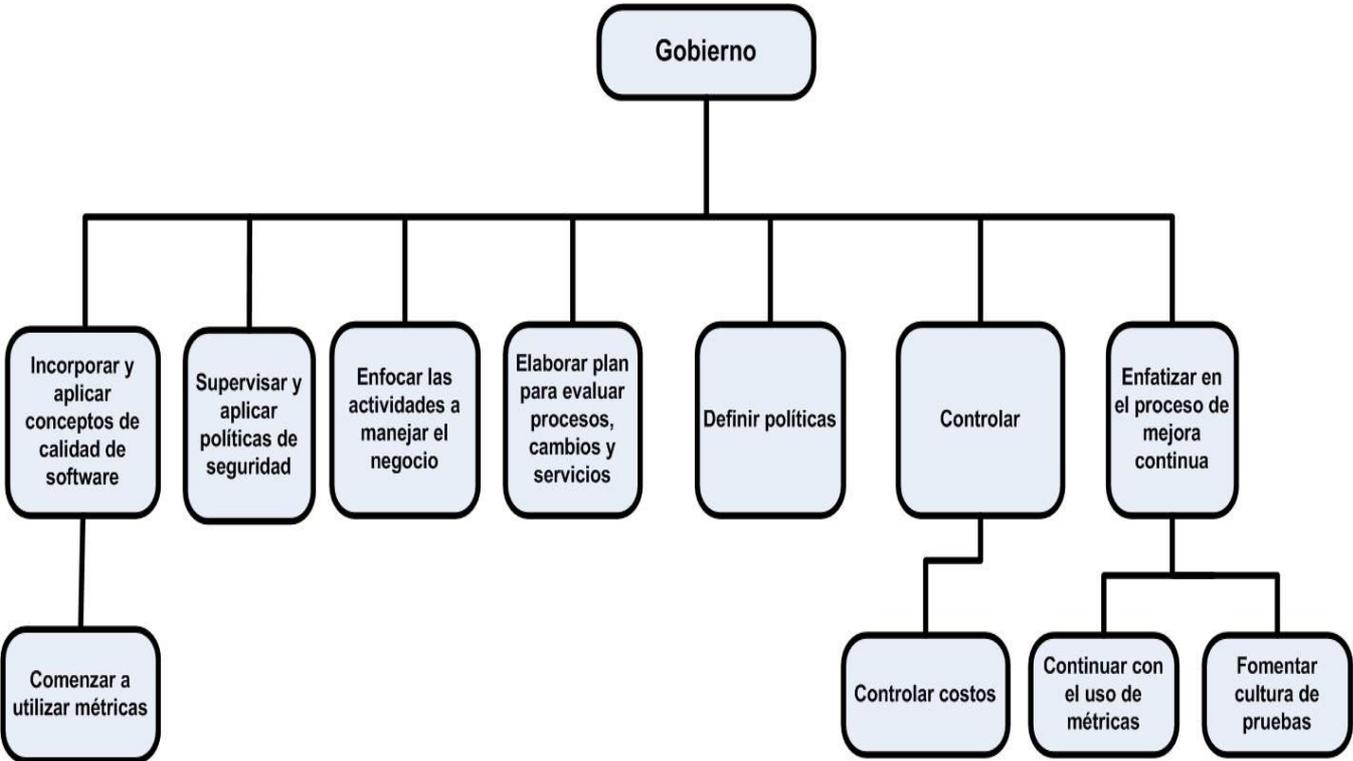
7. Realizar un despliegue paso a paso: Cuando se piensa en el despliegue de SOA se debe tener en cuenta que es mejor comenzar por un departamento o área determinada y después debe abarcar toda la UCI.

8. Integrar totalmente usuarios, datos y clientes: Una vez realizado el despliegue en un departamento o dominio del negocio, se deben integrar los usuarios, datos y clientes para comenzar luego a realizar pruebas enfocadas a calificar el funcionamiento de SOA.

9. **Usar BPM:** Para facilitar la modificación, reconfiguración y optimización de definiciones de procesos con herramientas gráficas que pueden ser utilizadas por analistas del negocio con menor orientación tecnológica.

2.5.4.5. 3ra Etapa: Gobierno

Cuando se llega a esta etapa es porque la adopción SOA ha avanzado por buen camino y el personal ha madurado, no solo en conocimientos sobre SOA sino que también han logrado llevar a la práctica conceptos relacionaos a este tipo de arquitectura, lo que hace que se encuentren mejor preparados y puedan comenzar a mejorar los procesos. El aumento de la flexibilidad requiere que las organizaciones establezcan una infraestructura para implementar la toma de decisiones, una capacidad de servicio mejorada y una mejor comunicación. El gobierno SOA ayuda a que se configure un mecanismo sólido para la toma de decisiones y las relaciones entre servicios, además de cumplir con las leyes, políticas, estándares y procedimientos a los cuales está sometida una organización. Ya en esta etapa se comienza a apreciar rendimiento en las inversiones realizadas en SOA, reduciéndose de gran manera los costes de coordinación y se comienza a medir la eficacia de SOA.



1. Incorporar y aplicar el concepto de calidad de software: La calidad de un software es el conjunto de cualidades que lo identifican y determinan su utilidad. Es imprescindible incorporar y aplicar el concepto de calidad de software ya que esta asegura una serie de mecanismos con los que a largo plazo se logra el éxito de un proyecto.

- ◆ **Comenzar a utilizar métricas:** Comenzar a utilizar todo tipo de métricas para caracterizar, medir y evaluar el desarrollo SOA en la UCI.

2. Supervisar y aplicar políticas de seguridad: Crear y utilizar políticas de seguridad para prevenir, proteger y manejar los distintos riesgos que en un futuro pueda resultar un problema de la solución SOA.

3. Enfocar las actividades en manejar el negocio: Consiste en orientar las actividades principalmente hacia la mejora en el manejo del negocio, la automatización y el cambio.

4. Elaborar un plan para evaluar los procesos, cambios y los servicios: Consiste en la elaboración de un plan estratégico que posea una serie de elementos fundamentales que de deben tener en cuenta para lograr un óptimo manejo de los procesos, cambios y servicios, paralelamente debe contener una misión y visión del negocio en general, los objetivos que se persiguen y las estrategias.

5. Definir políticas: Las políticas brindan cierto grado de direccionalidad para guiar la toma de decisiones, es por ello que para un buen mantenimiento y desarrollo SOA se considera de suma importancia que se declaren políticas.

- ◆ Definir políticas para mejorar la adquisición de servicios.
- ◆ Definir políticas para mejorar la estandarización de datos.
- ◆ Definir políticas para mejorar la consistencia en la aplicación.
- ◆ Definir políticas tácticas.
- ◆ Definir políticas para código fuente.
- ◆ Definir políticas arquitecturales.

- ◆ Definir políticas para ciclo de vida de los servicios.

6. Controlar: Controlar es fundamental para la UCI porque aunque esta cuente con los planes, estructura organizacional adecuada y una dirección eficiente solo se podrá verificar la situación en que se encuentre el desarrollo SOA con mecanismos que cercioren e informen que los hechos son de acuerdo con los objetivos que se plantearon inicialmente.

- ◆ **Controlar costos:** Realizar y llevar a la práctica mecanismos para que se minimicen los costos que se han ido concertando con la adopción de SOA.

7. Enfatizar en el proceso de mejora continua: La continua mejora de la capacidad y los resultados de la organización, en este caso la UCI, debe ser un objetivo permanente y debe alcanzarse en todos los campos, por lo que el proceso de desarrollar y adoptar SOA no escapa de ello. Incluye explotar al máximo las capacidades del personal, mejorar los procesos productivos, entre otros muchos factores y debe ser un proceso cotidiano para que se logre llegar a ser un hábito. Todo este proceso incluye:

- ◆ **Continuar con el uso de métricas:** Implica utilizar métricas que se definieron con anterioridad y definir y aplicar otras más para monitorizar, controlar y mejorar SOA.

- ◆ **Fomentar cultura de pruebas:** Destinar un personal estrechamente vinculado a SOA a formar un equipo para que se realicen todo tipo de pruebas destinadas a encontrar posibles huecos o fallas del sistema SOA para que sean depuradas. Incluye llevar a la práctica la mayor cantidad de conceptos vinculados a los tipos de pruebas que se le deben hacer a un software.

- Realizar pruebas de diagnóstico a la infraestructura SOA para descubrir y priorizar las fallas de los sistemas.
- Realizar pruebas de mantenimiento de los procesos.
- Realizar pruebas de adaptabilidad de los procesos.
- Otros tipos de pruebas.

2.6. Conclusiones Parciales.

En el desarrollo de este capítulo se analizaron los aspectos necesarios para realizar una guía que pueda permitirle a la UCI la adopción de forma segura una SOA. Además se hizo un análisis profundo de las metodologías y buenas prácticas propuestas o desarrolladas por las empresas líderes.

Una vez culminada la investigación y desarrollada la guía para la adopción de una SOA, se puede decir que se cumplió con el objetivo general propuesto. Con esta guía se pretende que las personas puedan contar con toda la información de apoyo necesaria, cumpliendo satisfactoriamente los objetivos. Se considera de gran importancia el estudio de la guía pues aunque existen algunos documentos referentes al tema tratado, estos se encuentran muy dispersos y en idioma inglés.

CAPÍTULO 3: VALIDACIÓN DE LA PROPUESTA

3.1. Introducción.

En la investigación, hasta el momento, se realizó un estudio de las metodologías y buenas prácticas para SOA propuestas o desarrolladas por las empresas líderes, teniendo en cuenta la importancia y los beneficios que esta arquitectura le puede proporcionar a la UCI. Luego de este estudio se elaboró la propuesta de una guía para adoptar una Arquitectura Orientada a Servicios (SOA) en la universidad. En el presente capítulo para la validación y aceptación de la guía propuesta en el Capítulo 2, se tomó como herramienta el uso del Criterio de un Panel de Especialistas y el empleo de técnicas propuestas en el Método Delphi, realizándose una descripción de cómo fue ejecutado el método y los resultados que fueron obtenidos.

3.2. Método de Evaluación de Expertos.

3.2.1. Antecedentes históricos.

El Método Delphi, considerado como una de las técnicas subjetivas de pronóstico más confiables, conocido también como Delfos, nombre cuyo origen proviene del oráculo de la antigua Grecia, fue creado alrededor de los años 1963-1964 por la Rand Corporation, y específicamente por Olaf Helmer, Dalkey y Gordon, con el objetivo de elaborar pronósticos a largo plazo, referentes a posibles acontecimientos en varias ramas de la ciencia, la técnica y la política. Así pues el Método Delphi es la utilización sistemática del juicio intuitivo de un grupo de expertos para obtener un consenso de opiniones informadas. [41]

Este método consiste en la organización de un diálogo anónimo entre los expertos consultados individualmente, mediante cuestionarios, con vistas a obtener un consenso general del tema que se está investigando.

3.2.2. Características del método Delphi.

Este método presenta tres características fundamentales:[42]

Anonimato: Durante un Delphi, ningún experto conoce la identidad de los otros que componen el grupo de debate. Esto tiene una serie de aspectos positivos, ya que impide la posibilidad de que un miembro del grupo sea influenciado por la reputación de otro de los

miembros o por el peso que supone oponerse a la mayoría. La única influencia posible es la de la congruencia de los argumentos.

Respuesta del grupo en forma estadística: La información que se presenta a los expertos no es sólo el punto de vista de la mayoría, sino que se presentan todas las opiniones indicando el grado de acuerdo que se ha obtenido.

3.3. Fases del método Delphi.

3.3.1. Elaboración del objetivo.

Formulación del objetivo de la evaluación por los especialistas: Valorar la guía propuesta para adoptar una Arquitectura Orientada a Servicios (SOA) en la UCI y la efectividad que se espera alcanzar con su aplicación.

3.3.2. Selección de los especialistas.

Se seleccionaron 7 especialistas, tomando como criterio de selección la efectividad profesional que realizan, la experiencia que poseen en el campo en que se desarrolla la propuesta, los años vinculados a la UCI y por su participación en diversos eventos relacionados con SOA. (Anexo II)

Elegir los especialistas atendiendo a las características mencionadas propicia obtener resultados con un alto nivel de calidad, junto a otras cualidades propias de éstos como pueden ser: la seriedad, la honestidad y la responsabilidad que hacen que las opiniones brindadas por ellos sean confiables y válidas para el objetivo propuesto.

3.3.3. Elaboración del cuestionario.

El cuestionario consta de seis preguntas de enfoque investigativo, sobre la validez de la solución propuesta al problema planteado. Estas preguntas le brindan la ventaja a los especialistas de proporcionar una mayor riqueza en las respuestas ofrecidas. También se le

dio la posibilidad de presentar su opinión general del objeto de evolución, para que tuvieran la libertad de expresar todo lo que se pudo obviar en la encuesta. (Anexo III)

En todo los casos los especialistas recibieron la documentación de la guía propuesta y se les requirió cumplir con un lapso de tiempo determinado para dar las respuestas o hacer las preguntas pertinentes que les hubiesen surgido al estudiar el documento presentado.

3.3.4. Resultados de la evaluación.

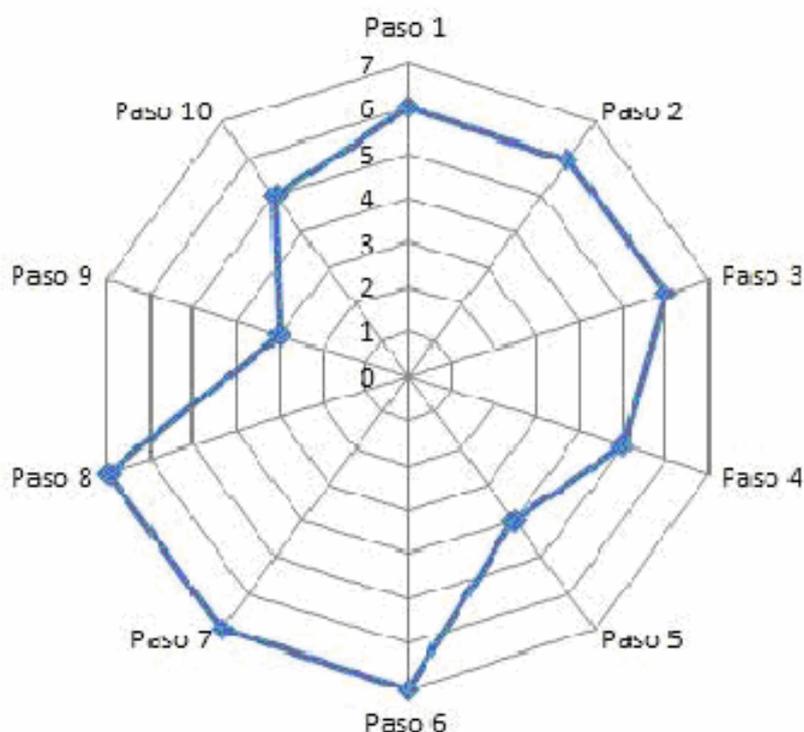
A partir de los resultados obtenidos en los cuestionarios se pudo determinar que los expertos estuvieron de acuerdo con la guía propuesta.

El 100 % de los expertos consideraron que es muy importante para la UCI adoptar una SOA, ya que esta forma se lograría comenzar a resolver problemas de integridad y flexibilidad de los sistemas que soportan la infraestructura tecnológica de la universidad y brindaría la posibilidad de una apertura a nuevos horizontes del mercado del software, porque SOA es un término de moda en el mundo actual y muchas empresas comienzan a notar las posibilidades que brinda.

En el cuestionario se indagó sobre cuáles pasos serían imprescindibles o primordiales para implantar una SOA en la UCI. En muchas respuestas hubo variabilidad en escoger los pasos más importantes, pero se pudieron definir de todas las respuestas los pasos que más prevalecieron y que son mostrados en la Tabla 3.1. En ella se reflejan las coincidencias que tuvieron las especialistas marcadas con un número. En caso que no involucraran cierto paso en sus respuestas o el paso que citaron no poseía ninguna otra coincidencia entonces se coloca 0 en la casilla correspondiente. En el gráfico 3.1 se muestra la totalización de las coincidencias que tuvieron los expertos y a la vez se puede definir el nivel de importancia que estos pasos poseen en la guía propuesta.

Pasos	Esp.1	Esp.2	Esp.3	Esp. 4	Esp.5	Esp.6	Esp.7	Total
1. Definir las entidades del negocio.	1	1	1	1	1	1	0	6
2. Distribuir las entidades en dominios.	1	0	1	1	1	1	1	6
3. Crear servicios de negocio a partir de las entidades.	1	1	0	1	1	1	1	6
4. Proyecto piloto.	1	0	0	1	1	1	1	5
5. Crear servicios Web.	1	0	1	0	0	1	1	4
6. Utilizar ESB.	1	1	1	1	1	1	1	7
7. Utilizar UDDI.	1	1	1	1	1	1	1	7
8. Utilizar BPM.	1	1	1	1	1	1	1	7
9. Definir bien como debe de quedar la arquitectura a medio y alto nivel.	0	0	1	0	1	1	0	3
10. Enfatizar en el proceso de mejora continua.	1	1	0	1	1	1	0	5

Tabla 3.1 Pasos propuestos para la adopción de una SOA.



3.2 Distribución radial de la importancia dada a los pasos para adoptar una SOA.

Como se puede apreciar en el gráfico, la gran mayoría de los especialistas coincidieron en los pasos propuestos para la adopción de una SOA en la UCI.

Referente a la tercera pregunta del cuestionario, los especialistas alegaron que la aplicación de la guía propuesta cumpliría su objetivo y facilitaría la forma de realizar las cosas. También consideraron que ayudaría a que el conocimiento de SOA se difundiera a todos los niveles.

En relación a la cuarta pregunta los especialistas fundamentaron que hay muchos elementos a favor de una correcta aplicación de la guía propuesta, puesto que existe la tendencia en la UCI a utilizar los trabajos de los estudiantes y seguir agregándoles valor, así como que sería la primera guía con que cuenta la UCI enfocada a comenzar a adoptar una SOA. Otro elemento a los cuales muchos especialistas se refirieron es al hecho que la UCI se encuentra enmarcada en un plan de orientación de metas para la organización y mejora de sus

procesos y con esta guía se ayudaría a que el proceso de adopción de SOA sea más cómodo y factible. Por otra parte, también brindaron ejemplos de elementos que se opondrían a la aplicación de la guía refiriéndose en la gran mayoría de los casos a la consecuencia que puede acarrear un desconocimiento del tema de SOA, esto hace que muchas personas piensen de modos muy diferente cuando se habla de este tipo de arquitectura, lo que provoca discrepancias y dificultades a la hora de aplicar la guía.

Respecto a la quinta pregunta la mayor parte de los expertos brindaron de ejemplo muchos beneficios que aportaría la guía propuesta a la UCI, entre los cuales figuraron que la guía propuesta se debe llevar a cabo mediante un proceso iterativo e incremental, lo cual favorecería a crecer con las experiencias y a agregarle con el seguimiento de la misma más valor y experiencia, además agregaron que los proyectos productivos de la universidad pueden servirse de la guía y de esta forma aumentar su conocimiento cultural respecto a cómo es que se desarrolla una SOA. La guía propuesta significa un punto de inicio para la adopción y la universidad puede usarla en su beneficio y agregarle valor.

3.4. Conclusiones.

La evaluación de la guía propuesta de la adopción de SOA en la UCI, se realizó por siete especialistas seleccionados según la efectividad de la actividad profesional que realizan y mediante el empleo de técnicas del Método Delphi. Esta evaluación permite concluir que con la aplicación de la guía en la UCI se aumentará los conocimientos culturales en los proyectos y se brindará a la UCI un punto de partida para hacer la adopción.

CONCLUSIONES

Con el deseo de darle solución a los problemas de integración y cambios mediante la guía desarrollada para la adopción de SOA en la UCI, se concluye este trabajo investigativo.

En la investigación realizada se cumplieron todos los objetivos propuestos:

- ◆ Se describió los principales conceptos relacionados con el tema que es proceso de investigación.
- ◆ Se realizaron entrevistas a especialistas en el tema para caracterizar la situación en que se encuentra la UCI con respecto a SOA.
- ◆ La guía propuesta fue evaluada y aceptada usando como herramienta el uso de un Criterio de un Panel de Especialistas y el empleo de técnicas propuestas por el Método Delphi.

RECOMENDACIONES

- ◆ Aplicar y dar seguimiento a la guía propuesta.
- ◆ Realizar un estudio de las herramientas que se puedan utilizar para desarrollar una SOA y vincularlas con la guía propuesta.

BIBLIOGRAFÍA

1. Pfleeger, S.L., *Ingeniería de Software: Teoría y Práctica*.
2. Wolf., D.E.P.y.A.L., "*Foundations for the study of software architecture*".
3. Dijkstra, E., *The Structure of THE Multiprogramming system.*" 1983.: p. 49-52.
4. Wirth., N., "*Program development by stepwise refinement*". 1971.
5. Kron, F.D.y.H., "*Programming-in-the-large versus programming-in-the-small*". *IEEE Transaction in Software Engineering*. 1976.
6. Sharp., I.P., *Comentario en discusión sobre teoría y práctica de la ingeniería de software en conference de NATO Science Committee*. 1969.
7. Clements., P., "*Coming attractions in Software Architecture*". 1996.
8. Frank Buschmann, R.M., Hans Rohnert, Peter Sommerlad y Michael Stal., *Patternoriented software architecture - A system of patterns*. 1996.
9. Fielding, R.T., "*Architectural styles and the design of network-based software architectures*". 2000.

Disponible en: <http://www.sei.cmu.edu/architecture/definitions.html>

10. Woods, E., *Software Architecture for Software-Intensive Systems*.

Disponible en: <http://www.sei.cmu.edu/architecture/definitions.html>

11. Clements, P., *A Survey of Architecture Description Languages*". *Proceedings of the International Workshop on Software Specification and Design*. 1996.
12. Garlan, D., "*Software Architecture: A Roadmap*". 2000.
13. ANSI/IEEE, *ANSI/IEEE Std 1471-2000, Recommended Practice for Architectural Description of Software-Intensive Systems*.
14. MSDN, M., *Microsoft MSDN-Mediacenter*.

Disponible en:

<http://www.microsoft.com/spanish/msdn/latam/mediacenter/webcast/architect.aspx>

15. SOA, *Principios de diseño de servicios : patrones y antipatrones de servicios*.

Disponible en:

<http://www.microsoft.com/spanish/msdn/articulos/archivo/121205/voices/SOADesign.mspix>

16. REYNOSO, B., *Arquitectura Orientada a Servicios*.
17. Servicios, *Definicion Concepto Servicios*.

Disponible en: www.mitecnologico.com/Main/DefinicionConceptoServicios

18. Delgado, A., *Metodología de desarrollo de aplicaciones con enfoque SOA (Service Oriented Architecture)*. 2005.

Disponible en: <http://www.fing.edu.uy/~adelgado/ExtensionSOA/index.htm>

19. Server, W., *Windows Server 2003 UDDI Services*.

Disponible en:

<http://www.microsoft.com/windowsserver2003/technologies/idm/uddi/default.msp>

20. Krafzig, D.B., K. Slama, D., *Enterprise SOA, Service Oriented Architecture Best Practices*, Prentice Hal. 2005.

21. Corporation, M., *La Arquitectura Orientada a Servicios (SOA) aplicada al mundo real*. 2006.

22. TIBCO, *El papel de un bus de servicios empresariales (ESB) en una SOA*.

Disponible en: http://www.tibco.com/international/spain/resources/es_esb_for_soa.pdf

23. PEOPLEWEB., *¿Qué es CMMI?* 2006.

24. Corella, J.M.F.y.M.A., *Web Services y Semantic Web Service*. 2006.

25. MET, 1999.

Disponible en: <http://www.analitica.com/vam/1999.05/ciencia/03.htm>

26. Kaplan, A., *The conduct of inquiry*. 1964.

27. Academia, R., *Real Academia de la Lengua Española*.

Disponible en: <http://buscon.rae.es/drael/html/cabecera.htm>

28. Sanchez, M.A.M., *Metodologías de desarrollo de software*. 2004.

Disponible en:

http://www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf

29. JOSÉ H. CANÓS, P.L., MA. CARMEN PENADÉS, *Metodologías Ágiles en el Desarrollo de Software*.

30. Fowler, M., *La nueva metodología*. 2003.

Disponible en: <http://www.programacionextrema.org/articulos/newMethodology.es.html>

31. Escribano, G.F., *Introducción a Extreme Programming*.

Disponible en:

<http://www.info-ab.uclm.es/asignaturas/42551/trabajosAnteriores/Trabajo-XP.pdf>

32. *Publicaciones MKM*. 2008.

Disponible en: <http://www.mkm-pi.com/mkmpi.php?article1452>

33. *SOA Governance (conceptos y diseño)*.

Disponible en: http://www.swassociates.es/s_governance.htm

34. Fernández, R., *IBM apuesta fuerte a SOA en la Región*. 2007.

Disponible en: <http://t1d.www03.cacheibm.com/industries/media/doc/content/bin/SOA>

35. Peña, J.R., *Contamos con la gama de prestaciones de administración e integración de procesos mas amplia del mercado actual*. 2005.

Disponible en: <http://www.cwv.com.ve/index.php?option=com>

36. IBM, *SOA Foundation: providing what you need to get started with SOA*. 2005.

Disponible en:

http://www.cwv.com.ve/index.php?option=com_content&task=view&id=78&Itemid=39

37. *Sitio Oficial de Software AG*. 2007.

Disponible en:

<http://www.softwareag.com/es/default.asp>

38. Puente, W., *TÉCNICAS DE INVESTIGACIÓN*.

Disponible en: <http://www.rppnet.com.ar/tecnicasdeinvestigacion.htm>

39. MI, *Metodología de Investigación*.

Disponible en: <http://www.aibarra.org/investig/tema0.htm>

40. Meza, J.D.C., *Acuerdo de Nivel de Servicio (ANS o SLA)*.

Disponible en: http://www.rcp.net.pe/downloads/Acuerdo_de_nivel_de_servicio.ppt#287

41. GORDON, H., DALKEY, *Método Delphi*. 1963-1964.

Disponible en: http://www.12manage.com/methods_helmer_delphi_method_es.html

42. Caseres, E., *El método Delphi. Características*. 2006.

Disponible en: <http://www.codesyntax.com>

BIBLIOGRAFÍA CONSULTADA

- REYNOSO, C. B. Introducción a la Arquitectura de Software, 2004.

Disponible en:

http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/intro.asp

- WADOOA SOA (Service-oriented Architecture) Arquitectura Orientada a Servicios, 2007.

Disponible en: <http://wadoo.com/doku.php/soa>

- PARRA, J. D. Hacia una Arquitectura Empresarial basada en Servicios.

Disponible en:

<http://www.microsoft.com/spanish/msdn/comunidad/mtj.net/voices/art143.asp>

- Corporation, M., La Arquitectura Orientada a Servicios (SOA) aplicada al mundo real. 2006.
- IBM, *SOA Foundation: providing what you need to get started with SOA*. 2005.

Disponible en:

http://www.cwv.com.ve/index.php?option=com_content&task=view&id=78&Itemid=39

- *Sitio Oficial de Software AG*. 2007.

Disponible en:

<http://www.softwareag.com/es/default.asp>

- Delgado, A., *Metodología de desarrollo de aplicaciones con enfoque SOA (Service Oriented Architecture)*. 2005.

Disponible en: <http://www.fing.edu.uy/~adelgado/ExtensionSOA/index.htm>

- *SOA Governance (conceptos y diseño)*.

Disponible en: http://www.swassociates.es/s_governance.htm

ANEXOS

Anexo I

Entrevista

Nombre: _____

Años vinculado a la UCI: _____

Cargo que desempeña: _____

1. **¿Considera usted que los estudiantes vinculados a proyectos productivos y profesores de la UCI posean conocimientos sobre SOA?**

Sí _____ No _____

En caso positivo, diga que por ciento del total de estudiantes vinculados a proyectos y profesores usted considera que poseen tales conocimientos.

2. **¿Conoce usted los niveles de madurez de una SOA?**

Sí _____ No _____

En caso positivo, diga en que nivel de madurez usted considera que se encuentre SOA en la universidad.

3. **¿Considera usted que en la UCI sea necesario comenzar a pensar sobre la adopción de una SOA?**

Sí _____ No _____

En caso positivo por qué motivos usted considera que sería de gran importancia para la universidad la adopción de una SOA. (Enumere al menos 3 de los que considere más relevantes).

4. ¿Se ha realizado en la UCI un análisis y revisión de los procesos para identificarlos y describirlos?

Sí _____ No _____

En caso positivo, mencione cuantos procesos considera que tiene la UCI descritos.

Muchos _____

Termino Medio _____

Pocos _____

Ninguno _____

5. ¿Hay servicios definidos en la UCI?

Sí _____ No _____

En caso positivo, cuántos servicios considera que se encuentran definidos en la universidad.

Muchos _____

Termino Medio _____

Pocos _____

Ninguno _____

6. Para desarrollar una SOA es necesaria la integración de todas las fuerzas de la UCI, estudiantes, estudiantes vinculados a proyectos productivos, profesores y directivos. Como considera usted que se encuentre este aspecto en la universidad.

Anexo II

Especialistas

Especialistas	Categoría Docente	Cargos	Centro de trabajo
1	Adiestrado	Especialista en SOA	Facultad 3
2	Adiestrado	Especialista en SOA	Facultad 10
3	Asistente	Arquitecto principal de Sistema de la Intranet y especialista de la Dirección de Informatización.	Dirección de Informatización
4	Asistente	Jefe de grupo de plataforma de la Dirección de Informatización	Dirección de Informatización
5	Instructor Recién Graduado	Especialista de la Dirección de Informatización.	Dirección de Informatización
6	Instructor Recién Graduado	Especialista en SOA	UCI
7	Instructor Recién Graduado	Especialista en SOA	UCI

Anexo III

Cuestionario para especialistas.

Los aspectos a tener en cuenta por los expertos para realizar la evaluación del proceso se muestran a continuación:

Usted a ha sido seleccionado por su conocimiento en Arquitectura Orientada a Servicios (SOA), por sus años de experiencia y los resultados alcanzados en su labor profesional, como experto a evaluar los resultados teóricos de esta investigación.

1. ¿Considera usted que es importante que la UCI adopte una SOA?

___ Si ___ No

¿Por qué? _____

2. ¿Cuáles considera usted, son los 10 principales pasos que se deberían tener en cuenta al adoptar una SOA?

- I. _____
- II. _____
- III. _____
- IV. _____
- V. _____
- VI. _____
- VII. _____
- VIII. _____

IX. _____

x. _____

3. ¿Considera usted que la aplicación en la UCI de la guía propuesta, pueda tener efectividad en la adopción de SOA y cumpliría con su objetivo?

 Si No

¿Por qué? _____

4. ¿Cuáles serían los elementos que se opondrían a la correcta aplicación de la guía propuesta en la UCI y cuales estarían a favor?

5. Haga una breve valoración de los beneficios o dificultades que pudiera presentar la aplicación de la guía propuesta en la UCI.

6. Haga un comentario o aporte sobre la guía que es por usted objeto de evaluación. (El comentario es libre y debe reflejar algún elemento de interés que aporte elementos a la mejora del proceso).



Para finalizar, queremos expresarle que sus criterios y opiniones se manejarán de forma anónima, además le agradecemos por anticipado su valiosa colaboración y estamos seguras que sus sugerencias contribuirán a perfeccionar la guía propuesta.

Muchas gracias por su cooperación y le pedimos disculpas por las molestias ocasionadas.

GLOSARIO

A

Arquitectura de Software: La AS es, a grandes rasgos, una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones.

API (Application Programming Interface - Interfaz de Programación de Aplicaciones): es el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

B

BPM (Business Process Management): Es el entendimiento, gestión e innovación de procesos bajo estándares internacionales, alineados con la estrategia de negocio para asegurar la efectividad del proceso y crear valor a la cadena productiva de la empresa y su sector.

C

Calidad: Una forma de hacer las cosas en las que, fundamentalmente, predominan la preocupación por satisfacer al cliente y por mejorar, día a día, procesos y resultados. Una forma de gestión que introduce el concepto de mejora continua en cualquier organización y a todos los niveles de la misma, y que afecta a todas las personas y a todos los procesos.

CMMI: Integración del Modelo de Madurez de las Capacidades.

E

ESB (Enterprise Service Bus): 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, auditoría, entre otros.

I

Ingeniería de software: La Ingeniería de Software es una tecnología multicapa en la que, según Pressman, se pueden identificar: los métodos, el proceso y las herramientas.

M

Método Delphi: Es una técnica prospectiva para obtener información esencialmente cualitativa, pero relativamente precisa, acerca del futuro. Consiste básicamente en solicitar de forma sistemática las opiniones de un grupo de expertos, pero prescindiendo de la discusión abierta, lo que permite evitar los inconvenientes de ésta.

Middleware: Es un software de conectividad que ofrece un conjunto de servicios que hacen posible el funcionamiento de aplicaciones distribuidas sobre plataformas heterogéneas. Funciona como una capa de abstracción de software distribuida, que se sitúa entre las capas de aplicaciones y las capas inferiores (sistema operativo y red).

MSF (Microsoft Solution Framework): Es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

P

Políticas: Declaraciones o interpretaciones generales que guían el pensamiento durante la toma de decisiones; la esencia de las políticas es la existencia de cierto grado de discrecionalidad para guiar la toma de decisiones.

R

RUP (Rational Unified Process): En un proceso que de manera ordenada define tareas y quién cómo y cuándo el Equipo de desarrollo las hará, son un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Define un ciclo de vida iterativo, priorizando el uso de lenguajes de modelado, casos de uso y centrado en la arquitectura.

S

SOA (Arquitectura Orientada a Servicios): Es una estrategia en la cual las aplicaciones hacen uso de los servicios disponibles en la red. Mediante SOA se pueden desarrollar aplicaciones que utilicen servicios o que se comporten como un servicio para que otros accedan a ellas o ambos.

SOAP (Simple Object Access Protocol): Protocolo basado en intercambio de mensajes XML. La especificación permite realizar el intercambio a través de múltiples protocolos de transporte como por ejemplo, HTTP o SNMP.

U

UDDI: UDDI son las siglas del catálogo de negocios de Internet denominado "Universal Description, Discovery and Integration". UDDI es un registro público diseñado para almacenar de forma estructurada información sobre empresas y los servicios que estas ofrecen. A través de UDDI, se puede publicar y descubrir información de una empresa y de sus servicios

UCI: Universidad de las Ciencias Informáticas.

W

WSDL (Services Description Language): Un formato XML que se utiliza para describir servicios Web

X

XP (Extreme Programming): Es una de las metodologías de desarrollo de software más exitosas en la actualidad utilizada para proyectos de corto plazo y corto equipo. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

