

**PROPUESTA PARA LA INTEROPERABILIDAD DE PROCESOS DE NEGOCIO  
EN SISTEMAS DE INFORMACIÓN HOSPITALARIOS BAJO UNA  
ARQUITECTURA ORIENTADA A SERVICIOS**



**MONOGRAFÍA**

**Peter Emerson Pinchao Solís  
Nicolás Rubén Caballero Ortega**

**Monografía para optar al título de Ingeniero en Electrónica y Telecomunicaciones**

Director  
PhD. Diego Mauricio López

*Universidad del Cauca*

**Facultad de Ingeniería Electrónica y Telecomunicaciones  
Departamento de Telemática  
Grupo de Ingeniería Telemática-GIT  
Línea de Investigación en eSalud  
Popayán, Marzo de 2011**



## TABLA DE CONTENIDO

INTRODUCCIÓN.....	1
<b>1. CAPITULO I. MARCO TEÓRICO Y ESTADO DEL ARTE. ....</b>	<b>6</b>
1.1. CONCEPTOS GENERALES. ....	6
1.1.1. <i>Interoperabilidad de Sistemas de Información.</i> .....	6
1.1.2. <i>Administración de Procesos de Negocio (BPM).</i> .....	7
1.1.3. <i>Arquitecturas Orientadas a Servicios (SOA).</i> .....	7
1.1.4. <i>Bus de Servicios Empresariales (ESB).</i> .....	9
1.2. ESTÁNDARES Y ESPECIFICACIONES PARA SALUD. ....	11
1.2.1. <i>Estándar de Mensajería HL7.</i> .....	11
1.2.2. <i>Healthcare Services Specification Project (HSSP).</i> .....	13
1.3. PROYECTOS RELACIONADOS. ....	14
1.3.1. <i>Proyectos que Emplean Arquitecturas Orientadas a Servicios.</i> .....	14
1.3.1.1. Healthcare Services Specification Project-HSSP (HealthCare Services Specification Project-HSSP). ....	14
1.3.1.2. Distributed E-Healthcare System (Firat Kart, 2009). ....	14
1.3.2. <i>Proyectos que Emplean Buses de Servicios Empresariales (ESB) y el Estándar de Mensajería HL7.</i> .	14
1.3.2.1. ePOC PDA. ....	15
1.3.2.2. SerAPI (BLOBEL). ....	15
1.3.3. <i>Proyectos que Emplean el Estándar de Interoperabilidad HL7.</i> .....	15
1.3.3.1. National Health Information Infrastructure Program (United Nations Economic and Social Commission for Asia and the Pacific, 2007) (BLOBEL). ....	15
1.3.3.2. National Programme for IT (BLOBEL). ....	15
1.3.3.3. Shareable EHR (BLOBEL).....	16
1.3.4. <i>Iniciativas Nacionales de Interoperabilidad.</i> .....	16
1.3.4.1. GEL-XML (Gobierno en Línea, 2008). ....	16
1.3.4.2. SISPRO. ....	16
1.3.5. <i>Otros Trabajos</i> .....	16
1.3.5.1. Intel SOA Expressway for Healthcare (Intel, 2010).....	17
1.4. SOA PARA SISTEMAS DE INFORMACIÓN HOSPITALARIOS. ....	17
1.4.1. <i>Orientación de la Arquitectura Hacia los Procesos de Negocio de las Entidades de Salud.</i> .....	17
1.4.1.1. Paso 1. Análisis de la Arquitectura Empresarial.....	17
1.4.1.2. Paso 2. Estado Actual del Entorno. ....	18
1.4.1.3. Paso 3. Identificación de Servicios.....	18
1.4.1.4. Paso 4. Definición de Estados Futuros. ....	21
1.4.2. <i>Construcción de la Arquitectura.</i> .....	21
1.4.2.1. Paso 5. Especificación de Arquitectura y Servicios. ....	21
1.4.2.2. Paso 6. Construcción de un Plan de Transición. ....	22
1.4.2.3. Paso 7. Realización de SOA. ....	22
1.4.2.4. Paso 8. Despliegue y Sostenimiento. ....	22
1.5. CONCLUSIONES .....	23
<b>2. CAPITULO II. PROPUESTA DE ARQUITECTURA SOA. ....</b>	<b>25</b>
2.1. REQUERIMIENTOS DE LA ARQUITECTURA PARA DAR SOLUCIÓN A LA PROBLEMÁTICA EN EL HSLV. ....	25
2.1.1. <i>Interface de Interoperabilidad HL7.</i> .....	26



2.2.	DEFINICIÓN DE LA METODOLOGÍA PARA LA CONSTRUCCIÓN DE LA ARQUITECTURA. ....	27
2.2.1.	<i>Modelo de Referencia RM-ODP.</i> .....	27
2.2.2.	<i>El Marco de Desarrollo para sistema de información en salud.</i> .....	28
2.2.2.1.	Punto de vista empresarial. ....	29
2.2.2.2.	Punto de vista de Información. ....	29
2.2.2.3.	Punto de vista computacional. ....	30
2.2.3.	<i>Metodología de Referencia SOA.</i> .....	30
2.2.3.1.	Fase 1. Análisis Orientado a Servicios. ....	31
2.2.3.2.	Fase 2. Diseño Orientado a Servicios. ....	31
2.2.3.3.	Fase 3. Desarrollo de Servicios. ....	32
2.2.3.4.	Fase 4. Prueba de Servicios. ....	32
2.2.3.5.	Fase 5. Despliegue de Servicios. ....	32
2.2.3.6.	Fase 6. Administración de Servicios. ....	33
2.2.4.	<i>Metodología SOMA.</i> .....	33
2.2.4.1.	Identificación de Servicios .....	34
2.2.4.2.	Especificación.....	34
2.2.4.3.	Realización.....	35
2.2.5.	<i>Discusión y Selección de la Metodología.</i> .....	35
2.3.	VISTA EMPRESARIAL DE LA ARQUITECTURA. ....	36
2.3.1.	<i>Vista Computacional de la Arquitectura.</i> .....	38
2.3.2.	<i>Descripción de Bloques y Componentes de Arquitectura Computacional.</i> .....	39
2.4.	ARQUITECTURA COMPUTACIONAL (MODELO DINÁMICO). ....	43
2.5.	JUSTIFICACIÓN DE LA ARQUITECTURA, BLOQUES Y COMPONENTES. ....	45
2.6.	VISTA DE LA INGENIERÍA DE LA ARQUITECTURA. ....	47
2.7.	VISTA TECNOLÓGICA DEL SISTEMA.....	49
2.8.	CONCLUSIONES. ....	50
<b>3.</b>	<b>CAPÍTULO III. IMPLEMENTACIÓN DE LA ARQUITECTURA SOA.</b> .....	<b>51</b>
3.1.	OPCIONES TECNOLÓGICAS PARA SOA. ....	51
3.1.1.	<i>Nivel de Transporte.</i> .....	51
3.1.2.	<i>Nivel de Protocolos de Comunicación de Servicios.</i> .....	53
3.1.3.	<i>Nivel de Descripción de Servicios</i> .....	53
3.1.4.	<i>Nivel de Servicio.</i> .....	54
3.1.5.	<i>Nivel de Procesos de Negocio.</i> .....	54
3.1.6.	<i>Nivel de Registro de Servicios.</i> .....	55
3.1.7.	<i>Nivel de Políticas.</i> .....	55
3.1.8.	<i>Nivel de Seguridad.</i> .....	55
3.2.	BUS DE SERVICIOS EMPRESARIALES – ESB.....	56
3.2.1.	<i>Criterios de Selección.</i> .....	56
3.2.1.1.	Enrutamiento de Mensajes y Servicios. ....	56
3.2.1.2.	Herramienta de Administración. ....	56
3.2.1.3.	Soporte de Protocolos de Transporte Abiertos. ....	56
3.2.1.4.	Transformación de los Datos.....	57
3.2.1.5.	Manejo Sincrónico y Asincrónico de los mensajes.....	57
3.2.1.6.	Basado en Estándares.....	57
3.2.1.7.	Seguridad.....	57
3.2.1.8.	Definición de Servicios.....	57
3.2.1.9.	“Extensibilidad” .....	57



3.2.1.10.	Orquestación de Servicios.....	58
3.2.1.11.	Calidad de Servicio. ....	58
3.2.1.12.	Soporte Técnico. ....	58
3.2.1.13.	Entorno de Desarrollo Integrado (IDE).....	58
3.2.1.14.	Versión Estable.....	58
3.2.1.15.	Soporte de Mensajes HL7. ....	58
3.3.	COMPARACIÓN DE ESB BASADOS EN SOFTWARE LIBRE. ....	58
3.4.	SELECCIÓN DEL ESB.....	60
3.5.	OPCIONES TECNOLÓGICAS PARA HL7.....	61
3.5.1.	<i>HL7 Building Component</i> . ....	61
3.5.2.	<i>Motor de Interfaces HL7 (Mirth)</i> . ....	62
3.5.3.	<i>ChainBulder</i> . ....	63
3.6.	CONCLUSIONES .....	63
<b>4.</b>	<b>CAPITULO IV. EVALUACIÓN DE LA ARQUITECTURA SOLUCIÓN.....</b>	<b>65</b>
4.1.	DESCRIPCIÓN DE LOS PROCESOS DE NEGOCIO.....	65
4.2.	ANÁLISIS DE SISTEMAS LEGADOS.....	67
4.3.	APLICACIÓN DE HL7 EN LA ARQUITECTURA SOLUCIÓN.....	68
4.4.	IMPLEMENTACIÓN EMPLEANDO HL7 BUILDING COMPONENT. ....	71
4.5.	ESENAIO DE INTEROPERABILIDAD DE SERVICIOS. ....	72
4.6.	CRITERIOS DE EVALUACIÓN (EVALUACIÓN DE LA ARQUITECTURA).....	73
4.6.1.	<i>Objetivos de Evaluación</i> . ....	73
4.6.1.1.	Evaluación Comparativa. ....	74
4.6.2.	<i>Evaluación de Escalabilidad</i> .....	74
4.6.2.1.	Métricas de evaluación.....	74
4.6.2.2.	Metodología Evaluación Escalabilidad.....	75
4.6.3.	<i>Pruebas de Flexibilidad</i> .....	75
4.6.3.1.	Paso a la evolución. ....	76
4.6.3.2.	Métricas de costo de evolución del software.....	76
4.7.	ESCENARIOS DE EVALUACIÓN. ....	78
4.7.1.	<i>Escenarios de Evaluación de la Escalabilidad</i> . ....	78
4.7.1.1.	Pruebas de carga.....	79
4.7.1.2.	Pruebas de stress. ....	80
4.7.2.	<i>Escenarios de Evaluación de la Flexibilidad</i> . ....	81
4.8.	EVALUACIÓN DE LA ARQUITECTURA. ....	82
4.8.1.	<i>Pruebas del Escalabilidad</i> . ....	82
4.8.1.1.	Establecimiento del punto de comparación (Benchmarking).....	82
4.8.1.2.	Pruebas de Carga. ....	83
4.8.1.3.	Pruebas de stress. ....	84
4.8.2.	<i>Prueba de Flexibilidad</i> . ....	85
4.9.	ANÁLISIS DE RESULTADOS. ....	87
4.9.1.	<i>Análisis de los Resultados de las Pruebas de Escalabilidad</i> . ....	87
4.9.1.1.	Análisis en las Pruebas de Carga.....	87
4.9.1.2.	Análisis en las Pruebas de Strees.....	89
4.9.2.	<i>Análisis de los Resultados de la prueba de Flexibilidad</i> . ....	90
4.10.	CONCLUSIONES .....	91
<b>5.</b>	<b>CAPITULO V. CONCLUSIONES Y RECOMENDACIONES.....</b>	<b>92</b>

***Propuesta para la Interoperabilidad de Procesos de Negocio en Sistemas de Información Hospitalarios Bajo una Arquitectura Orientada a Servicios***

---



5.1. CONCLUSIONES. .... 92

5.2. RECOMENDACIONES Y TRABAJOS FUTUROS. .... 93

**REFERENCIAS..... 95**



## **LISTA DE FIGURAS**

FIGURA 1. ENFOQUE PUNTO A PUNTO. ....	10
FIGURA 2. ENFOQUE ESB.....	10
FIGURA 3. USANDO MENAJES PARA EL EVENTO DE TRASFERENCIA DE MENAJES (HEITMANN, Y OTROS, 1999). ....	12
FIGURA 4. TAREAS, ARTEFACTOS Y ROLES EN EL PROCESO DE DESARROLLO DE LA ARQUITECTURA DEL SISTEMA. ADAPTADO DE (AGUILAR, Y OTROS, 2009) .....	29
FIGURA 5. VISTA LÓGICA DE LA ARQUITECTURA DE REFERENCIA SOA. TOMADA DE (ERL, 2005) .....	30
FIGURA 6. ETAPAS DEFINIDAS POR LA METODOLOGÍA SOMA. TOMADA DE (ALI ARSANJANI, 2004) .....	34
FIGURA 7. VISTA COMPUTACIONAL DE LA ARQUITECTURA PROPUESTA SOA-SIS PARA EL HSLV. ....	38
FIGURA 8. MÓDULOS DE LA ARQUITECTURA SOA-SIS. ....	40
FIGURA 9. COMPONENTES DEL MÓDULO DE REPOSITORIO DE DATOS. ....	41
FIGURA 10. MÓDULO DE BODEGA DE DATOS. ....	41
FIGURA 11. MÓDULO DE INFRAESTRUCTURA SOA-SIS. ....	42
FIGURA 12. COMUNICACION DE COMPONENTES.....	44
FIGURA 13. CLASIFICACIÓN DE SERVICIOS. ....	46
FIGURA 14. VISTA DE INGENIERIA DE LA ARQUITECTURA SOLUCION. ....	49
FIGURA 15. VISTA TECNOLÓGICA. MODELO DE DESPLIEGUE.....	50
FIGURA 16. TECNOLOGÍAS PARA SOA. ....	51
FIGURA 17. REPRESENTACIÓN DEL LA COMPOSICION DEL PROCESO DE AUDIOTORIA. ....	66
FIGURA 18. REPRESENTACIÓN DEL A COMPOSICION DEL PROCESO TELE-CONSULTA. ....	66
FIGURA 19. JERARQUIA DE COMPOSICION DEL SERVICIO DE NEGOCIO DE AUDITORIA. ....	73
FIGURA 20. ESCENARIO DE PRUEBA DE ESCALABILIDAD. ....	80
FIGURA 21. LATENCIA EN LAS PRUEBAS DE CARGA.....	88
FIGURA 22. THROUGHOUT EN LAS PRUEBAS DE CARGA.....	88
FIGURA 23. LATENCIA EN LAS PRUEBAS DE CARGA EN AMEDICAS .....	89
FIGURA 24. THROUGHOUT EN LAS PRUEBAS DE CARGA EN AMEDICAS. ....	89
FIGURA 25. PRUEBAS DE STRESS. ....	90



## **LISTA DE TABLAS**

TABLA 1. ESPECIFICACIONES DEL PROYECTO HSSP (HSSP, 2009).....	13
TABLA 2. EJEMPLO DE MAPA DE FUNCIONES DE NEGOCIO. ....	20
TABLA 3. VISTA EMPRESARIAL DEL HSLV / MAPA DE FUNCIONES DEL NEGOCIO.....	37
TABLA 4. CRITERIOS DE SELECCIÓN DEL ESB. ....	59
TABLA 5 SERVICIOS .....	65
TABLA 6. TABLA ANALISIS DE SISTEMAS LEGADOS .....	67
TABLA 7. PRUEBAS DE CARGA PARA ESTABLECER EL BENCHMARKING. ....	79
TABLA 8. PRUEBAS DE CARGA. ....	80
TABLA 9. PRUEBAS DE STRESS.....	81
TABLA 10. RESULTADO DEL BENCHMARKING. ....	82
TABLA 11. NIVEL DE ACEPTACIÓN DE SERVICIOS (BENCHMARKING) .....	83
TABLA 12. PRUEBAS DE CARGA SOBRE LA ARQUITECTURA. ....	83
TABLA 13. PRUEBAS DE CARGA AMEDICAS. ....	83
TABLA 14. PRUEBA DE STRES SOBRE LA ARQUITECTURA UN NODO EN EL CLUSTER. ....	84
TABLA 15. PRUEBA DE STRESS SOBRE LA ARQUITECTURA CON 2 NODOS EN EL CLUSTER. ....	84
TABLA 16. PRUEBA DE STRESS SOBRE LA ARQUITECTURA CON 3 NODOS EN EL CLUSTER. ....	85
TABLA 17. LIMITES DE LA ARQUITECTURA. ....	85
TABLA 18. MEDIDAS DEL COSTO DE EVOLUCIÓN SIN REMAPEAR LAS ENTIDADES USANDO JPA.....	86
TABLA 19. MEDIDAS DEL COSTO DE EVOLUCIÓN REMAPEANDO LAS ENTIDADES USANDO JPA .....	86
TABLA 20. NIVELES DE CARGA EN LA PRUEBAS DE CARGA. ....	87



## **INTRODUCCIÓN.**

### **Escenario Actual.**

En la actualidad los Sistemas de Información de Salud (SIS) en Colombia experimentan continuamente importantes reformas, orientadas en gran medida a garantizar la prestación de servicios a mejorar la eficiencia y efectividad de los procesos. Un ejemplo de ello es la ley 715 del 21 de Diciembre de 2001, donde se establecen las políticas de descentralización (en Salud y Educación) a nivel municipal y departamental que involucran la creación de nuevos procesos administrativos, que se suman a las responsabilidades ya otorgadas a las entidades territoriales en salud en lo referente al aseguramiento, la salud pública y el control a entidades y contratos<sup>1</sup> (Colombia, 2001). Estas reformas demandan gran esfuerzo y responsabilidad por parte de las entidades territoriales en salud, constituyéndose en un imperativo la automatización de todos sus procesos a través de un sistema integrado institucional y territorial, el cual debe garantizar la operación, gestión y control de cada uno de sus componentes, tanto al interior como entre cada instancia territorial.

Esta problemática no es ajena a las Instituciones Prestadoras de Servicios de Salud (IPS), y constituyen un desafío importante, especialmente en aquellas que hacen uso de las Tecnologías de Información y Comunicación (TIC) que emplean para soportar sus procesos de negocio. Los Sistemas de Información en Salud (SIS) tienen la finalidad de responder a estas necesidades, logrando en cierta parte soportar estos procesos, para lograr la gestión de información que va desde historias clínicas, sistemas de laboratorios, radiología e imágenes médicas, gestión de citas médicas, farmacología, información terapéutica y facturación de servicios así como también datos básicos de conocimiento en salud importante como información poblacional, epidemiológica, información relacionada con factores medioambientales, y últimamente indicadores de salud que permiten obtener medidas de eficiencia, como resultado del tratamiento de información demográfica (Cámpoli, 2005). La variedad de SIS existentes da cuenta del volumen de la información que manejan los sistemas de salud y la importancia que tiene emplear los avances que ofrecen las ciencias de la informática para su tratamiento adecuado.

Una de las decisiones más importantes que deben tomar las entidades de salud, es la de adquirir o desarrollar una solución integral que soporte todos sus procesos de negocio, normalmente lograda mediante la adquisición de soluciones informáticas de un mismo proveedor a fin de garantizar interoperabilidad<sup>2</sup> entre procesos, o hacer una integración de los componentes de múltiples proveedores que atienden procesos de negocio específicos. La primera alternativa es una solución adecuada si la solución informática es lo suficientemente robusta y altamente configurable; no obstante estos sistemas sofisticados tienen un alto costo de adquisición y de soporte, que en muchos casos lo hacen inaccesible para algunas instituciones. La segunda alternativa es adquirir sistemas de información existentes en el mercado para soportar determinados procesos de negocio que aunque no permiten interoperabilidad, dificultando la

---

<sup>1</sup> Relacionado con las entidades Prestadoras de Planes y Beneficios (EPS, ARS, Accidentes entre otros).

<sup>2</sup>El término Interoperabilidad, establecido en el Glosario Estándar de la IEEE de Terminología para la Ingeniería del Software.





ejecución otros procesos y ocasionando redundancia de datos; es una alternativa menos costosa y por ende, más accesible que la adquisición de una única solución.

### **Planteamiento del Problema.**

El Hospital Susana López de Valencia (HSLV) de la ciudad de Popayán no es ajeno a estos problemas y aunque tienen la mayor parte de sus procesos de negocio soportados con sistemas de información de un proveedor principal, la incompatibilidad de las versiones de los paquetes tecnológicos que componen estos sistemas, no los hacen completamente interoperables; a esto se suma el hecho de que cuentan con otros sistemas de información de otros proveedores. Estos sistemas ofrecen información en distintos formatos y operan bajo diferentes plataformas tecnológicas lo cual genera una barrera para la interoperabilidad.

El anterior escenario muestra la dificultad de intercambiar información de forma efectiva entre sistemas, así como responder ágilmente a los cambios en las normativas del ministerio de la protección. De esta manera, se puede decir que los Sistemas de Información del HSLV están en un Entorno Heterogéneo<sup>3</sup>, lo cual dificulta el proceso de Administración de Procesos de Negocio en la institución apoyados por las Tecnologías de la Información (TI). Sin embargo, es importante tener en cuenta las grandes inversiones en la adquisición de estos sistemas, que aunque no trabajan muy bien en conjunto, atienden procesos de negocio específicos de una forma aceptable.

Del análisis de estos problemas se puede identificar la necesidad de contar con una solución que permita garantizar la interoperabilidad entre los distintos servicios, así como la reutilización de los mismos con el fin de ampliar las capacidades de cada sistema. Esto debe lograrse con una notable reducción de costos y tiempo de desarrollo, obteniendo sistemas acordes con las necesidades cambiantes del dominio de la aplicación. Es necesario reutilizar estas aplicaciones, de tal manera que se habiliten los Sistemas Legados<sup>4</sup> en una nueva arquitectura que provea un nivel de interoperabilidad suficiente para que puedan trabajar como un solo Sistema de Información.

Se han estudiado alternativas para afrontar estos problemas, y en el caso particular del HSLV el principal proveedor de sistemas de información ofrece una actualización y adición de sistemas, que permitiría mejorar la administración de la mayor parte de los procesos de negocio. El principal inconveniente de esta alternativa son los altos costos. Sin embargo, no es la única barrera que se presenta, ya que uno de los requerimientos impuestos por el proveedor, es que la empresa debe prescindir de los sistemas de información actuales mínimo por 3 días, para lograr los efectos de migración a nuevas estructuras de almacenamiento; a esto se suman los procesos de capacitación necesarios. Esto hace que sea más difícil de aceptar la propuesta, ya que el flujo de información del centro de servicio en salud es alto y no es fácil considerar la implementación de un sistema contingente para poder prescindir de los sistemas de información (además de la redefinición de funciones y procedimiento durante la adaptación), que generarían costos adicionales en carga laboral.

---

<sup>3</sup> Entorno Heterogéneo: Sistema de Computación compuesto por subsistemas soportados por distintas tecnologías, protocolos y herramientas.

<sup>4</sup> Sistemas Legados: Sistemas Informáticos activos y existentes en un entorno de computación.



Lo anterior exige que la solución a implementar sea altamente escalable, de tal forma que los futuros Sistemas de Información puedan ser incorporados dentro de la arquitectura empresarial del HSLV. Las Entidades de Salud desean migrar sus sistemas hacia nuevos paradigmas tecnológicos que atiendan de una mejor manera sus necesidades, sin embargo, no desean prescindir totalmente de los Sistemas actualmente en uso. Es por esta razón que organizaciones como HL7 (Health Level Seven) (HL7, 2007) y OMG (Object Management Group) (OMG, 2010) promueven el establecimiento de estándares para la Industria de la Salud que permitan la interoperabilidad entre sistemas de información en ambientes heterogéneos, así como el aprovechamiento de las habilidades en la utilización de Modelos de Arquitecturas como SOA. Estas organizaciones hacen grandes esfuerzos en la Identificación y documentación de especificaciones de Servicios y funcionalidades.

La reducción de costos y la posibilidad de habilitar a los departamentos de sistemas (o tecnología, según la estructura organizacional de la entidad) de las IPS para que puedan adaptar, desarrollar y reconfigurar los procesos de negocio, hace que el uso de software de libre distribución sea una buena alternativa. Las comunidades de software libre son bastante dinámicas y ofrecen soporte ya sea de forma directa o indirecta para el uso y apropiación de las tecnologías ofrecidas.

### **Solución Propuesta.**

Teniendo en cuenta lo descrito anteriormente, este trabajo de grado propone una Arquitectura Orientada a Servicios que permita la interoperabilidad de los sistemas de información que actualmente operan en el HSLV (sistemas heredados) y la integración de nuevos sistemas de información dentro de una arquitectura flexible y escalable, que responda a los continuos cambios funcionales al interior de la organización. Esta arquitectura emplea estándares abiertos, así como tecnologías libres que permitan una solución de bajo costo, accesible para pequeñas y medianas empresas.

Cabe destacar, que se utilizará un Bus de Servicios Empresariales (ESB) como elemento principal de la arquitectura propuesta, con la finalidad de habilitar las virtudes que ofrece SOA, las cuales son descritas en este proyecto.

### **Contenidos.**

Esta monografía consta de cuatro capítulos orientados a responder al objetivo general: “construir una Arquitectura Orientada a Servicios que soporte la interoperabilidad de los sistemas de información actual, así como los nuevos sistemas de información del Hospital Susana López de Valencia utilizando herramientas basadas en software libre”.

Capítulo I. Incluye una descripción de los estándares y protocolos empleados en la construcción de la Arquitectura Orientada a Servicios para el sector Salud. También describe los proyectos en el ámbito de interoperabilidad y SOA para Sistemas de Información Hospitalarios (SIH). Finalmente, expone las recomendaciones propuestas por organismos de estandarización para desplegar una SOA en el ámbito Salud.

Capítulo II. Describe las necesidades que orientan el diseño de la arquitectura solución, así como las metodologías de referencia existentes para el diseño y modelado de una arquitectura orientada



a servicios, posibilitando un espacio para la discusión y selección de una de ellas. Detalla la Arquitectura Orientada a Servicios propuesta (descrita en el primer objetivo específico) que responde a los requerimientos de interoperabilidad de Procesos de Negocio. Finalmente describe los componentes que constituyen la Arquitectura.

Capítulo III. Describe las tecnologías empleadas para la implementación de la arquitectura SOA y la justificación su empleo de la arquitectura solución. Expone los criterios de selección de tecnologías basadas en software de libre distribución que permitan la implementación de la Arquitectura Orientada a Servicios (en el cumplimiento del segundo objetivo específico) que den soporte a la arquitectura propuesta en el Capítulo II. Se identifican los requerimientos de Interoperabilidad específicos de los sistemas existentes y futuros del HSLV. Finalmente presenta una descripción detallada de la Arquitectura y los sistemas que la componen.

Capítulo IV. Presenta los Procesos de Negocio de Hospital Susana Lopez de Valencia a Interoperar seleccionados. Describe el Análisis Orientado a Servicios, Modelado de Servicios, Modelado de Procesos de Negocio, Diseño de Servicios y Diseño de Procesos de Negocios de acuerdo a la Metodología adoptada en el capítulo II. Se establecen los criterios de evaluación de la arquitectura propuesta. Se hace una presentación de los escenarios de evaluación. Finalmente, presenta la evaluación (alcanzando el tercer objetivo específico) de la arquitectura y desarrolla una discusión basada en los resultados obtenidos.

#### **Aportes y Contribuciones.**

Este trabajo de grado ofrece una base conceptual de la utilización de SOA para SIS, exponiendo las principales herramientas, tecnologías y estándares que puedan emplearse en este ámbito.

Ofrece una contribución en la utilización de los Buses de Servicios Empresariales (ESB) para la construcción de SOA-SIS.

Este trabajo hace una implementación del Estándar HL7 v3 para soportar la interoperabilidad en SIS.

Contribuye a la investigación y desarrollo tecnológico mediante la utilización de múltiples herramientas informáticas de libre distribución que establecen una arquitectura coherente basada completamente en estas tecnologías abiertas, haciendo una gran contribución a la comunidad de software libre en la construcción de arquitecturas y en la implementación de las mismas, empleado una metodología ampliamente aceptada.

- La arquitectura propuesta, por estar basada en tecnologías abiertas, la hace bastante accesible para pequeñas y medianas empresas. De esta manera, la constituye en una alternativa a las empresas que desean redefinir sus arquitecturas empresariales y emplear SOA.
- Contribución a la línea de investigación eSalud. Concretamente, aportará conocimiento sobre los requerimientos funcionales y no funcionales para la prestación de los Servicios de Información Hospitalarios bajo el paradigma Orientado a Servicios y Administración de Procesos de Negocios (Business Process Management, BPM), haciendo un aporte en la



consolidación de la línea de e-Salud y la participación de la Universidad del Cauca en la fundación HL7 Colombia.



## **1. CAPITULO I. MARCO TEÓRICO Y ESTADO DEL ARTE.**

En este capítulo se escriben los conceptos generales necesarios para la correcta comprensión de este trabajo así como las organizaciones que regulan y promueven estándares para SIS. Inicia con una descripción de los conceptos generales, estándares y protocolos empleados en la construcción de la Arquitectura Orientada a Servicios para el sector Salud. También describe los proyectos relacionados con este trabajo y sus contribuciones. Estos proyectos son clasificados de acuerdo a las tecnologías utilizadas, además propone una discusión sobre las limitantes de estos proyectos, y cómo contribuyen como base para el desarrollo de la propuesta que este trabajo de grado propone alcanzar. Finalmente presenta las recomendaciones que aportan las organizaciones de estandarización que deben emplearse en la construcción de una SOA para SIS.

### **1.1. CONCEPTOS GENERALES.**

#### **1.1.1. INTEROPERABILIDAD DE SISTEMAS DE INFORMACIÓN.**

La interoperabilidad es la condición según la cual sistemas heterogéneos pueden intercambiar procesos y/o datos; definiéndose como *“la capacidad de los sistemas de información, y por ende de los procedimientos a los que estos dan soporte, de compartir datos y posibilidad de intercambio de información y conocimiento entre ellos”* (García, 2009). Existen definiciones que sólo contemplan la faceta tecnológica de la interoperabilidad, sin embargo, en el contexto de los procesos de negocio se puede ir más allá: para que una administración pueda considerarse interoperable debe estar activamente comprometida con la interminable tarea de gestionar sus sistemas, procedimientos, organización y cultura, de manera que maximice las oportunidades de intercambio, reutilización de la información y funcionalidades, tanto interna como externamente. Esto hace clasificar la interoperabilidad en tres aspectos fundamentales a saber:

- **Interoperabilidad Técnica:** está enmarcada en la interoperabilidad de sistemas y servicios de tecnologías de la información (García, 2009). La interoperabilidad técnica afronta los problemas de definición de interfaces para la interacción, la interconexión de componentes tecnológicos, integración de datos y/o servicios, presentación de información, definición y despliegue de mecanismos de accesibilidad y seguridad.

Las Arquitecturas Orientadas a Servicios son un componente fuerte que provee mecanismos para lograr la Interoperabilidad Técnica.

- **Interoperabilidad Semántica:** enfocada en la interoperabilidad de la información a intercambiar. Está orientado a brindar mecanismos que permitan que la información sea interpretable por diversos sistemas en forma automática (por aplicaciones y servicios). Esto implica que los sistemas deben compartir modelos de datos, los cuales están compuestos por un conjunto de definiciones de los diversos tipos de datos, actores y objetos, de manera que estén identificados inequívocamente (Modelo Conceptual);



además, se deben establecer las interrelaciones de los datos (Modelo Lógico) y las reglas y convenciones que son necesarias para su descripción e intercambio (Modelo Físico).

Iniciativas maduras como HL7, que en la actualidad es un estándar para el intercambio electrónico de la información en Salud, proveen especificaciones para abordar los aspectos enmarcados como Interoperabilidad Semántica en SIH.

- **Interoperabilidad Organizativa:** es la clase de interoperabilidad que define las capacidades de las entidades y la colaboración entre procesos que permite alcanzar los logros que corresponden al servicio que prestan. Implica un conocimiento y reconocimiento de los procedimientos que definen los procesos de negocio y dan soporte a los datos a intercambiar (García, 2009).

La Administración de Procesos de Negocio es un mecanismo que contribuye al establecimiento de la interoperabilidad organizativa.

### **1.1.2. ADMINISTRACIÓN DE PROCESOS DE NEGOCIO (BPM).**

Es una metodología empresarial que tiene como objetivo principal mejorar la eficiencia de los procesos de negocio mediante la gestión sistemática de los mismos, permitiendo a las organizaciones, optimizar el desempeño de sus negocios a través del descubrimiento, modelado, documentación, automatización y mejora continua de los procesos de negocio incrementando la eficiencia y reduciendo costos (Chamberland, y otros, 2010).

La OMG es uno de los principales organismos de estandarización en informática que entiende la importancia de vincular procesos de gestión de la calidad en las entidades de salud; en orden a esto, ofrecen certificaciones al personal administrativo así como de los departamentos de gestión tecnológica, de forma que les acrediten las habilidades para el establecimiento de BPM en las empresas de salud (Object Management Group - OMG, 2010), a través del programa OMG Certified Expert in BPM (OCEB™).

### **1.1.3. ARQUITECTURAS ORIENTADAS A SERVICIOS (SOA).**

Una Arquitectura Orientada a Servicios es aquella arquitectura de software donde los componentes básicos que componen su sistema se relacionan intrínseca y extrínsecamente, donde los componentes básicos son servicios (IEEE, 2000). Las aplicaciones SOA son creadas como una composición de varios servicios, los cuales pueden ser compartidos entre múltiples aplicaciones (Microsoft Corporation, 2006). Define aspectos como la localización de servicios y los medios de comunicación entre ellos, así como su papel dentro del sistema. SOA no es una tecnología, es más bien un concepto de arquitectura para el desarrollo de sistemas que integran aplicaciones que son accesibles a través de la red, independientemente del lenguaje o plataforma tecnológica que las soportan. (Microsoft Corporation, 2006).

SOA es implementado tradicionalmente a través de Servicios Web (WS). Estos son un conjunto de aplicaciones o tecnologías con la capacidad de interoperar en la web e intercambiar datos entre sí con el objetivo de ofrecer nuevos servicios.



Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos mediante el Protocolo Simple de Acceso a Objetos (Simple Object Access Protocol, SOAP) (W3C, 2010). De esta manera, los WS proporcionan un mecanismo de comunicación estándar entre diferentes aplicaciones para interactuar entre sí y presentar información dinámica al usuario final.

SOA no es solo un medio para generar más componentes de TI; la potencia de SOA radica en su habilidad de expresar capacidades técnicas en términos de negocio y de permitir a las empresas volver a combinar sus componentes para crear nuevas soluciones (Erl, 2005). A continuación son descritos algunos conceptos inherentes a estas arquitecturas que deben considerarse en el despliegue de los procesos de negocio:

- **Granularidad:** describe el tamaño de los componentes que constituyen un sistema. SOA prefiere los componentes de mayor tamaño (de grano grueso) conocidos como servicios de negocio. Generalmente estos son construidos a partir de otros servicios técnicos más pequeños (de grano fino) que ya existen. Esto es importante porque las piezas más grandes favorecen a que el personal de la empresa comprenda, reutilice y maneje los servicios de la Arquitectura (Matsumara, y otros, 2009).
- **Interfaz vs implementación:** diferencia entre lo que hace un servicio de los detalles técnicos de su implementación. Esto es importante porque así, el usuario del negocio centra su atención en lo que hace el servicio y no en su funcionamiento.
- **Los contratos de servicio:** definen las obligaciones entre el proveedor y el consumidor del servicio. Pueden contemplar expectativas sobre el servicio tales como disponibilidad, fiabilidad, indicadores clave de rendimiento, costes y asistencia (Matsumara, y otros, 2009).
- **Acoplamiento:** es el modo de diseñar servicios más flexibles y menos dependientes unos de otros. Esto facilita el ensamblaje de los servicios y su recombinación en nuevos contextos.

El comportamiento de un servicio compuesto por otros servicios requiere de un mecanismo para la definición de los procesos que modele correctamente el proceso de ejecución de los servicios, en orden a esto existe WS-BPEL (Service - Business Process Execution Language), que es un lenguaje estándar basado en XML que permite presentar el comportamiento de los procesos de negocio orquestando los Servicios Web, es decir describiendo el flujo trabajo de ejecución de los servicios miembro (Oasis, 2007). Esto permite la implementación de la lógica de negocio; y, a través de WS-BPEL, permite la creación de nuevos servicios.

Un proceso de BPEL está compuesto de los siguientes elementos:

- **Variables:** permiten almacenar datos o el estado de un mensaje determinado.
- **Partner Links:** representan un intercambio de mensajes bidireccional entre dos partes.
- **Actividades Básicas:** describen los pasos elementales del comportamiento de un proceso. Estas son: invoke, receive y wait.
- **Actividades Estructuradas:** describen el orden de ejecución en que una colección de actividades es ejecutado. Estas son: for, switch, pick, while, throw.

Las Arquitecturas Orientadas a Servicios apoyadas en estas tecnologías permiten la reutilización de sistemas legados, los cuales representan una inversión por parte de las empresas. Estos



sistemas conforman el núcleo de todo portafolio de negocios, ya que soportan cada uno de los procesos de negocio de una empresa. Aun cuando estos sistemas deban ser reemplazados, no implica que deba ser así, sino más bien, hacer buen uso de ellos y aprovechar la información que contienen y los procesos que atienden, para incorporarlos dentro de una nueva Arquitectura Empresarial coherente.

#### **1.1.4. BUS DE SERVICIOS EMPRESARIALES (ESB).**

Uno de los aspectos críticos en el diseño de una SOA es desplegar servicios basados en procesos de negocio de forma eficiente y estructurada; para esto se puede considerar tres enfoques: una Integración de Aplicaciones Empresariales (Enterprise Application Integrator, EAI), una arquitectura punto a punto y la basada en ESB. Los EAI son considerados como enrutadores de mensajes, simplifican el intercambio de estos y centralizan la ejecución de procedimientos de software propietario, mediante el uso de adaptadores, y que adicionalmente implementa una SOA permitiendo incorporar composición de servicios con WS-BPEL, como el producto ofrecido por Intel (Intel, 2010).

El Bus de Servicios Empresarial no es un producto, sino más bien, un conjunto de herramientas que establece un bus de mensajería de tipo empresarial que combina infraestructura de mensajería con la transformación de mensajes y el enrutamiento basado en contenido en una capa de la lógica de integración entre consumidores y proveedores de servicios. El objetivo principal del ESB es virtualizar los recursos de la empresa, permitiendo que la lógica de negocio sea desarrollada y gestionada de forma independiente de la infraestructura. Los recursos en el ESB son modelados como servicios que ofrecen una o más operaciones de negocios (Endrei, y otros, 2004).

La Figura 1 muestra el enfoque de arquitectura punto a punto. Este enfoque o modelo presenta el problema exponencial de conexiones que generan este tipo de integración, donde el número de conexiones crece exponencialmente por cada sistema que es agregado. Este enfoque dificulta la administración de servicios debido al alto acoplamiento que presenta, dificultando el despliegue de los contratos del servicio.



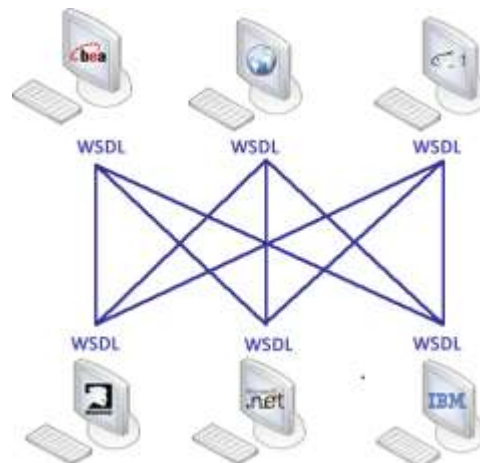


Figura 1. Enfoque punto a punto.

La Figura 2 expone el enfoque de arquitectura utilizando el Bus de Servicios empresarial, donde cada aplicación es “conectada” solo una vez a una infraestructura troncal común: el Bus, lo cual reduce al mínimo las conexiones y proporciona una ubicación centralizada de los contratos de servicio para la administración y gestión de los sistemas que integran la arquitectura.



Figura 2. Enfoque ESB.

Además de lo expuesto en los dos párrafos anteriores, en general, los ESB tienen una serie de características comunes:

- **Sincronismo y Asincronismo:** el modelo sincrónico permite la ejecución de tareas dentro de un lapso determinado. Para las tareas que exigen mayor tiempo, el modelo asincrónico es un medio para su ejecución adecuada.
- **Enrutamiento:** esta capacidad provee al ESB la habilidad de administrar la comunicación entre componentes con la finalidad de mantener un acoplamiento débil entre recursos.
- **Transformación de Mensajes:** el ESB tiene la habilidad de procesar mensajes y transformarlos haciendo uso del estándar XSLT (eXtensible StyleSheet Language Transformation) en otros formatos de mensaje.



- **Orquestación de servicios:** es la coordinación de múltiples servicios de implementación presentados como un único servicio agregado.
- **Procesamiento de eventos:** esta capacidad del ESB permite la interpretación de eventos, su correlación y emparejamiento basándose en un conjunto de patrones y reglas.
- **Administración:** provee una interfaz de usuario para la monitorización, auditoría y registro.
- **Soporte multiprotocolo:** esta habilidad permite establecer conectores con aplicaciones de terceras partes, así como conectar diversos sistemas y aplicaciones a través de protocolos comunes como HTTP y SOAP, entre otros.
- **Servicios de seguridad:** (cifrado y firma), entrega confiable, administración de transacciones y soporte a protocolos como HTTPS.

## **1.2. ESTÁNDARES Y ESPECIFICACIONES PARA SALUD.**

Es importante considerar los estándares internacionales para la implementación de SIS, por ello se destacan aquí aquellos que serán de apoyo para el desarrollo de este trabajo.

Existen Organizaciones, como la OMG, promueven la especificación de estándares para las Organizaciones de la industria de la Salud que proponiendo la utilización de tecnología en la optimización de los procesos de negocios con la finalidad de innovar (OMG, 2010).

La organización HL7 es una organización sin fines de lucro, creada en 1987, acreditada por el Instituto Nacional de estándares en USA (American National Standards Institute, ANSI), con el fin de desarrollar estándares que permitan la integración, el intercambio y la recuperación de información en salud en forma electrónica, para soportar la práctica clínica, así como la gestión y ejecución de servicios de salud (HL7, 2007). Los principales estándares a cargo de HL7 son:

- **Mensajería HL7:** es el estándar de mensajería para el intercambio electrónico de datos de salud basada en el Modelo de Referencia de Información (Reference Information Model, RIM)
- **CDA HL7** (Clinical Document Architecture, CDA). Es el estándar de Arquitectura de Documentos Clínicos Electrónicos.
- **SPL HL7** (Structured Product Labeling, SPL). Estándar electrónico de Etiquetado de Medicamentos.
- **HL7 Medical Records:** estándar de Administración de Registros Médicos.
- **GELLO:** estándar para la expresión de reglas de Soporte de Decisiones Clínicas.
- **Arden Syntax:** es un estándar sintáctico para compartir reglas de conocimiento clínico.
- **CCOW:** es un marco de trabajo estándar para compartir contextos entre aplicaciones.

La OMG y HL7 en los últimos años han estado trabajando conjuntamente en la especificación de servicios para SIS enfocándose en el establecimiento de estándares para Arquitecturas Orientadas a Servicios (OMG, 2010). A esta iniciativa se le conoce como el proyecto Healthcare Services Specification Project (Capítulo I, Pág. 14).

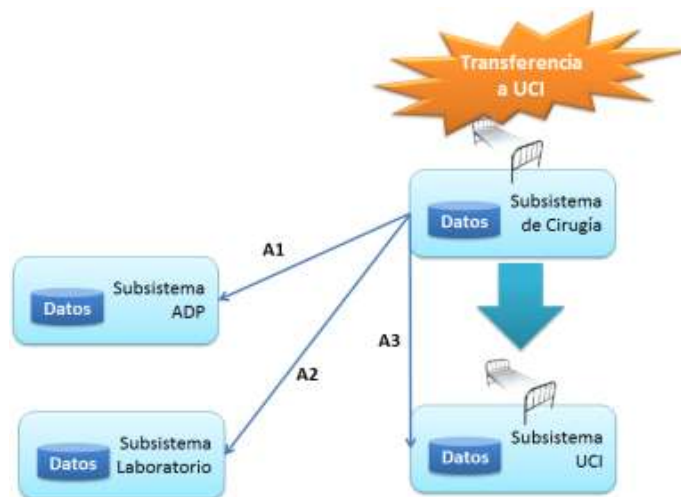
### **1.2.1. ESTÁNDAR DE MENSAJERÍA HL7.**



El estándar de mensajería HL7 permite comunicar sistemas de información dispersos en entornos heterogéneos, los cuales comparten información en diferentes formatos; estos sistemas generalmente envían o reciben mensajes que soportan procesos de registro, admisión y transferencia de pacientes así como pedidos de información al sistema, órdenes, resultados, observaciones clínicas, facturación y actualización de información de archivos maestros (López, 2009)

El intercambio electrónico de datos clínicos y administrativos viene siendo soportado por los mensajes HL7. Actualmente, estas estructuras están definidas por dos versiones del estándar: HL7 Versión 2 y HL7 Versión 3.

Los principales objetivos de HL7 son: la definición de la estructura y la presentación de los mensajes, así como la definición de eventos (Heitmann, y otros, 1999). Un evento (por ejemplo un diagnóstico médico) en la práctica, dispara mensajes que pueden originar un intercambio de datos entre dos o más sistemas que requieren de la información relevante a este proceso.



**Figura 3. Usando mensajes para el evento de transferencia de mensajes (Heitmann, y otros, 1999).**

La Figura 3 muestra el evento de la transferencia de un paciente del Subsistema de Cirugía a la Unidad de Cuidados Intensivos (UCI), esto dispara un evento que envía al Subsistema de Administración de Paciente (ADP, encargada de la admisión, descargo y transferencia de pacientes), de Laboratorio y de Cuidados Intensivos, los mensajes A1, A2 y A3 respectivamente.

Características de HL7:

- **Contempla sintaxis XML:** siendo la tendencia a nivel mundial para el intercambio de mensajes (a partir de la Versión 3 del estándar).
- **Principios de Orientación a Objeto (POO) y Lenguaje Unificado de Modelado (UML):** esto contribuye a los procesos de desarrollo estándar en la industria TI.
- **Uso de vocabularios controlados:** emplea codificaciones internacionales.



Básicamente, los mensajes HL7 Versión 2 utilizan codificación ASCII, para transmitir los datos contenidos en un formato plano, aunque, actualmente hay disponibles especificaciones que emplean otras codificaciones para estos mensajes basados en XML (HL7v3).

### **1.2.2. HEALTHCARE SERVICES SPECIFICATION PROJECT (HSSP).**

Este proyecto (HealthCare Services Specification Project-HSSP) propone ofrecer un conjunto de especificaciones de servicios que hacen uso estándares, así como una guía para la implementación de SOA en SIS (HSSP, 2009). Las especificaciones definidas están enumeradas en la Tabla 1:

**Tabla 1. Especificaciones del proyecto HSSP (HSSP, 2009).**

<b>ESTANDAR</b>	<b>DESCRIPCIÓN</b>	<b>ORGANIZACIÓN</b>
<b>Servicio de Identificación de Entidad: especificación técnica.</b>	Especificación Técnica para el Servicio de Identificación de Entidad, incluyendo los requisitos de la plataforma para SOAP/WSDL/HTTP.	OMG
<b>Servicio de Identificación de Entidad: modelo funcional.</b>	Estándar y Normativa HL7 para la funcionalidad de EIS.	HL7
<b>Servicio de Actualización, Localización y Recuperación.</b>	Especificación Técnica para el Servicio de Actualización, Localización y Recuperación incluyendo los requisitos de la plataforma para SOAP/WSDL/HTTP.	OMG
<b>Servicio de Actualización, Localización y Recuperación: modelo funcional.</b>	Adoptado un Estándar en Construcción para Uso de Prueba de funcionalidades de RLUS.	HL7
<b>Servicio de Soporte de Decisiones: modelo funcional</b>	Adoptado un Estándar en Construcción para Uso de Prueba de funcionalidades de DSS.	HL7
<b>Servicio de Identificación de Entidad: solicitud de propuesta</b>	EIS Solicitud de Especificaciones Técnicas para el servicio EIS.	OMG
<b>Servicio de Recuperación, Localización y Actualización: solicitud de propuesta</b>	EIS Solicitud de Especificaciones Técnicas para el servicio RLUS.	OMG
<b>Servicio de Soporte de Decisión: solicitud de propuesta</b>	EIS Solicitud de Especificaciones Técnicas para el servicio DSS.	OMG
<b>Servicios de Terminología Comunes: solicitud de propuesta</b>	RFP solicitud de especificaciones técnicas para sistemas de toma de decisión (DSS).	OMG

Ofrecen también dos referencia informativas (HL7; OMG, 2008) como guías prácticas, la primera presenta un enfoque para las empresas interesadas en SOA, discutiendo sobre la base de un ejemplo práctico. La guía trata sobre el uso de las normas y estándares de la industria, y el uso de sistemas legados, además de la planificación de líneas de negocio. La guía práctica número dos es un caso de estudio para un sistema de gestión de inmunización, en el cual se describe un modelo de referencia que hace uso de HL7 en sistemas e historias clínicas.



### **1.3. PROYECTOS RELACIONADOS.**

A continuación está la descripción de los proyectos que actualmente están orientados al establecimiento de interoperabilidad en el sector salud. Es importante aclarar que hasta el conocimiento que se tiene en Colombia no existe un proyecto similar al propuesto en este trabajo de grado, que involucre Arquitecturas Orientadas a Servicios para resolver el problema de interoperabilidad técnica y semántica en el sector salud, haciendo uso de estándares abiertos y de herramientas de libre distribución.

#### **1.3.1. PROYECTOS QUE EMPLEAN ARQUITECTURAS ORIENTADAS A SERVICIOS.**

##### **1.3.1.1. Healthcare Services Specification Project-HSSP (HealthCare Services Specification Project-HSSP).**

Este proyecto es el resultado de la unión de esfuerzos entre los organismos de estandarización HL7 y OMG en busca del desarrollo de estándares para SOA en la industria de la Salud, cuya finalidad es promover la implementación de SOA y estándares como HL7 para mejorar la interoperabilidad entre organizaciones y al interior de estas. Para esto, publican recomendaciones que definen responsabilidades, comportamientos y las interfaces adecuadas para lograr el cometido a través de implementaciones y productos. Actualmente han definido especificaciones funcionales para servicios básicos como el Servicio de Identificación de Entidades (Entity Identification Service, EIS), el servicio de Recuperación, Localización y Actualización (Retrieve, Locate, Update Service, RLUS), y el Servicio de Apoyo a la Toma de Decisiones (Decision Support Service, DSS). Cabe destacar que HSSP solamente desarrolla especificaciones funcionales mientras que la especificación de servicios es responsabilidad de la OMG.

##### **1.3.1.2. Distributed E-Healthcare System (Firat Kart, 2009).**

Este proyecto utiliza SOA para reforzar los principios de arquitectura de software y proveer interoperabilidad entre diferentes plataformas y aplicaciones de computación. En su arquitectura plantea tres módulos claramente definidos: módulo Clínico, Farmacia y el de Pacientes, los cuales proveen servicios reales para el sistema de e-Salud distribuido. Este sistema está disponible por medio de una red inalámbrica y puede ser utilizado desde diversos dispositivos como laptops, PDA y Smartphones.

Para facilitar el desarrollo, las implementaciones hacen uso de POJO (Plain Old Java Objects) y del Framework de Spring. Este sistema hace uso de Servicios Web y de un software de reconocimiento de voz llamado DynaSpeak, el cual soporta múltiples lenguajes y no requiere de mucho entrenamiento para un buen desempeño. Emplea tecnologías como Atom y Rss, basadas en XML y facilitan compartir información.

#### **1.3.2. PROYECTOS QUE EMPLEAN BUSES DE SERVICIOS EMPRESARIALES (ESB) Y EL ESTÁNDAR DE MENSAJERÍA HL7.**



### **1.3.2.1. ePOC PDA.**

Es un proyecto del Australian Research Council para la obtención de datos clínicos mediante PDA y la interacción de este con un PC, que permite agregar y consultar observaciones clínicas al sistema de información. El proyecto se basa y hace uso de un trabajo previo que propone mapear los conceptos y relaciones entre SNOMED CT<sup>5</sup> y HL7. Este trabajo describe el uso de un ESB para la comunicación entre dispositivos y presentan la arquitectura con la que se hizo posible la implementación. Los autores mencionan el uso de mensajes XML definidos por HL7 versión 3, para establecer la comunicación del PDA con el PC, pero no especifican si la naturaleza del middleware ESB utilizado es propietario o de libre distribución.

### **1.3.2.2. SerAPI (BLOBEL).**

Este trabajo fue llevado a cabo para apoyar la integración de las necesidades emergentes de los departamentos de sistemas dentro del hospital del distrito de Satakunta en Finlandia en colaboración con el proyecto SerAPI de la Universidad de Kuopio. Plantea un enfoque práctico que envuelve las aplicaciones para servicios web utilizando un Bus de Servicios Empresariales, con el fin de integrar procesos con los sistemas existentes de grupos de diagnósticos relacionados, EPR (Electronic Patient Records), y otras aplicaciones usadas por los funcionarios en momentos determinados. Emplea el protocolo de mensajes HL7 Versión 2 para la comunicación y BPEL para la orquestación de servicios.

## **1.3.3. PROYECTOS QUE EMPLEAN EL ESTÁNDAR DE INTEROPERABILIDAD HL7.**

### **1.3.3.1. National Health Information Infrastructure Program (United Nations Economic and Social Commission for Asia and the Pacific, 2007) (BLOBEL).**

Actualmente los países asiáticos están financiando programas y proyectos orientados a infraestructuras avanzadas para e-Salud Registros Personales de Salud (Personal Health Record, PHR) Registros Personales de Salud. Todas las soluciones consideran estándares internacionales y especialmente especificaciones HL7 versión 3. Utilizan el paradigma de componentes y su implementación utilizando Servicios Web, haciendo énfasis en la necesidad de privacidad y seguridad de servicios.

### **1.3.3.2. National Programme for IT (BLOBEL).**

Es un programa del Servicio Nacional de Salud del Reino Unido para el establecimiento de TI en todo el sistema de Salud del país. Después de definir e implementar una infraestructura para el sistema, ya hay diversos programas que están en funcionamiento, como el Servicio de Registro para el cuidado de NHS y una aplicación para las formulas médicas (prescripciones) electrónicas. No considera el uso de la arquitectura SOA, pero sí los estándares HL7. Este proyecto ha

---

<sup>5</sup> Es una organización que define la terminología médica, y cubre la mayor parte de la información clínica como las enfermedades, resultados, procedimientos, productos farmacéuticos, etc.



demandado altos gastos y retrasos en su puesta en marcha, por eso es bien observado por otros países como caso de éxitos y falla.

### **1.3.3.3. Shareable EHR<sup>6</sup> (BLOBEL)**

Finlandia propone un concepto de sistema EHR distribuido. En ese contexto, la comunicación inter-organizacional e inter-dominio es soportada por sistemas EHR regionales, ofreciendo una vinculación legal de extractos EHR dentro de la arquitectura de seguridad avanzada. La arquitectura y la comunicación EHR está basada en HL7 usando HL7 CDA y mensajes HL7 versión 3.

### **1.3.4. INICIATIVAS NACIONALES DE INTEROPERABILIDAD.**

#### **1.3.4.1. GEL-XML<sup>7</sup> (Gobierno en Línea, 2008).**

GEL-XML es una iniciativa suscrita dentro del ámbito de Gobierno en línea, contempla la definición y gestión de un lenguaje común orientado a que diferentes sistemas de información puedan entenderse e intercambiar información, referente a la prestación y/o demanda de trámites y/o servicios entre: entidades de gobierno, y entidades del sector privado, con el objeto de ofrecer al ciudadano información de forma adecuada, oportuna, eficiente y de calidad. Es concebido para ser utilizado en la implementación de procesos electrónicos de intercambio de información. Sin embargo, considera la extensión de su aplicabilidad a la definición, desarrollo e implementación de nuevos sistemas de información. Tiene en cuenta un componente para integración en e-Salud que está en desarrollo.

#### **1.3.4.2. SISPRO.**

El Sistema Integrado de Información de la Protección Social, es técnicamente una solución de bodega de datos que integra de forma estática las fuentes de datos de sistemas de información en salud, riesgos profesionales, pensiones, asistencia social y empleo en Colombia. Alrededor de SISPRO están en desarrollo varios aplicativos misionales que son sistemas de información específicos. Teniendo como objetivo principal el construir un Sistema de Información, que bajo el concepto de la Protección Social, permita tomar decisiones que apoyen la elaboración de políticas, el monitoreo regulatorio y la gestión de servicios. Actualmente existe en funcionamiento un Portal Web para que los usuarios accedan a la Información del SISPRO, trabajen con los reportes del Sistema de Gestión de Datos, ingresen a los Aplicativos Misionales y al aprendizaje electrónico.

### **1.3.5. OTROS TRABAJOS**

---

<sup>6</sup> Registro Clínico Electrónico (Electronic Health Record, EHR).

<sup>7</sup> GEL-XML (Gobierno En Línea – XML).



### **1.3.5.1. Intel SOA Expressway for Healthcare (Intel, 2010)**

Es un producto diseñado por Intel, con la finalidad de proveer interoperabilidad semántica de datos de Salud. Es una solución basada en redes de servicios de salud y una forma canónica de XML en lugar de utilizar Adaptadores o Buses de Servicios Empresariales.

Ofrece Aceleración XML, diseño con menos código y un incremento en la manejabilidad. Ataca el problema del establecimiento de redes de servicios de salud proveyendo servicios pre construidos específicos, un modelo informático estandarizado y un ecosistema validado robusto para encontrar servicios de salud comunes en el intercambio de información en salud.

Puede integrar y operar entornos heredados .Net, Oracle y J2EE haciendo uso de adaptadores que le permiten centralizar la ejecución de procesos en sí mismo y la mediación entre HTTP, FTP, JMS. Aunque es flexible y basado en estándares (SOAP, REST, ASTM CCS, HL7v2, HL7v3, CD/CCD, IHE, HTTP, HTTPS), implementa una SOA soportando orquestación de servicios empleando WS-BPEL. Caracterizándose como una herramienta de enrutamiento de mensajes de alto costo y baja escalabilidad.

## **1.4. SOA PARA SISTEMAS DE INFORMACIÓN HOSPITALARIOS.**

Una Arquitectura orientada a Servicios representa una vista técnica de la solución de automatización de procesos de negocio, basados en los principios de Orientación a Servicios. Este enfoque es el punto de inicio para el análisis de negocios, que conlleva al proceso de modelado de Servicios orientados por los procesos de negocio. El análisis orientado a Servicios tiene por objetivo determinar cómo los requerimientos de automatización de procesos de negocio pueden ser representados por medio de Servicios. De esta manera, SOA está dirigido por los procesos de negocio y no por los servicios, habilitando la agilidad de negocios, ya que mejora la flexibilidad en los procesos al ser remodelados (Erl, 2005).

### **1.4.1. ORIENTACIÓN DE LA ARQUITECTURA HACIA LOS PROCESOS DE NEGOCIO DE LAS ENTIDADES DE SALUD.**

La OMG y HL7 ofrecen lineamientos (HSSP, 2009) a considerar para la adopción de una SOA para las instituciones en el sector salud, de forma independiente a la metodología empleada en el proceso de implementación, abstrayendo en forma sintética (en ocho pasos) los aspectos a considerar de alta relevancia presentes en ellas. También sugiere algunas consideraciones para la selección de tecnologías.

#### **1.4.1.1. Paso 1. Análisis de la Arquitectura Empresarial.**

Una de las virtudes más importantes que ofrece SOA, es la capacidad de incorporar sistemas Legados junto con nuevos sistemas de información y permitir un trabajo conjunto. Una buena práctica adoptada para la implementación de SOA sugiere el aprovechamiento de la Arquitectura Empresarial Organizacional existente.





Es factible adoptar SOA como parte o extensión de una nueva Arquitectura Empresarial. Para esto, es sugerido el modelo TOGAF (The Open Group, 2008) el cual ha sido definido para el establecimiento de Arquitecturas Empresariales. Modelo que contiene cuatro dominios de arquitectura a saber: negocio, aplicación, datos y tecnológico.

La importancia de establecer la arquitectura empresarial es que permite visualizar el futuro de la organización e institucionalizar la normatividad para garantizar que las nuevas inversiones lleven a la empresa a esa visión. Esta arquitectura empresarial provee una visión global y más entendible de las funciones de negocio, además de permitirnos definir los servicios que contribuirán a la construcción de una SOA.

Este mejor entendimiento permite diferenciar entre las Funciones Principales que corresponden a la organización como tal, y las funciones de soporte/habilitadoras, las cuales permiten a la organización realizar esas necesidades.

#### **1.4.1.2. Paso 2. Estado Actual del Entorno.**

Propone establecer las Líneas de negocio que están siendo soportadas por el entorno de TI de la organización de salud. Algunas de las posibles líneas de negocio serían: farmacia, laboratorio, administración de pacientes, administración de personal, registro, enfermería, radiología/imágenes diagnósticas.

Antes de identificar los servicios necesarios para soportar la empresa, es necesario entender bien el estado actual del entorno. Para esto, es necesario desarrollar una serie de actividades enunciadas a continuación:

- Las funciones de negocio soportadas por la organización.
- Inventario de sistemas existentes.
- Identificación de negocios claves que no están siendo soportados.
- Identificación de los objetivos de negocio y objetivos de calidad.

Una vez adquirido un entendimiento suficiente del entorno actual, es necesario identificar: las redundancias no planeadas, inconsistencias de negocio y necesidades funcionales compartidas, que permitan un adecuado diseño orientado a servicios.

Una correcta identificación y clasificación de los servicios permite emplear adecuadamente la reutilización, composición y modelado de los procesos de negocio. Los siguientes pasos son una referencia que ofrece la OMG y HL7 para garantizar la correcta ubicación de los servicios en la arquitectura.

#### **1.4.1.3. Paso 3. Identificación de Servicios.**

Las organizaciones HL7 y OMG recomiendan hacer una adecuada clasificación de servicios, donde agrupar los servicios comunes, como son los de infraestructura y de Negocio, de los Servicios Únicos para Salud, y así asociarlos de forma adecuada a una Línea de negocio (por ejemplo: farmacia, laboratorio, etc.) y no incurrir en el error de establecer una relación uno a uno



entre servicios y aplicaciones, que ocasiona una redundancia de procesos así como un alto acoplamiento de los servicios que ocasiona una reducción de la flexibilidad del sistema.

De esta forma, contando con sistemas lo suficientemente flexibles, puede responderse ágilmente a los cambios constantes en los servicios de Salud debido a factores externos como lo son las modificaciones en la reglamentación de prestación de servicios por parte del Gobierno.

La clasificación de servicios propuesta es:

- **Servicios Técnicos de Infraestructura:** son todos aquellos comunes a todas las líneas de negocio, que en gran medida, soportan o deben soportar la mayoría de procesos de negocio y que están presentes en todos los dominios de aplicación. Por ejemplo, los Servicios de *Control de Acceso*, y el servicio de *Auditoría* son servicios comunes a todas las líneas de negocio, que pueden ser intervenidos por agentes auditores de procedimientos. En el caso del servicio de control de acceso; la mayoría de líneas de negocio lo requieren para hacer cualquier tipo de procedimiento.
- **Servicios de Negocio:** describen las capacidades que soportan los procesos de negocio; sin embargo no son únicos o exclusivos para un sector de mercado.
- **Servicios Únicos para Salud:** a diferencia de los Servicios de negocio, que no garantizan que sean exclusivos para un mercado vertical específico, estos servicios si deben corresponder exclusivamente para que den soporte a los procesos de negocio específicos para salud.

Es importante en este punto establecer una clara y específica distinción de servicios, que permitirán tener una visión que modela de forma correcta los procesos de negocio de las Entidades de Salud y poder integrarlos al interior de una arquitectura coherente.

La Tabla 2 ilustra un ejemplo de cómo se deben clasificar los servicios, agrupados en tres tipos: de Infraestructura, de Negocio y Únicos de Salud. De igual manera, establecen las líneas de negocio para un caso genérico.

Estos servicios candidatos pueden ser identificados mediante metodologías como “Top Down” o “Bottom Up”. La metodología Bottom Up (Erl, 2005), es importante para la definición de servicios por áreas funcionales de la empresa, que comparten necesidades, en forma de funciones

Para la Línea de Negocio Farmacia puede apreciarse que hace uso de los servicios únicos de salud como Orden de Ingreso, Valoración de Pacientes, Datos de Farmacia, Registro Clínico Electrónico, y de igual manera, de los servicios de negocio como Terminología de Servicios, Facturación así como de Servicios de infraestructura como: autenticación y control de acceso.



Tabla 2. Ejemplo de mapa de funciones de negocio.

LÍNEAS DE NEGOCIO	SERVICIOS ÚNICOS DE SALUD						SERVICIOS DE NEGOCIO						SERVICIOS DE INFRAESTRUCTURA					
	Orden de Ingreso	Orden de Salida	Valoración de Pacientes.	Datos de Laboratorio	Datos de Farmacia	Registro Clínico Electrónico.	Administración de Alertas y Eventos.	Índice Maestro de Personas.	Terminología de Servicios.	Demográficos.	Facturación.	Scheduling.	Servicio de Auditoría.	Administración de Excepciones.	BPM	Autenticación.	Directorio de Servicios.	Control de Acceso.
Farmacia.	X	X	X		X	X	X	X	X	X	X		X	X	X	x		x
Laboratorio.	x	x	X	x		x	x	x	x	x	x	x	X	x	x	x		X
Administración de Pacientes.							x	x		x	x		X	x		x	x	X
Adm. Orden de Ingreso.	x	x				x	x	x	x	x	x	x	X	x	x	x	x	X
Scheduling.							x	x		x		x	X	x	x	x	x	X
Registro.							x	x		x	x	x	X	x	x	x		X
Administración de Atención.	x	x	X	x	x	x	x	x	x	x	x	x	X	x	x	x	x	X
Administración de Remisiones.	x	x	X	x	x	x	x	x	x	x	x	x	X	x	x	x	x	X
Enfermería.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Emergencias.	x	x	X	x	x	x	x	x	x	x	x	x	X	x		x	x	X
Facturación.		x		x	x	x	x	x		x	x		X	x	x	x		X
Imágenes Diagnósticas.		x				x	x	x	x	x	x	x	X	x	x	x	x	X
Soporte para Decisiones Clínicas.			X	x	x	x	x		x	x			X	x		x		X
Facilidades de Administración.							x					x	X	x		x	x	X
Administración para la Nutrición.	x	x		x	x	x	x	x	x	x	x	x	X	x	x	x	x	X



#### **1.4.1.4. Paso 4. Definición de Estados Futuros.**

Definir los estados futuros de la arquitectura es establecer un entorno objetivo dentro del cual el sistema hará transiciones y debe alinear la infraestructura para reducir la redundancia no planeada, maximizar la reutilización y fomentar la coherencia de procesos de negocio.

Mientras estos objetivos son comunes, la arquitectura de estados futuros debería estar diseñada como una plataforma para realizar los objetivos de negocio. Las siguientes actividades están orientadas a contribuir en el establecimiento de estados futuros.

- Evaluar la finalidad de su entorno de negocios
- Identificar áreas de funciones, datos o necesidades compartidas
- Identificar actores de negocio; orientado o dirigido por las entidades que interactúan con la arquitectura.
- Considerar la Organización y sus políticas. Diseñar para anticipar y mitigar cambios.

#### **1.4.2. CONSTRUCCIÓN DE LA ARQUITECTURA.**

Los siguientes lineamientos están orientados a la construcción de una SOA para SIS.

##### **1.4.2.1. Paso 5. Especificación de Arquitectura y Servicios.**

Las recomendaciones hasta aquí están dirigidas a garantizar el óptimo diseño de la arquitectura, sin embargo, es de tener en consideración los requerimientos no funcionales que desean de esta como la disponibilidad, fiabilidad, escalabilidad, entre otros que puedan ser definidos. Los siguientes lineamientos buscan alcanzar estos objetivos.

**Establecer el enfoque arquitectónico:** las implementación de SOA empleando un ESB es la más común, aunque ya está probado que puede establecerse una arquitectura SOA para salud usando otras aproximaciones como la presentada en (Firat Kart, 2009).

Las características a considerar para la elección pueden ser:

- ¿Es un ESB comercial o libre? ¿Para qué servicios?
- ¿Cuál es el protocolo de mensajería preferido? ¿HL7?
- ¿Qué acuerdos de nivel de servicio ofrece?
- ¿Es de latencia aceptable?
- ¿Qué topología se va a apoyar?
- ¿Es una Arquitectura Centralizada? ¿Federada<sup>8</sup>?
- ¿En qué granularidad deseamos gestionar y auditar nuestra seguridad?
- ¿Cómo permite la integración de aplicaciones legadas?

---

<sup>8</sup> Dos o más, EAI o ESB en conjunto para poder ofrecer escalabilidad se pueden configurar como un sistema distribuido que lega funcionalidades.



**Capacidad de encapsulamiento:** es importante identificar la capacidad de encapsulamiento y ocultación de los detalles de la implementación ya que esto ofrece un alto grado de flexibilidad.

**Facilidad de personalización:** antes de negociar la solución (en caso de ser propietaria), estudiar el nivel de integración con los Sistemas Legados.

#### **1.4.2.2. Paso 6. Construcción de un Plan de Transición.**

Considerado como el paso más importante para realizar cualquier tipo de transformación en una organización reside en la capacidad de desarrollar e implementar la planificación de la transición. Es importante tener en cuenta los siguientes lineamientos, sobre todo si en algunas soluciones particulares requieren o estiman una migración hacia nuevas estructuras de datos.

**Establecer fases incrementales basadas en propiedades:** como SOA requiere de servicios comúnmente usados por otros servicios es recomendable identificar y priorizar estos procesos.

**Establecer un enfoque de inversión:** las entidades de salud han hecho grandes inversiones en sistemas de información que atienden sus procesos de negocio aunque poco eficientes trabajando en conjunto. SOA debe proveer un camino para obtener el máximo provecho de estas inversiones, mediante la incorporación de estos sistemas en una nueva arquitectura empresarial, así como verse reflejadas en el Retorno de la Inversión (Return of Investment, ROI) hecha en estos sistemas.

**Plan empresarial de problemas de acceso y dependencias:** es importante definir las dependencias críticas y las rutas contingentes para desplegarlas.

#### **1.4.2.3. Paso 7. Realización de SOA.**

Dada la selección adecuada de la tecnología o el proveedor (comunidad en caso de software libre) es un factor de éxito fundamental; por ende, presentan algunos ítems a considerar como son:

- **Equilibrio entre tecnologías maduras y emergentes.**
- **Usar ESB comerciales:** es una de las características más comunes dadas las necesidades del soporte, sobre todo en las etapas de transición.
- **Establecer métricas de satisfacción:** establecer especificaciones para ir midiendo el nivel de acoplamiento y satisfacción.

#### **1.4.2.4. Paso 8. Despliegue y Sostenimiento.**

En este paso es importante habilitar sistemas de configuración y de control de versiones de las adaptaciones de SOA a los requerimientos dinámicos. Para esto recomienda:

- Publicar reglas de despliegue de servicios y roles para habilitar su ejecución.
- Implementar sistemas de pruebas rigurosas para el despliegue y ejecución de procesos.
- Mantener las responsabilidades de las líneas de negocio.
- Establecer y mantener una firme gobernanza.



- En el diseño se debe considerar:
  - Métricas de granularidad de los servicios.
  - Interfaces usables.
  - Discriminar el uso de interfaces CRUD<sup>9</sup>.

## **1.5. CONCLUSIONES**

Del análisis de los proyectos SOA para salud puede concluirse que la mayoría de estos han hecho un gran esfuerzo en el establecimiento de una SOA basada en Herramientas Middleware como los Buses de Servicios Empresariales, la implementación de Servicios Web y la incorporación de Estándares como es el proyecto HSSP (HealthCare Services Specification Project-HSSP). Este último proyecto, aparte de trabajar fuertemente en la especificación de servicios de salud, plantea una visión clara de cómo emplear SOA para atender a las necesidades de negocio para el Sector Salud.

El proyecto Distributed E-Healthcare System (Firat Kart, 2009), es una implementación de SOA bastante particular, debido a que no usa tecnología de Middleware, sino que más bien, implementa algunas tecnologías para habilitar la capacidad de compartir información entre sistemas como RSS, esto debido a las bajas características de los dispositivos que conforman el sistema.

Proyectos como SerAPI y ePOC PDA descritos en (BLOBEL), hacen uso de Buses de Servicios empresariales para facilitar la comunicación entre dispositivos; sin embargo, subutilizan una herramienta como esta, debido a que usan pocas de las capacidades que estos ofrecen. Cabe resaltar, la importancia en la utilización de Estándares, y el soporte de este Software de Middleware para el intercambio de Mensajes.

Otros proyectos descritos en (BLOBEL), hacen uso de HL7 como estándar de interoperabilidad, sin embargo, no utilizan ningún tipo de Arquitectura como SOA, pero hacen uso de Servicios Web. Estos proyectos, ven la necesidad de plantear a futuro, soluciones más robustas, que respondan realmente a la realidad de los procesos de negocio que atienden.

También existen proyectos que son un intento de interoperabilidad como lo son GEL-XML (Gobierno en Línea, 2008) y SISPRO (Ministerio de la Protección Social, 2008); Gel-XML pretende proveer una pila de estándares soportados para los procesos gubernamentales mediante la definición de un único estándar llamado Gel-XML, mientras el SISPRO propone una integración a nivel de datos de la información contenida por el sistema de la protección social. Sin embargo, estos proyectos no hacen uso de algún tipo de Arquitectura como SOA ni de Buses de Servicios Empresariales; tampoco hacen uso de Servicios Web a excepción de SISPRO, aunque utilicen estándares abiertos. Cabe resaltar la importancia de la utilización de estándares, en la búsqueda de interoperabilidad entre diversos sistemas.

---

<sup>9</sup> Operaciones Crear, Leer, Actualizar y Eliminar(Create, Read, Update, Delete, CRUD).



Interesante resulta encontrar algunas soluciones, como es el caso de Intel SOA Expressway (Intel, 2010), el cual es una Solución Basada en Arquitecturas Orientadas a Servicios, pero no hacen uso de Buses de Servicios Empresariales, aunque, soporta diversos protocolos. Estas soluciones tienden a ser bastante costosas, lo que las hacen inaccesibles para pequeñas y medianas empresas, además de ser soluciones poco escalables.

Dadas las experiencias de los proyectos descritos anteriormente, y teniendo como modelo algunos de estos proyectos (en particular HSSP), cabe resaltar la importancia de emplear los ESB como un componente importante dentro de una SOA, que tenga características de escalabilidad y de interoperabilidad haciendo uso de estándares abiertos.

Hasta ahora se han expuesto los principales estándares y especificaciones de salud en el ámbito de la construcción de sistemas de información enfocados principalmente en lo referente a la interoperabilidad (HL7), y centrando especial atención en la especificación HSSP que es una referencia en el establecimiento de una SOA para SIS, permitiendo con ello concentrarse en los principales puntos a considerar en el establecimiento de una arquitectura empresarial de información permitiendo así establecer un enfoque para el diseño, planificación, e implementación de la arquitectura para SIS, independientemente de la metodología a emplear en el establecimiento de la arquitectura sistémica y tecnológica. Estos enfoques metodológicos se describirán en el próximo capítulo, de manera que permitan hacer una selección adecuada para el despliegue de la SOA solución objetivo de este trabajo.



## **2. CAPITULO II. PROPUESTA DE ARQUITECTURA SOA.**

Este capítulo describe las necesidades que debe soportar la Arquitectura solución atendiendo las recomendaciones de las Organizaciones Internacionales y la organización objetivo de este estudio (Hospital Susana López de Valencia, HSLV). También describe las referencias metodológicas existentes para el diseño y modelado de una SOA; posteriormente plantea una discusión que permita seleccionar la metodología apropiada para el ámbito de Salud y el establecimiento de SOA-SIS. Propone la arquitectura solución a la problemática que atiende este trabajo de grado. Finalmente, son descritos los componentes que constituyen la arquitectura propuesta así como su justificación y la forma en que atiende las necesidades del HSLV.

### **2.1. REQUERIMIENTOS DE LA ARQUITECTURA PARA DAR SOLUCIÓN A LA PROBLEMÁTICA EN EL HSLV.**

Actualmente el Hospital Susana López de Valencia cuenta con una arquitectura empresarial que soporta diversos sistemas de gestión automática de procesos de negocio implementados en diferentes plataformas tecnológicas; es importante que estos sistemas permanezcan operativos ya que la automatización de procesos y el flujo de información que maneja son bastante altos. La arquitectura a proponer debe incluir estos sistemas de manera que estén acompañados de un plan de transacción hacia una arquitectura orientada a servicios a partir de la arquitectura empresarial existente (HL7; OMG, 2008). Esto permitiría al personal de la institución apoyarse en las interfaces de usuario conocidas, facilitando la adición de nuevos sistemas y disminuir el impacto en el empleo de nuevos productos que hacen parte de la arquitectura.

La nueva arquitectura debe dar soporte a las operaciones entre organizaciones, permitiendo compartir recursos y procedimientos con entes territoriales y nacionales que son quienes regulan y fiscalizan el correcto desempeño de las políticas en los servicios prestados por el HSLV. Lo anterior debe permitir el establecimiento de la gestión de seguridad y de acceso.

Actualmente, el departamento de sistemas del HSLV hace frente a cambios en los procesos de negocio y requerimientos que deben soportar los sistemas de información que desempeñan la gestión de procesos y de datos; estos sistemas al ser de naturaleza propietaria, requieren de la intervención directa del proveedor para mantener actualizados los requerimientos funcionales.

Mantener una estrecha relación con las terceras partes (por ejemplo proveedores de tecnología), lleva consigo un compromiso financiero que muchas veces no es fácil de mantener entre los gastos fijos de operación, además, considerando la magnitud de los costos, muchas veces el departamento de sistemas del HSLV adquiere el compromiso de agregar las nuevas funcionalidades operacionales a los procesos soportados. Sin la completa disposición de los códigos fuentes<sup>10</sup> de los sistemas pero con el conocimiento suficiente de las estructuras de almacenamiento, esta labor es llevada a cabo de forma aceptable, es decir: cuando es requerido algún cambio funcional y/o de procesamiento, son empleados procesos almacenados y su

---

<sup>10</sup> Código Fuente de los sistemas de información.





relación con los eventos<sup>11</sup> en los motores de almacenamiento y es realizada agregando elementos<sup>12</sup> a las estructuras de datos de ser requerido; esto significa la omisión de los actuales estándares para la construcción de soluciones dado que la lógica del negocio es desplazada al nivel de procesamiento de datos.

Lo descrito anteriormente sugiere el requerimiento no funcional de **flexibilidad** que debe soportar la arquitectura, de manera que, estando orientada por los procesos de negocio, permita responder a los nuevos requerimientos funcionales sin que los que procesos de negocios que dependen de los sistemas presentes sean afectados en su operación. Lo anterior provee mecanismos para responder ágilmente a los cambios requeridos.

Si lo expuesto anteriormente es soportado por la arquitectura, con la orientación a servicios para la construcción de sistemas de gestión de procesos de negocio, garantiza la administración de estos, convirtiéndola en una base sólida para la construcción de nuevos sistemas informáticos.

También el HSLV desea adquirir a futuro nuevos sistemas de información que automaticen parte de sus procesos de negocio que están pendientes. Para ello, requiere que la arquitectura sea **escalable** para que puedan incorporar sus nuevos sistemas sin afectar los ya existentes.

De esta manera, la arquitectura debe soportar diversas plataformas tecnológicas, y lo más importante, que este en la capacidad de manejar estándares abiertos tanto para salud, como para comunicación en entornos de computación.

Además de esto, el departamento de sistemas del HSLV requiere que las tecnologías utilizadas para la solución, cuenten con herramientas que les permita hacer desarrollos rápidos, así como una documentación adecuada de las nuevas soluciones soportadas por la arquitectura, de forma que sea reutilizable por el recurso humano del departamento.

### **2.1.1. INTERFACE DE INTEROPERABILIDAD HL7.**

La Arquitectura a definir debe incorporar capacidades de interoperabilidad mediante el uso de estándares de intercambio de información médica, concretamente el estándar HL7. Esto puede lograrse incorporando un motor de interfaces HL7. Las principales funciones de estos motores de interfaces son:

- Garantizar el almacenamiento de estructura de mensajes.
- Transformación de mensajes (mapeando datos de estructuras de almacenamiento u otros mensajes).
- Canales de comunicación y enrutamiento de los mensajes.
- Interface gráficas de usuario para la configuración y administración.

Mediante el uso de un motor de interfaz HL7 es posible obtener los beneficios de los sistemas existentes de información hacer inversión en nuevas tecnologías, reduciendo los costos y

---

<sup>11</sup> Ejecución de procedimientos mediante SQL o Triggers.

<sup>12</sup> Agregar tablas a las estructuras actuales de almacenamiento.



extender la vida y la eficiencia de los sistemas actuales. También hay oportunidad de vincular a los sistemas fuera de la institución de la salud, tales como proveedores de servicios externos.

Los motores de interfaz HL7 son software que funciona como un intermediario entre los diferentes sistemas. Siguen de cerca los diferentes tipos de interfaces y puntos de comunicación y realiza acciones de acuerdo a reglas definidas por la organización.

El estándar HL7 trabaja con una serie de normas (Conceptuales normas, normas de documentos y normas de aplicación y las normas de mensajería). Define estándares de mensajería cómo la información es empaquetada y comunicada de un sistema a otro.

Varias de estas funcionalidades pueden ser proveídas si el enfoque arquitectónico está orientado hacia el empleo de un Bus Empresarial de Servicios que permita la transformación de mensajes HL7.

## **2.2. DEFINICIÓN DE LA METODOLOGÍA PARA LA CONSTRUCCIÓN DE LA ARQUITECTURA.**

Desde sus comienzos, el desarrollo de software ha tenido una rápida evolución, apareciendo durante este proceso diversos paradigmas o enfoques como son la programación estructurada, la modular, la orientada a objetos, los sistemas distribuidos y en particular los distribuidos orientados a objetos. Esto ha impulsado a el establecimiento de diferentes disciplinas basadas en métodos, técnicas y herramientas que ayuden a los desarrolladores en las diversas etapas de la construcción de los sistemas de Información, para cada uno de estos enfoques (de la Torre, 2008). Esta sección expone las ayudas metodológicas de aplicación para sistemas basados en SOA, seleccionado la más ajustada al proceso de diseño.

Existen actualmente importantes referencias metodológicas a considerar, pero es indispensable emplear la más adecuada. Esto significa que esté estrechamente relacionada con las exigentes características de la gestión de procesos de negocio, y así dar soporte a ellos en forma estructurada y globalizada, es decir, que cumpla con estándares internacionales. De esta forma es necesario el empleo de estándares abiertos permitiendo que la interoperación con otros sistemas basados en ellos sea realizable. A continuación son descritas las aproximaciones más importantes.

### **2.2.1. MODELO DE REFERENCIA RM-ODP.**

Los organismos de estandarización internacional establecen modelos de referencia para garantizar el procesamiento distribuido de sistemas abiertos. A un nivel general, la metodología ha adoptar para la construcción de una SOA debe seguir los lineamientos de un modelo de referencia ampliamente aceptado como lo es el Modelo de Referencia para el Procesamiento Abierto Distribuido (Reference Model – Open Distributed Processing, RM-ODP) de la Organización Internacional para la Estandarización (International Organization for Standarization, OSI) que se describe a continuación.



Tal y como lo definen ISO e ITU-T (International Telecommunication Union-Telecommunication Standardization Sector), el Modelo de Referencia de Procesamiento Abierto y Distribuido proporciona un marco de coordinación para la normalización del desarrollo de aplicaciones abiertas y distribuidas, creando una arquitectura capaz de soportar de forma integrada, aspectos tales como la distribución, interoperabilidad y portabilidad de los sistemas, objetos y componentes (Vallecillo, 2001).

RM-ODP maneja la complejidad a través de una separación de incumbencias, direccionando los problemas específicos desde diferentes puntos de vista. Provee una clara y explícita especificación de conceptos y construcciones que definen semánticas, independientemente de la representación, metodologías, herramientas y procesos utilizados en el desarrollo de aplicaciones abiertas.

Los puntos de vista definidos por el marco de referencia son:

- **Empresarial.** Define y describe los requerimientos del negocio y la forma de responder a ellos, centrándose en el propósito, ámbito y políticas para el sistema.
- **Información.** Establece y define la estructura de la información que debe soportar el sistema, así como las restricciones a considerar en su interpretación definiendo su semántica y su procesamiento.
- **Computacional.** Describe la funcionalidad del sistema, habilitando la distribución a través de su descomposición y organización funcional basándose en objetos definidos por interfaces los cuales interactúan entre sí.
- **Ingeniería.** Establece la infraestructura necesaria para soportar el procesamiento distribuido, las formas de distribución de datos y las operaciones que dan soporte a la funcionalidad del sistema.
- **Tecnológica.** Describe la tecnología que dará soporte al sistema basado en el hardware, software y comunicaciones, que permitan la representación y distribución de los datos.

Las principales características que aporta este modelo de referencia son:

- Emplea un modelo de objetos común apoyándose en unas técnicas de orientación a objetos.
- Define unas fachadas como mecanismos para ocultar la complejidad en la definición de cualquier sistema distribuido.
- Describe, como parte de su arquitectura básica, funciones comunes a todos los sistemas abiertos y distribuidos. Organizadas en cuatro grupos –funciones de gestión, coordinación, repositorio y seguridad.

### **2.2.2. EL MARCO DE DESARROLLO PARA SISTEMA DE INFORMACIÓN EN SALUD**

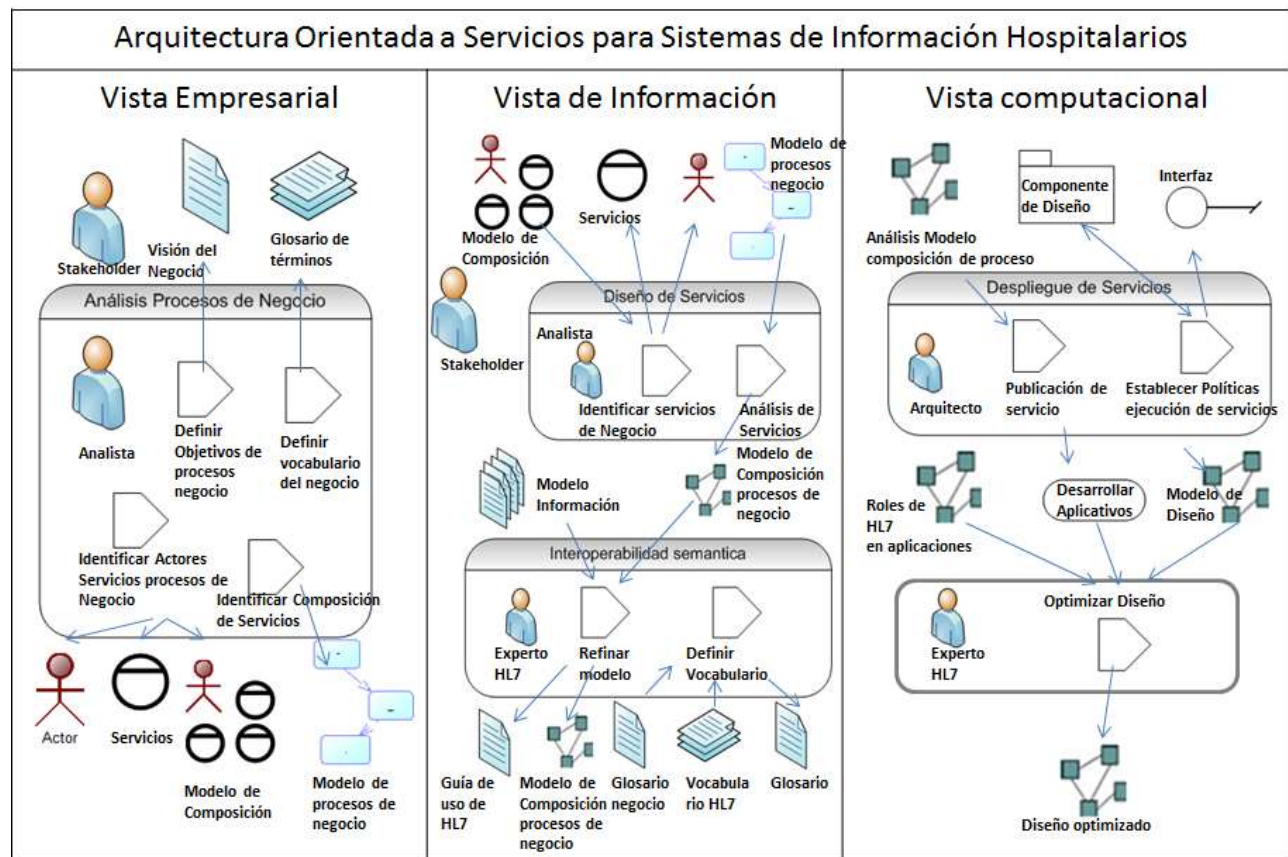
HIS-DF (Health Information System – Developer Frametwork) es una metodología especialmente diseñada para el análisis de requerimientos, diseño, implementación, evaluación, uso y mantenimiento de sistemas de información en salud basada en el estándar RM-ODP. HIS-DF provee una base de conocimiento de referencia de tareas, roles, artefactos y la guía necesaria



para el análisis de la arquitectura dentro de proyectos de desarrollo de software en salud. HIS-DF soporta el desarrollo de HIS mediante el uso de modelos UML organizados de acuerdo a los puntos de vista de RM-ODP como se muestra en la Figura 4 (López, 2009)

### 2.2.2.1. Punto de vista empresarial.

El Punto de Vista Empresarial analiza la arquitectura empresarial y su entorno en el contexto de los objetivos de negocio. En esta etapa son analizados los procesos de negocio y los actores externos de los sistemas asociados. Los principales modelos de la arquitectura generados en el Punto de Vista Empresarial son: la Visión del Negocio, Modelo de Procesos de Negocio, el Modelo de composición de Servicios y el Modelo de Análisis del Negocio.



**Figura 4. Tareas, artefactos y roles en el proceso de desarrollo de la arquitectura del sistema. Adaptado de (Aguilar, y otros, 2009)**

### 2.2.2.2. Punto de vista de Información.

El punto de partida son los servicios que representan los procesos de negocio y el modelo de composición de estos. El objetivo del diseño del modelo de composición de procesos de negocio sus datos de entrada y sus datos de salida son la información que alimenta un proceso de análisis y definición de la semántica de la información. Así son obtenidos los principales productos, entre ellos: la Guía de uso de HL7, el vocabulario HL7; ellos aportan a la optimización del Modelo de Composición de modelo de negocio.



### 2.2.2.3. Punto de vista computacional.

El Punto de Vista Computacional es el encargado de la descomposición funcional del Subsistema de Integración de componentes definidos por la funcionalidad en la arquitectura, especificando los componentes lógicos y sus interacciones. HIS-DF describe un enfoque top-down, en el cual, los componentes computacionales son derivados a partir de: los casos de uso del sistema, en el Punto de Vista Empresarial y los modelos de análisis, en el Punto de Vista de Información. La descripción computacional de la arquitectura incluye produce los siguientes artefactos: Roles de HL7, servicios y modelo de Diseño de la arquitectura.

### 2.2.3. METODOLOGÍA DE REFERENCIA SOA.

La definición de una Solución Orientada a Servicio toma sentido cuando es constituida una arquitectura para un entorno específico, que establece los servicios, componentes y flujos de estos que soportan de manera colectiva los procesos y objetivos de negocio de la empresa (Bieberstein, y otros, 2008). La arquitectura de referencia provee características y definiciones para cada una de las capas que define y las relaciones entre ellas permitiendo ubicar cada uno de los bloques construidos en su capa correspondiente, así como es presentado en la Figura 5.

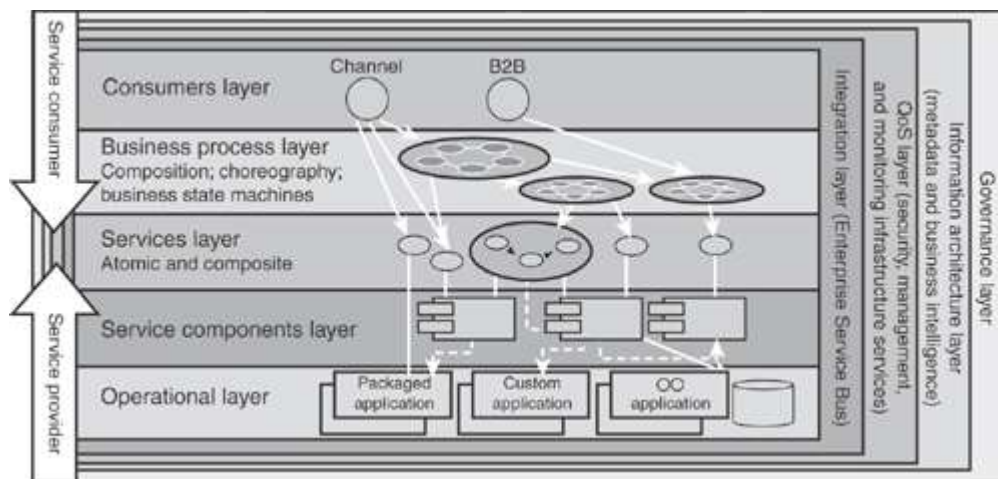


Figura 5. Vista Lógica de la Arquitectura de Referencia SOA. Tomada de (Erl, 2005)

La arquitectura de Referencia SOA define cinco capas verticales y cuatro horizontales, como puede verse en la Figura 5. Como toda arquitectura basada en capas, existen capas de nivel superior que pueden acceder a las capas inferiores y sus Bloques Construidos para la Arquitectura (Architecture Building Blocks, ABB), mientras que las capas inferiores no pueden hacer lo mismo con los superiores y sus ABB. Las capas verticales pueden ser aplicadas o utilizadas por uno o muchos ABB de las capas horizontales.

A continuación se enumeran las capas de la Arquitectura de Referencia:

- **Capa 1. Sistemas Operacionales:** todos los sistemas existentes en un entorno de TI que soportan las actividades de negocio.



- **Capa 2. Componentes de Servicio:** abstracción de las funcionalidades de los activos existentes (Capa 1) y se encapsulan para generar nuevas funcionalidades, expuestas en componentes de Servicio.
- **Capa 3. Servicios:** está compuesta por todo el conjunto de servicios para un portafolio de Servicios empresariales. Estos servicios pueden ser compuestos o simples, para los cuales se define la sintáctica y semántica de la información.
- **Capa 4. Procesos de Negocio:** representa los procesos como una orquestación o una composición de servicios débilmente acoplados. Esta capa encarga de la administración del ciclo de vida de los procesos con sus orquestaciones y coreografías
- **Capa 5. Consumidor:** expone los medios de comunicación entre los sistemas y los usuarios, suministrando las facilidades para la construcción de servicios; además, también proveen mecanismos de acceso a las funcionalidades del Sistema.
- **Capa 6. Integración:** los requerimientos no funcionales (Seguridad, Latencia, QoS<sup>13</sup>) son implementados en los ABB de esta capa. Esta capa se encarga de dar soporte a la transformación de protocolos basados en estándares abiertos, permitiendo la interoperabilidad.
- **Capa 7. Calidad de Servicio:** provee las capacidades de infraestructura para realizar los requerimientos no funcionales.
- **Capa 8. Arquitectura de la Información:** asegura la representación apropiada de los datos y la información requerida en una SOA.
- **Capa 9. Gobernanza:** asegura una administración apropiada del ciclo de vida completo de los servicios.

Definen un conjunto de fases, que comprenden una estructura metodológica que soporta la construcción de SOA. Las fases básicas para la construcción de SOA comprenden Análisis y Diseño Orientados a Servicios, Desarrollo, Pruebas, Despliegue y Administración de Servicios, lo que en general define el ciclo de vida de una Solución SOA según la metodología propuesta por Erl (Erl, 2005).

A continuación son descritas las consideraciones particulares de la metodología, a tener en cuenta cuando es realizada la construcción de una SOA.

#### **2.2.3.1. Fase 1. Análisis Orientado a Servicios.**

En esta fase, los requerimientos y funcionalidades del sistema deben traducirse en servicios, clasificados en distintos niveles, los cuales posteriormente serán modelados como servicios candidatos que conformarán una base preliminar de los servicios de la Arquitectura. Esto nos permite estimar el potencial que tiene para modelar y soportar procesos de negocio.

#### **2.2.3.2. Fase 2. Diseño Orientado a Servicios.**

---

<sup>13</sup> Calidad de Servicio (Quality of Service, QoS).



Una vez definido las soluciones a construir, es necesario determinar cómo debería ser construido. Esta fase debe ser orientada por estándares<sup>14</sup> que incorporen las convenciones de la industria y los principios de orientación a servicios dentro de procesos de diseño de servicios (Erl, 2005). Esta fase confronta los diseñadores de servicios con las decisiones claves que establecen los límites de la lógica encapsulada en estos servicios.

#### **2.2.3.3. Fase 3. Desarrollo de Servicios.**

Esta fase de construcción de los servicios. Aquí las características de las plataformas específicas de desarrollo definen el tipo de servicio. Es decir el lenguaje de desarrollo escogido y el entorno de desarrollo determinaran la forma física de los servicios y los procesos de negocio orquestados de acuerdo con sus diseños.

#### **2.2.3.4. Fase 4. Prueba de Servicios.**

Debido a la naturaleza genérica y al potencial para ser reutilizados y compuestos, los servicios deben someterse a pruebas rigurosas para su posterior despliegue en un entorno de producción.

Aspectos a considerar son los siguientes:

- Verificar los tipos de consumidores de los servicios y sus niveles de acceso.
- Nivel de Satisfacción de las políticas de Servicio.
- Comprobar las condiciones y excepciones a las que está sujeto el servicio.
- Facilidad para la Composición de Servicios.
- Facilidad para el Descubrimiento de Servicios.
- Comprobar el efecto que tiene la introducción de nuevos servicios en el Desempeño de la Arquitectura.

#### **2.2.3.5. Fase 5. Despliegue de Servicios.**

La fase de implementación trae consigo los elementos para la instalación y configuración de componentes distribuidos, interfaces de servicios y software de mediación en entornos de producción.

En esta fase, los indicadores de evaluación son:

- Distribución de los servicios.
- Verificar si la infraestructura soporta completamente los requerimientos de procesamiento de todos los servicios.
- Establecer si la introducción de nuevos servicios afectará las aplicaciones y servicios existentes.
- Determinar el impacto de la introducción de Mediadores en el entorno.

---

<sup>14</sup> En el dominio de este proyecto (e-Salud) los estándares son ofrecidos como lineamientos de la OMG y HL7: HSSP (Pág.18).



- Configuraciones de seguridad.
- Soporte de los requerimientos de escalabilidad.
- Determinar cómo serán sostenidos y monitoreados los sistemas legados encapsulados.

#### **2.2.3.6. Fase 6. Administración de Servicios.**

Los aspectos más relevantes de esta fase son:

- Después del despliegue de los servicios, toman importancia los aspectos de administración de aplicaciones estándar.
- La preocupación viene a centrarse en las formas en que serán monitoreados los servicios que son utilizados o consumidos.
- Seguimiento de desempeño y la administración de los mensajes.

#### **2.2.4. METODOLOGÍA SOMA.**

Una alternativa metodológica es la Arquitectura y Modelado Orientado a Servicios (Service Oriented Modeling and Architecture, SOMA), propuesta por IBM para la construcción de SOA. Esta Metodología plantea cuatro fases para el desarrollo de servicios, que no son las mismas fases del Proceso de Desarrollo Unificado (RUP). Estas son: fase de Identificación, Especificación, Realización y Despliegue de Servicios. (Ali Arsanjani, 2004) Sin embargo, estas fases se pueden ver claramente definidas en las establecidas por RUP, por ejemplo, la identificación y especificación de servicios son principalmente actividades de la fase de Elaboración y las tareas asociadas son similares a las de la disciplinas del RUP (Delgado, y otros).

El lenguaje de SOMA es sencillo, claro y muy centrado en ofrecer una metodología para resolver los retos a los que está expuesta la comunidad de desarrollo de TI con respecto al diseño orientado a servicios. Es importante que las herramientas de desarrollo incluyan soporte para los procesos o métodos de desarrollo software. RUP extiende sus técnicas de procesos para dar soporte al Diseño Orientado a Servicios haciendo uso de los métodos definidos en SOMA (extensiones de RUP para SOMA). Este método está disponible como una extensión en un producto del Rational Software llamado Rational Method Composer (Bieberstein, y otros, 2008).



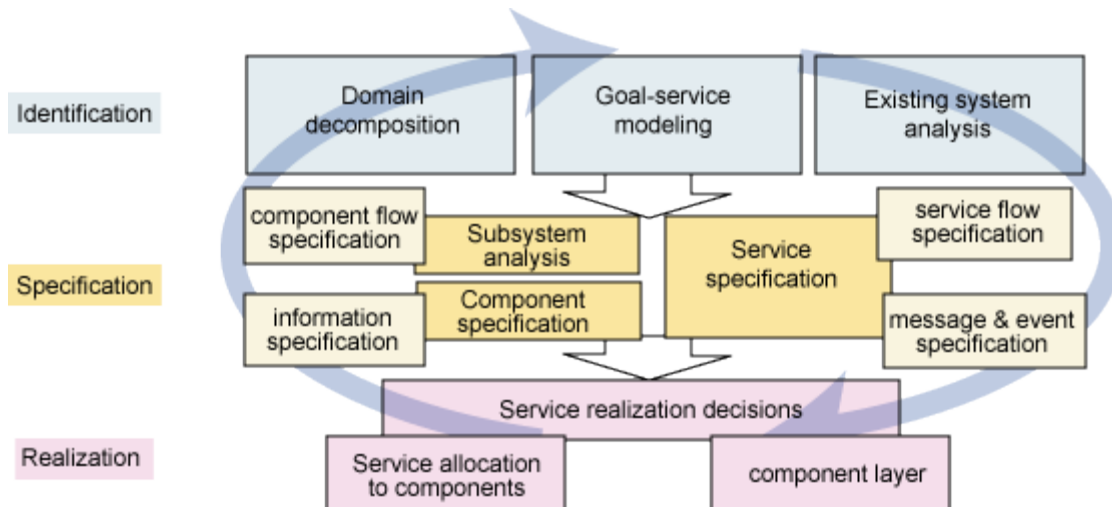


Figura 6. Etapas definidas por la metodología SOMA. Tomada de (Ali Arsanjani, 2004)

A continuación son descritas las etapas de la metodología SOMA mostradas en la Figura 6.

#### 2.2.4.1. Identificación de Servicios

Este proceso consiste de la combinación de análisis Top-Down, Técnicas de Mediación de Descomposición de Dominios, análisis de actividades existentes y Modelado de Objetivos de Servicios. La descomposición de dominio es dividida en otras áreas funcionales y subsistemas, incluyendo sus flujos o descomposición de procesos en otros procesos, sub procesos y casos de uso de negocio de alto nivel. Estos casos de uso frecuentemente son buenos candidatos para servicios de negocio.

En el Análisis de Sistemas Existentes<sup>15</sup>, los sistemas heredados (existentes en la organización) son analizados y seleccionados como candidatos viables para proveer soluciones de bajo costo para la implementación de funcionalidades de servicios que soporten los procesos de Negocio. En este proceso, son analizadas las Interfaces Programadas de Aplicación (Application Programming Interface, API), transacciones y módulos de Aplicaciones Legadas y Empaquetadas.

En algunos casos, la reestructuración en componentes de sistemas heredados es necesaria para volver a hacer modular las actividades existentes, a fin de soportar funcionalidades de servicio.

Existen enfoques orientados al modelado de servicios a través de la definición de los objetivos que permite encontrar otros servicios que no fueron identificados por las técnicas de Top-down y Bottom-up.

#### 2.2.4.2. Especificación

**Clasificación o Categorización de Servicios:** es importante para empezar la clasificación de servicios emplear una jerarquía que refleje la composición o naturaleza de los servicios: los

<sup>15</sup> Empleando la metodología Bottom Up para los Procesos actuales.



servicios pueden y deberían estar compuestos de componentes y servicios de grano fino (fine grained). Además, esto ayuda a evitar el síndrome de proliferación de servicios en el cual un creciente número de servicios de baja granularidad consigue definir, diseñar y desplegar servicios con muy poca Gobernanza, resultando en un mayor desempeño, escalabilidad y aspectos de manejo.

**Análisis de Subsistemas:** toman los subsistemas identificados en la descomposición de dominios y se establecen las relaciones entre ellos. El análisis del subsistema consiste en la creación de modelos de objetos para representar el funcionamiento interno y los diseños de los subsistemas contenidos que expondrán los servicios y su realización. La construcción del diseño de subsistemas serán realizados como la construcción de una implementación de componentes de baja granularidad realizando los servicios en posteriores actividades.

**Especificación de Componentes:** detallan los componentes que implementan lo servicios para lo cual es necesaria la especificación de datos, reglas, servicios, perfiles de configuración, variaciones. La mensajería, la especificación de eventos y definición de manejo ocurren en esta actividad.

### **2.2.4.3. Realización**

**Distribución de Servicios:** asignan servicios a los subsistemas que han sido identificados, haciendo una correspondencia entre servicios y subsistemas en cada una de las capas de la SOA.

**Realización de Servicios:** este proceso implica integración, transformación, suscripción y publicación de partes de la funcionalidad usando Servicios Web. Es importante identificar qué módulos de los sistemas legados servirán para construir nuevos servicios, así como cuáles deben ser creados. Es importante tener en cuenta decisiones de realización que incluyan aspectos de seguridad, manejo y supervisión de servicios.

### **2.2.5. DISCUSIÓN Y SELECCIÓN DE LA METODOLOGÍA.**

RM-ODP y HIS-DF son metodologías genéricas que no están adaptadas a los requerimientos esenciales de soluciones basadas en SOA. Las metodologías SOA y SOMA, abordan las características de gestión para modelado de negocio, el diseño de la solución y lineamientos para su implementación de soluciones SOA, haciendo una definición de actividades, entregables y roles, que estructuran un proceso de desarrollo metodológico adecuado para este tipo de soluciones (Erl, 2005) (IBM Corp, 2007). Otro factor común para la identificación de servicios en ambas metodologías es la importancia de combinar los tres enfoques: top-down, bottom-up y análisis de modelado de objetivos. En ambas los procesos de negocio son modelados como casos de uso.

Adicional a la Identificación, especificación y realización, las metodologías incluyen otras actividades del enfoque de modelado orientado a servicios que incluyen el despliegue, monitoreo, manejo y gobernanza requerida para soportar completamente el ciclo de vida SOA.



Aun cuando las dos metodologías están basadas en los casos de uso del negocio, la metodología SOA involucra otras actividades y entregables de la disciplina de requerimientos del RUP, mientras que SOMA deriva los servicios directamente del caso de uso de negocio, siendo esta una diferencia en el enfoque metodológico.

Por la familiarización que existe con el Proceso unificado de Desarrollo RUP en el personal del hospital Susana López de Valencia les permite utilizar SOMA tal cual está definido, a pesar del esfuerzo de su incorporación dada la cantidad de nuevos elementos que define. Otra opción es particularizar el enfoque metodológico para necesidades específicas, para lo cual la metodología SOA es una guía útil para la selección de elementos más relevantes, siendo también una opción menos compleja que presenta un enfoque conceptual similar.

Es importante destacar que adjunto a las directivas metodológicas, SOMA aporta herramientas para el proceso de desarrollo (como lo es la extensión de Rational Method Composer), y otras herramientas propietarias que facilitan y reducen algunas actividades en el desarrollo bajo el modelo. Sin embargo estas herramientas no son de libre distribución.

Finalmente la disponibilidad de herramientas de libre distribución es el factor decisivo que es pertinente a los intereses de este proyecto, por lo tanto empleará la metodología de referencia SOA, considerando también que es posible incluir allí las especificaciones de HSSP (estándares para el dominio de eSalud). La presentación de las diferentes vistas de la arquitectura será teniendo en cuenta las principales vistas arquitectónicas descritas en el estándar RM-ODP.

### **2.3. VISTA EMPRESARIAL DE LA ARQUITECTURA.**

La arquitectura propuesta busca dar solución a los requerimientos identificados en la sección 2.1, además debe tener un guía que permita al HSLV adoptar las tecnologías SOA-SIS para el desarrollo de futuros proyectos. La arquitectura propuesta está soportada por tecnologías y herramientas de libre distribución, lo que permite la reducción de costos en su implementación, así como la incorporación de sus sistemas existentes y futuros dentro esta arquitectura empresarial.

El análisis y diseño orientado a servicios requiere una correcta definición de las líneas de negocio ya que estas permitirán la correcta ubicación de los servicios dentro de la arquitectura, de esta forma el análisis de arquitectura empresarial propuesto en HSSP es útil en la revisión de operación de grupo de los servicios realizada en el análisis orientado a servicios propuesto por la metodología de referencia SOA, de esta forma el mapa de funciones de negocio (estudiadas en el anexo A), corresponden a la vista empresarial del modelo RM-ODP a un nivel alto de abstracción.



Tabla 3. Vista empresarial del HSLV / Mapa de funciones del negocio.

LÍNEAS DE NEGOCIO	SERVICIOS ÚNICOS DE SALUD						SERVICIOS DE NEGOCIO						SERVICIOS DE INFRAESTRUCTURA					
	Orden de Ingreso	Orden de Salida	Valoración de Pacientes.	Datos de Laboratorio	Datos de Farmacia	Registro Clínico Electrónico.	Administración de Alertas y Eventos.	Índice Maestro de Personas.	Terminología de Servicios.	Demográficos.	Facturación.	Programación de Citas.	Servicio de Auditoría.	Administración de Excepciones.	BPM	Autenticación.	Directorio de Servicios.	Control de Acceso.
Farmacia.	X	X	X		X	X	X	X	X	X		X	X	X	X		X	
Laboratorio.	X	X	X	X		X	X	X	X	X	X	X	X	X	X		X	
Administración de Pacientes.	X	X					X	X		X	X		X	X		X	X	
Adm. Orden de Ingreso.	X	X				X	X	X	X	X	X	X	X	X	X	X	X	
Programación de Citas.	X						X	X		X		X	X	X	X	X	X	
Administración de Atención.							X	X		X	X	X	X	X	X		X	
Administración de Remisiones.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
Enfermería.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
Emergencias.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	
Cirugías.	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X	X	
Salud Mental.		X		X	X	X	X	X		X	X		X	X	X	X	X	
Facturación.		X				X	X	X	X	X	X	X	X	X	X	X	X	
Cartera.			X	X	X	X	X		X	X		X	X		X		X	
Contabilidad.		X			X		X				X	X	X		X	X	X	
Cartera.	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	
Imágenes Diagnósticas.	X	X	X		X	X	X	X	X	X		X	X	X	X		X	
Gestión de la calidad.	X	X	X	X		X	X	X	X	X	X	X	X	X	X		X	
Administración para la Nutrición.							X	X		X	X		X	X		X	X	



Tabla 3 contempla solo la relación de operación el grupo de los servicios que corresponden a los procesos de negocio con los que fue diseñada la arquitectura (Anexo A) y que serán descritos en el capítulo IV.

### 2.3.1. VISTA COMPUTACIONAL DE LA ARQUITECTURA.

La Figura 7 muestra vista computacional de la arquitectura propuesta, permite observar los diferentes componentes así como las interfaces de adaptación entre ellos. Esta arquitectura es lo suficientemente modular para desacoplar los proveedores y los consumidores, dejando que la infraestructura SOA sea la encargada de las transacciones, publicación y suscripción de servicios, así como del enrutamiento de mensajes y su transformación de formato.

Esta es una Arquitectura que centraliza la administración, lo que permite monitorear y evaluar el desempeño de los elementos que la componen. Esta expone de forma adecuada todos los recursos disponibles en el entorno empresarial.

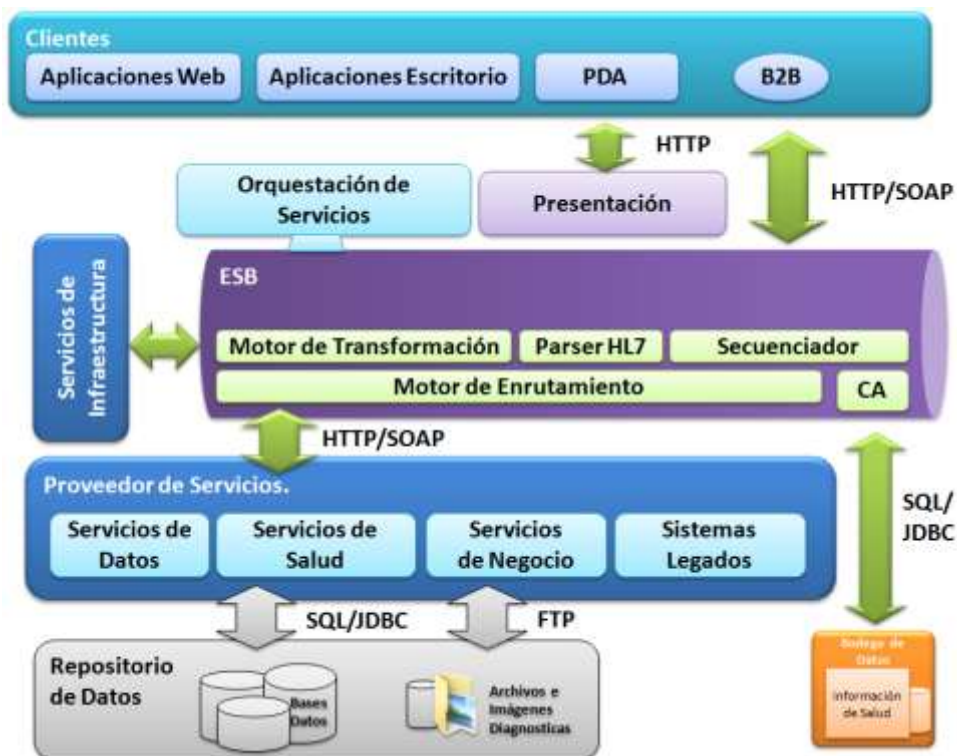


Figura 7. Vista computacional de la Arquitectura Propuesta SOA-SIS para el HSLV.

Está basada en un Bus de Servicios Empresariales como componente principal, el cual permitirá emplear al máximo las características que ofrece SOA para SIS en el establecimiento de un adecuado nivel de interoperabilidad entre procesos.

La arquitectura propuesta está diseñada para soportar la reutilización de los recursos y activos existentes en el entorno de computación, lo que facilita su aprovechamiento en la composición de nuevos sistemas de información; estos deben establecer únicamente una comunicación con el Bus de Servicios Empresariales para poder acceder a estos recursos. Esta característica es una



ventaja, ya que centraliza la administración, las reglas de seguridad y de autenticación que facilitan un adecuado manejo de los recursos disponibles en el ESB.

Estos activos soportan algunos procesos específicos de la empresa, motivo por el cual, es importante mantenerlos en funcionamiento e incluirlos dentro de la nueva arquitectura empresarial, de forma que reutilicen las capacidades individuales que aportan para generar nuevas funcionalidades que exijan su trabajo en conjunto. Conservarlos reduce el impacto en la transición hacia una nueva arquitectura basada en el paradigma de orientación a servicios.

El adecuado aprovechamiento de estos activos depende de la estructuración y clasificación de servicios empleados para habilitarlos. El paradigma de Orientación a Servicios hace hincapié en el establecimiento de interfaces de conectividad adecuadas para el acceso y manejo de la información contenida en ellos.

En este punto es importante resaltar el papel del Bus de Servicios Empresariales en la integración de recursos y activos existentes dentro de una nueva arquitectura. Este componente permite ventajas como el enrutamiento basado en contenidos y la transparencia en las ubicaciones de los servicios, que permite un bajo acoplamiento extremo a extremo. Es importante mencionar el soporte de protocolos de seguridad que proveen facilidades para administrar reglas de acceso. Este componente soporta protocolos y estándares así como transformación de mensajes para el intercambio de información entre distintos formatos, generando los medios para establecer un nivel adecuado de interoperabilidad entre sistemas.

Las capacidades del ESB, para la exposición y disponibilidad de los servicios necesarios para modelar los Procesos de Negocio, hacen a la arquitectura altamente flexible en la definición de los actuales y nuevos procesos que soporten los cambios en los objetivos de negocio, haciendo uso de los activos disponibles en el entorno.

La arquitectura propuesta es altamente escalable haciendo que la adición de más sistemas de información no genere un problema en la comunicación entre ellos, debido a que el bus de servicios empresariales es el único mediador, facilitando los servicios básicos que permitan el intercambio de información entre procesos.

### **2.3.2. DESCRIPCIÓN DE BLOQUES Y COMPONENTES DE ARQUITECTURA COMPUTACIONAL.**

De acuerdo con el bajo acoplamiento que ofrece la arquitectura propuesta son establecidos dos módulos, ellos corresponden con los proveedores de servicios y los consumidores de los mismos: módulo Clientes y módulo de Servicios respectivamente.

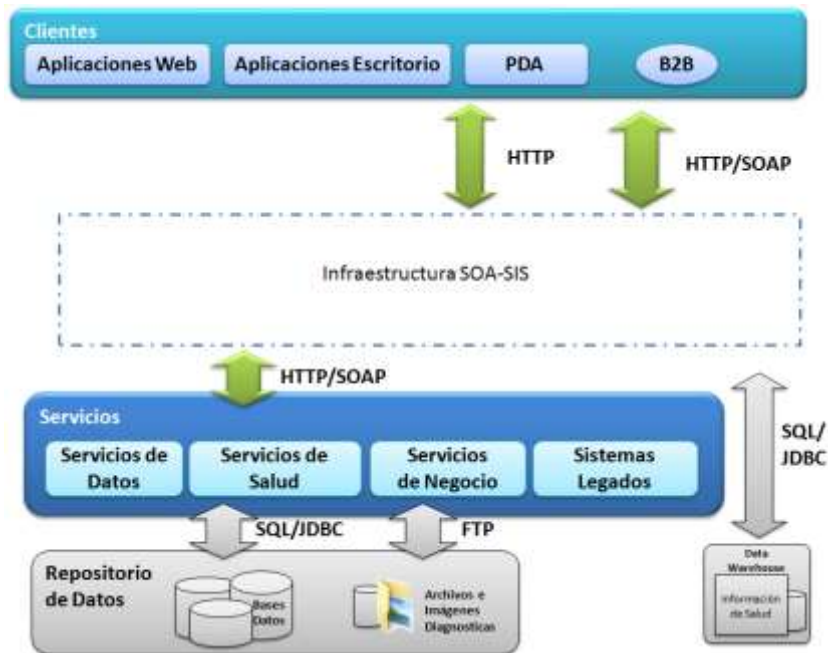


Figura 8. Módulos de la Arquitectura SOA-SIS.

A continuación son descritos los módulos mostrados en la Figura 8.

### Módulo Clientes.

En este módulo están localizados todos los aplicativos que hacen uso de los recursos expuestos por la arquitectura. Estas aplicaciones pueden ser web, de escritorio o móviles y estar soportadas por cualquier plataforma tecnológica, para ser visualizadas por diversos dispositivos como Portátiles, Smartphone, así como Asistentes Digitales Personales (PDA) entre otros.

Este módulo hace énfasis en la construcción de aplicaciones SOA que puedan ser reutilizables, o que aprovechen la flexibilidad que ofrece la arquitectura para re definir sus procesos y hacer uso de los recursos existentes.

### Módulo Servicios.

Este módulo especifica los servicios que son utilizados para que los activos existentes puedan ser expuestos en el ESB y utilizados por el módulo de clientes de la arquitectura. Son identificados cuatro tipos de servicios siguiendo las recomendaciones de los organismos internacionales (HSSP, 2009):

- **Servicios de Datos:** proveen las interfaces necesarias para realizar las operaciones CRUD (Create, Read, Update and Delete) en los repositorios de datos, los cuales, pueden ir desde información contenida en Bases de datos así como documentos que contienen gran cantidad de información como es el caso de las imágenes diagnósticas.
- **Servicios de Salud:** estos soportan los servicios específicos del ámbito salud. Proveen todos los mecanismos para realizar operaciones sobre los recursos expuestos en la arquitectura.



- **Servicios de Negocio:** son todos aquellos servicios de propósito general que dan soporte a los procesos de negocios.
- **Sistemas Legados:** expone todos los servicios extraídos de los Sistemas existentes. Estos permitirán la utilización de algunos de sus recursos y exponerlos en la arquitectura. Esto con el fin de aprovechar los procesos ya definidos e incorporarlos como activos reutilizables.

### **Módulo Repositorio de Datos.**

En este módulo son ubicados todos los recursos sistemas de almacenamiento existentes que proveen algún tipo de información importante. Entre estos encontramos activos como Bases de Datos y Repositorios de archivos (documentos, imágenes. entre otros ver figura 9).



**Figura 9. Componentes del Módulo de Repositorio de Datos.**

- **Bases de Datos:** empleadas por los diversos sistemas de información dentro de la arquitectura empresarial. Estas son caracterizadas por estar implementadas bajo diversas estructuras y motores de bases de datos.
- **Archivos e Imágenes Diagnósticas:** estos contienen todo tipo de documentos electrónicos que son necesarios de compartir en todo el entorno empresarial. Están en diferentes formatos y contienen información de diversa naturaleza. Estos están disponibles mediante servidores de archivos.

### **Módulo Bodega de Datos.**

Este módulo obedece a reglas y especificaciones directas del ESB para el almacenamiento de información mostrado en la figura 10.



**Figura 10. Módulo de Bodega de Datos.**

En él están expuestas las facilidades para el almacenamiento de datos en un contenedor que habilita las capacidades de inteligencia de negocios (IB), obedece a las reglas expuestas en el Módulo de Infraestructura SOA-SIS.

Este módulo busca dotar a la arquitectura de la capacidad para hacer un adecuado almacenamiento y tratamiento de la información, de tal manera que a futuro pueda contarse con un sistema de almacenamiento robusto y eficiente, que contenga información coherente.





## Módulo Infraestructura SOA-SIS.

El Bus de Servicios Empresariales, para la arquitectura propuesta, ofrece diversas capacidades que lo hacen el componente principal de la infraestructura. En este módulo también están las reglas para la orquestación de servicios que darán soporte a los procesos de negocio de la organización de salud. Expone las facilidades que ofrece la capa de Presentación de recursos para hacerlos disponibles para el módulo de clientes.



Figura 11. Módulo de Infraestructura SOA-SIS.

- **Bus de Servicios Empresariales:** es el componente troncal de la infraestructura. A través de la arquitectura empresarial es el encargado de exponer y mantener disponibles los recursos para todos los sistemas que componen la arquitectura. Entre las funciones que cumple este componente se encuentran:
  - **Motor de Enrutamiento:** direcciona todos los mensajes provenientes de los sistemas origen hacia los sistemas de destino adecuados. Esta función obedece a reglas definidas para la operación y comportamiento del ESB. Soporta diferentes modos de operación, de tal manera que sea flexible en la creación de controles y políticas de encaminamiento de mensajes así como el tratamiento de excepciones.
  - **Parser<sup>16</sup> HL7:** permite hacer un reconocimiento de los mensajes que llegan de los diversos sistemas de información que son agregadas a la arquitectura. Verifica los servicios, de tal manera que sean reconocibles por el ESB y el resto de la arquitectura.
  - **Transformador:** permite llevar los mensajes provenientes de un determinado sistema, el cual tiene un formato específico a otro formato basado o no en XML (como es el caso de HL7), de tal manera, que las aplicaciones origen y destino manejen un mismo lenguaje, o que la respuesta a una solicitud la reciba en un formato que sea entendible por las dos partes.
  - **Secuenciador:** define la lógica entre *parser*, enrutador y transformador, los cuales, deben operar para la recepción adecuada de mensajes, así como en su transformación de formato, para que posteriormente sea direccionado al sistema de información destino.
- **Servicios de Infraestructura:** este elemento contiene los servicios básicos que son empleados para las de seguridad, administración de excepciones, autenticación, control de acceso y otros servicios de administración de procesos empresariales. Este elemento es

<sup>16</sup> Analizador Sintáctico.

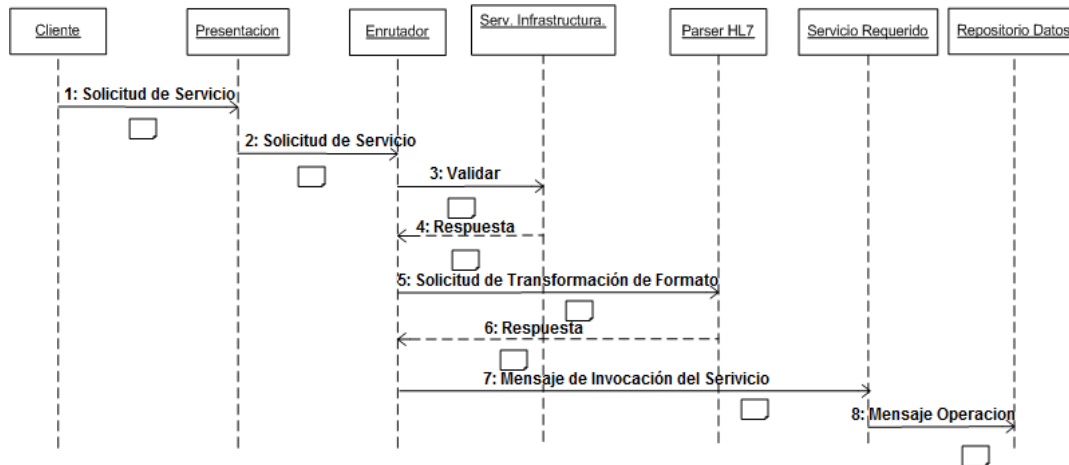


incluido como parte fundamental de la arquitectura SOA-HIS en atención a las recomendaciones del proyecto HSSP.

- **Control y Adaptación:** Permite el registro de la información pertinente a la ejecución de los procesos de negocio en estructuras de bodegas de datos.
- **Orquestación de Servicios:** este elemento especifica muchos de los servicios que deben ser orquestados para su definición, es decir, que hacen uso de otros servicios más básicos para su operación. Estos servicios orquestados definen funciones específicas, las cuales, estarán disponibles en la capa de presentación para su composición dentro de aplicaciones del módulo clientes. Cabe resaltar que, no se está haciendo alusión a reglas de negocio o especificaciones de operación del negocio, sino de definición de servicios de grano grueso que posteriormente apoyarán la definición de procesos de negocio.
- **Presentación:** este subcomponente acondiciona todos los recursos y funcionalidades disponibles en el Bus de Servicios Empresariales. Define los formatos de datos y todo tipo de información que será presentada al módulo de Clientes. De esta manera, el módulo clientes interactúa directamente con este componente para utilizar los recursos expuestos en el ESB, o directamente para acceder a otros recursos de más bajo nivel, como es el caso de algunos servicios específicos de acceso a datos que están definidos.
- **Administración:** el ESB debe contar con un subcomponente de Administración, el cual, debe permitir las funcionalidades como monitoreo y control. De igual manera, es de suma importancia la administración de estadísticas y errores, que permitan el establecimiento de alertas para un buen manejo de la arquitectura en general.

## **2.4. ARQUITECTURA COMPUTACIONAL (MODELO DINÁMICO).**

La Arquitectura garantiza la comunicación e interacción entre los componentes que la conforman, la figura 12 representa la llamada a un proceso que recorre los módulos de la arquitectura. Posteriormente, en la definición de protocolos y considerando el ESB seleccionado será presentado en detalle la comunicación al interior de los módulos con los respectivos protocolos empleados para tal fin.



**Figura 12. Comunicación de componentes**

El cliente puede llamar un servicio que puede ver (servicio virtual) y a través del ESB consumir el servicio real siguiendo los pasos que serán detallados. Para efectos prácticos solo es considerada la comunicación desde el Modulo Clientes hasta el Modulo repositorio de datos y empleando solo algunos sub componentes (como el secuenciador que interviene a lo largo del establecimiento de mensajes).

1. El cliente envía un mensaje (el especifica la naturaleza de su contenido y la del mensaje respuesta, XML, HL7 u otro) al subsistema de presentación enviando los parámetros necesarios para realizar dicha solicitud.
2. El subsistema de Presentación pasa la solicitud al ESB quien emplea el motor de enrutamiento para seleccionar el servicio de infraestructura (Modulo servicios de infraestructura) que debe ejecutar (de Acceso y Permisos de Usuarios validar u otro de ser necesario según la naturaleza de la solicitud).
3. El subsistema de servicios de infraestructura envía un mensaje al enrutador sobre la validación del recurso a consumir.
4. Si el proceso requiere trasformación de mensajes HL7, El enrutador le envía un mensaje al Parser HL7 que es el encargado de esta tarea.
5. El Parser HL7 le envía una respuesta con el mensaje en el formato solicitado al enrutador.
6. El enrutador envía un mensaje al servicio requerido (empleando el motor de orquestación si el servicio lo requiere).
7. El servicio requerido recibe el mensaje y es puesto en marcha.
8. El Servicio en ejecución requiere acceder a al repositorio de datos, enviando el mensaje apropiado (por ejemplo la ejecución de un SQL).

En otras arquitecturas orientadas a soluciones específicas, donde son pocos y bien definidos los procesos de negocio, son omitidos los servicios de infraestructura de red y son definidas capas de lógica de aplicación, este no es el caso del HSLV donde son múltiples los procesos y es deseable una completa gobernabilidad de los mismos.



## **2.5. JUSTIFICACIÓN DE LA ARQUITECTURA, BLOQUES Y COMPONENTES.**

Es importante hacer una separación de la Lógica de Negocio de la estructura de datos; con este fin, luego de extraer todos los repositorios de datos de los sistemas de información existentes. Deben especificarse y analizarse detalladamente para posteriormente, implementar la interfaz más adecuada para su acceso. Debido a la complejidad de estos sistemas de almacenamiento debe tenerse sumo cuidado en su manipulación, ya que, contienen demasiados procesos almacenados que correlacionan tablas, donde, determinada consulta ejecutada produce un impacto en alguna estructura o en el desempeño la misma; por ende, debe estudiarse detenida y cuidadosamente, para tener claros los procedimientos almacenados y la lógica de funcionamiento de los motores de bases de datos.

Muchas de estas estructuras de datos contienen parte de la lógica de negocio, por ende, debe tratarse en lo posible de evitarlo, ya que estaría ligando los datos a la funcionalidad del negocio, complicando la fácil manipulación de información. Las estructuras de almacenamiento de datos, en todos los sistemas de información, son diferentes. Aparte de la diversidad en la implementación tecnológica también es observado en el enfoque de su construcción.

De igual manera, existen documentos electrónicos que contienen gran cantidad de información. En el ámbito Salud existen documentos como las Imágenes diagnósticas, cuyo formato, aunque visual, tiene una especificación especial, que contiene gran cantidad de información. Cuando los aplicativos son propietarios hacen uso de mecanismos para compartir recursos en red, de tal manera que puedan ser accedidos desde diversos puntos de localización dentro de la infraestructura física. Como consecuencia es necesario emplear un medio adecuado, y alternativas que soporten protocolos óptimos para su acceso, como FTP u otros.

En este punto es visible la gran necesidad de separar toda esta información, de manera que pueda ser reutilizada de acuerdo a la especificación y estudio hecho, para hacer un manejo adecuado de ella. Es importante, como es mencionado, identificar todos aquellos factores como procedimientos almacenados y disparadores que guardan cierta lógica del negocio, ya que, el objetivo es sacarla de este contexto.

Una vez disponibles estos recursos, son definidas las interfaces de conectividad, que, como es presentado en la Figura 7, deben establecerse de acuerdo al tipo de información a manejar.

Para el caso de Bases de datos, requiere del uso de conectores que provean la comunicación necesaria para ejecutar las Sentencias SQL específicas de cada motor de almacenamiento. Así mismo, deben emplear protocolos de transferencia de archivos que permiten el manejo de grandes documentos en formato digital estableciendo los medios para acceder a la información contenida en los repositorios de datos.

El módulo de servicios pretende especificar todos los servicios aquellos que permitan acceder a los recursos disponibles en los repositorios de Datos. De esta manera, expone y los hace accesibles para el resto de la arquitectura.



La definición de estos servicios debe ser cuidadosa, de manera que representen de forma adecuada el manejo que debe dársele a estos recursos. La finalidad es entonces, llevarlos a un alto grado de manejabilidad y disponibilidad en toda la arquitectura.

Estos servicios serán definidos bajo estándares abiertos, que permitan su exposición y localización, y que sean entendibles por diferentes plataformas tecnológicas. Para esto, organizaciones internacionales definen los lineamientos para la implementación de servicios así como las recomendaciones para seguridad (HSSP, 2009).

En el caso de los servicios para acceso a datos, es importante la utilización de marcos de trabajo de abstracción de datos como los son los proveedores de mapeo Objeto-Relacional (Object-Relational mapping, ORM) como Hibernate, DAO o TopLink como herramienta para el uso del API de Persistencia para Java (Java Persistence API, JPA); los cuales permite hacer una abstracción de ER a Objetos, que facilita su entendimiento y su posterior programación, permitiendo concentrarse en los procesos de negocio.

Para los servicios extraídos de los Sistemas legados, son especificados para permitir el entendimiento de sus operaciones y la forma en que podrá ser utilizado. De esta manera, de las operaciones contenidas en estos servicios web, permitirá abstraer algunas de las funcionalidades en cuanto al acceso a datos u otros recursos contenidos en los sistemas de información existentes.

Están separados los servicios de Negocio de los servicios específicos de salud, con el objetivo de maximizar la reutilización de estos servicios. Estas, aparte de prestar servicios sociales, tienen un carácter de empresa que introduce algunas operaciones que son comunes en muchos mercados verticales<sup>17</sup>. De esta manera, especificar qué servicios son únicos de salud da una ventaja debido a que reduce y deja ver claramente su funcionamiento como organización de salud, mientras que, los servicios de negocio, soportan algunas transacciones necesarias para la prestación de estos servicios.



**Figura 13. Clasificación de Servicios.**

Es importante entonces la separación, debido a que, los protocolos en la prestación de servicios cambia de acuerdo a políticas gubernamentales y otros factores, mientras que, muchas operaciones de negocio varían pero en el costo económico que representa el servicio u otros que son diferentes por las variaciones producidas en los servicios únicos de salud. De esta manera, son separados estos servicios de forma que provea una abstracción adecuada de funcionalidad que brinde a la arquitectura la flexibilidad necesaria para responder a los cambios de todos los niveles.

---

<sup>17</sup> Mercados donde la oferta y la demanda de determinada categoría de producto interactúan en un punto.



El ESB dentro de la arquitectura propuesta habilita la infraestructura que requiere SOA para soportar los procesos de negocios: proveer una manejabilidad en todos los niveles de los servicios disponibles así como operar e integrar entornos heterogéneos. Estos requerimientos van más allá de la capacidad de enrutamiento y transporte.

El módulo de seguridad es un componente bastante funcional, debido a que soporta estándares de seguridad para entornos de computación, permitiendo una administración de las reglas y políticas de autenticación y confidencialidad, protegiendo la integridad de la información que es transportada sobre el bus y garantizando que la información llegue correctamente a su destino.

Además, ofrece posibilidades de administración, no solo de los recursos sino de sí mismo como parte de la arquitectura. Estos elementos traen consigo herramientas administrativas para monitorear y medir su desempeño así como el de otros recursos disponibles en la arquitectura. De esta manera, es importante la centralización de funcionalidades como las que se describen anteriormente, que permiten hacer una administración globalizada, y definir las políticas necesarias según la experiencia obtenida y poder hacer una buena gestión.

Los elementos de la arquitectura deben soportar protocolos de comunicación que garanticen un buen desempeño de la infraestructura permitiendo la integración de sistemas bajo una sola infraestructura. De esta manera, proveen niveles de interoperabilidad que van de acuerdo a la implementación de estándares adecuados y al manejo de la transformación de formatos.

Existe un componente importante conocido como Orquestación de Servicios. Esta funcionalidad permite al bus definir y publicar servicios más grandes o baja granularidad que representan alguna funcionalidad más compleja, de tal forma que pueda ser reutilizable en diversos contextos. Esta funcionalidad es importante, debido a que, puede trabajar con los servicios a distintos niveles, desde servicios muy específicos, hasta otros más complejos. Esto facilita, además, la administración de procesos de negocio, debido a que, está separada la lógica de los servicios con la lógica de procesos.

La capa de presentación expone todos los recursos en formatos entendibles; esto debido a que, por medio de esta capa, los futuros sistemas de información obtienen la información de forma adecuada, para desplegarla óptimamente. De esta manera, es provisto un camino que no requiere interactuar directamente con el bus, sino más bien, utilizar las funcionalidades realmente necesarias para la composición y desarrollo de otros sistemas.

El nivel de clientes está de manera apartada del bus, de tal forma que interactúe a través de la capa de presentación, quien es el encargado de proveer la información en sus formatos adecuados. En este nivel van todas las aplicaciones compuestas y lógicas de negocio. De esta manera dota de robustez a la arquitectura, separando incumbencias y haciendo que los cambios o variaciones sean exclusivos de cada ámbito y no un inconveniente común.

## **2.6. VISTA DE LA INGENIERÍA DE LA ARQUITECTURA.**

Es establecida a partir de JEE (ver sección A3 del Anexo A) como plataforma tecnología para el desarrollo de los nuevos aplicativos que soportaran la SOA solución, sin embargo en JEE existen



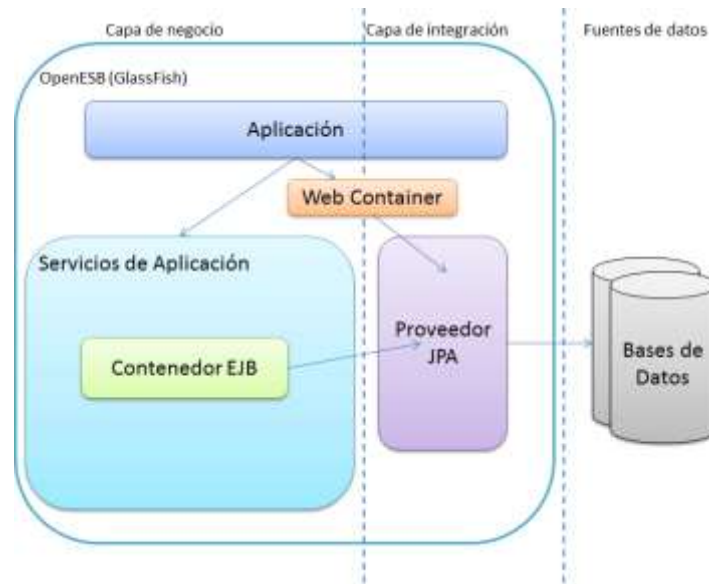
muchas tecnologías y es de imperante importancia definir la arquitectura de tecnológica que comprende muchos aspectos como:

- Establecer el rol de cada tecnología en nivel de negocio, por ejemplo definir cuándo emplear JMS cuando emplear RMI, BC, EJB etc.
- Cuando llegar a la capa de procesamiento la carga transaccional y cuando procesar en los motores de almacenamiento.
- Estándares de nombramiento de clases variables, paquetes, procesos almacenados etc.

Lo anterior lleva a definir una plantilla arquitectónica y una serie de documentos y códigos fuente que son el marco de trabajo de los programadores. La creación de todos estos componentes de negocio conlleva implícitamente a la conformación y establecimiento un framework propietario (Minoli, 2008).

El proceso formal para la obtención de la vista de la ingeniería de la arquitectura es la de definir varios prototipos de arquitectura sobre algún aplicativo prototipo similar a los reales, para ser estudiados mediante la prueba evaluación de escalabilidad e integración principalmente. Dado que el objetivo principal de este trabajo es el despliegue de una SOA sobre la plataforma JEE (como se discute en Anexo A.3) y con el animo de estandarizar la tecnología con las mejores enfoques tecnológicas existentes actualmente determina que la creación de los nuevos servicios web bajo una capa de negocio soportada por EJB y la capa de modelo soportada por JPA y en la capa de presentación Java Faces, como lo muestra la figura 14.

Lo anterior en atención a fortalecer la escalabilidad del sistema como es detallado en el Anexo C.4 en cuanto a la administración de conexiones (pool de conexiones), es deseable una capa de integración y aprovechando los recursos del ESB, son habilitadas las conexiones a la bases de datos de manera que estén disponibles para los servicios web que conforman la arquitectura dispongan de ellos de manera que sea administrable el acceso a datos (el anexo C.4 muestra como a partir de un Driver JDB puede ser configurado para crear un pool de conexiones administradas por el ESB). De esta forma pueden ser ofrecidos los recursos de acceso a datos desde el ESB de forma flexible y extensible.



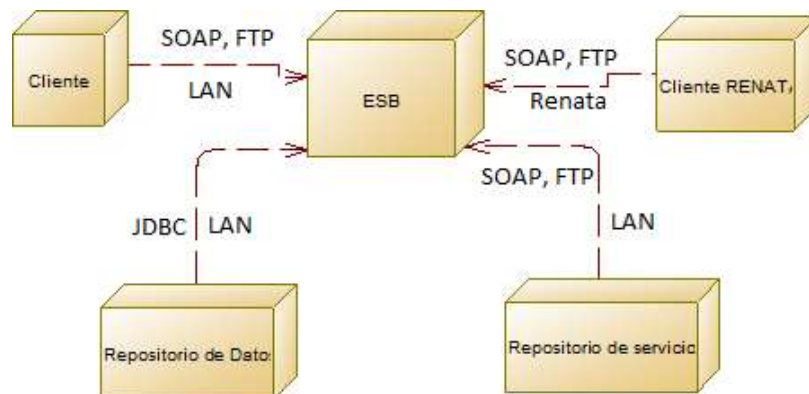
**Figura 14** Vista de Ingeniería de la Arquitectura solución.

La Figura 14 está dividida en tres partes principales con el fin de separar la distribución de los recursos que dispone para la creación de aplicaciones a partir de los sistemas existentes en el ESB. Los servicios de aplicación son establecidos mediante la construcción de EJB en el contenedor EJB como parte fundamental para el desarrollo de servicios web tanto para ser empleados en los servicios o como para que los EJB sean consumidos como un servicio web. Esto es altamente deseable ya que la capa de negocio estará compuesta de EJB y su estrecha relación con JPA para administrar la carga transaccional de los servicios web. El objetivo que persigue el diseño de esta API es el de no perder las ventajas de la orientación a objetos al interactuar con una base de datos (siguiendo el patrón de mapeo objeto-relacional).

## **2.7. VISTA TECNOLÓGICA DEL SISTEMA.**

El presente Diagrama de Despliegue muestra de manera gráfica los nodos que conforman el sistema, además los describe indicando la localización de las tareas en los nodos físicos. Cada nodo representa un recurso de ejecución como una computadora o servidor y está conectado con otros mediante un enlace de comunicación (Minoli, 2008). En este caso, la red será de tipo LAN, para el caso institucional.





**Figura 15. Vista Tecnológica. Modelo de Despliegue.**

Como puede verse en la Figura 15, el ESB como mediador de los recursos es un punto de convergencia de los componentes que conforman el modelo de despliegue. A continuación una descripción del hardware y tipo de red que redes a las que prestan los servicios cada elemento de la red sin considerar las tecnologías software sobre la que esta implementado el ESB (el tema será tratado en el siguiente capítulo).

- **ESB:** será instalado en un equipo de cómputo con características de servidor, con a lo sumo dos interfaces de red ya que este equipo hará parte de la Red de Tecnología Avanzada (RENATA) y la red LAN del HSLV, esto facilita un escenario de interoperabilidad entre la red global y la intranet.
- **Cliente Renata:** otros equipos suscritos a RENATA con capacidad de consumir servicios web a mediante el protocolo SOAP y de esta forma pueden acceder a los servicios de la plataforma de interoperabilidad que proveerá el HSLV a través del ESB.
- **Cliente:** son los clientes naturales del entorno empresarial del HSLV es decir que se encuentran en red de datos corporativa mediante SOAP, pueden ser aplicaciones o clientes o servidores de aplicación web.
- **Repositorio de Datos:** es el sistema distribuido de almacenamiento que actualmente está implementado en el HSLV que consta de aplicaciones en equipos estación (Dinámica Fox y .Net, Sagita, DataLab, Carestream) y un clúster de SQL Server (compuesto por 2 servidores).
- **Repositorio de servicios:** hace referencia a los servicios que se pueden obtener de los sistemas legados o de próximos sistemas a adquirir.

## **2.8. CONCLUSIONES.**

Este capítulo expone los enfoques metodológicos empleados para el establecimiento de una SOA en el marco del modelo de referencia de sistemas distribuidos y abiertos (RM-ODP). Define la metodología de referencia SOA como propuesta metodológica para la construcción de una SOA, empleada para el establecimiento y despliegue de la arquitectura solución, como está expuesto en el capítulo 3 de este documento. Además describe la arquitectura solución empleando la metodología HIS-DF, ella permitió el análisis y diseño de la misma.



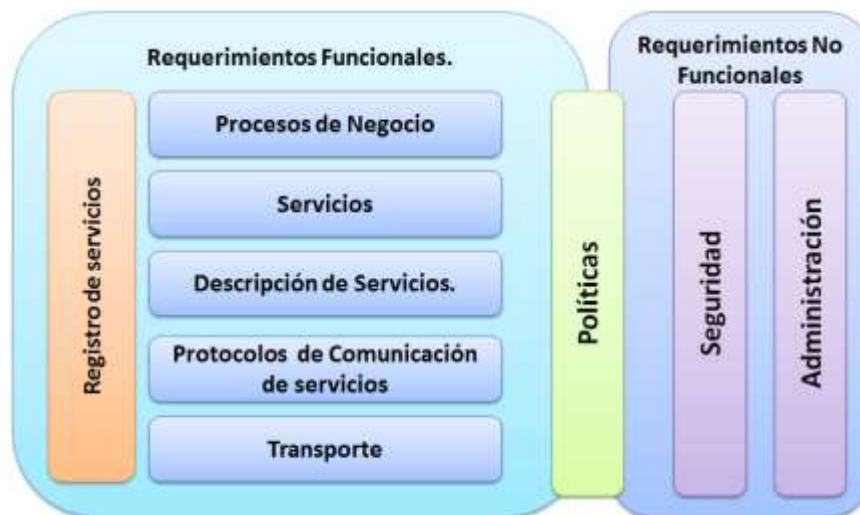
### **3. CAPÍTULO III. IMPLEMENTACIÓN DE LA ARQUITECTURA SOA.**

La importancia de la utilización de los buses de servicios empresariales reside en soportar la infraestructura adecuada para el despliegue de una SOA. Por esto es necesario establecer un mediador como los Buses de Servicios Empresariales que ayudan a aprovechar al máximo las potencialidades de la SOA. Tiene como objetivo evaluar las ventajas o beneficios que ofrece la utilización de estas herramientas que si bien son completas, son diversas y cumplen de una u otra forma la totalidad o parcialidad de los requisitos para el despliegue adecuado de una SOA.

Tomando en consideración las recomendaciones de SOA para SIS, descritas a partir del paso 5 propuestos en la especificación HSSP, este capítulo estudia los ESB candidatos para la arquitectura solución y las herramientas tecnológicas a emplear.

#### **3.1. OPCIONES TECNOLÓGICAS PARA SOA.**

En esta sección son discutidas las tecnologías maduras y emergentes que hacen parte de la implementación técnica para soportar una SOA. La Figura 16 muestra unos niveles así como las posibles tecnologías que lo soportan. A continuación son tratadas estas tecnologías para establecer sus ventajas y desventajas con el fin de determinar cuál puede soportar las exigencias de la arquitectura propuesta en el capítulo 2.



**Figura 16. Tenologías para SOA.**

##### **3.1.1. NIVEL DE TRANSPORTE.**

Este nivel está relacionado con los mecanismos que permiten llevar las solicitudes de servicios desde los consumidores hacia los proveedores y las respuestas de estos a los consumidores. Entre estos mecanismos encontramos tecnologías como: HTTP (Hypertext Transfer Protocol Secure), JMS (Java Message Service), SMTP (Simple Mail Transfer Protocol), HTTPR (Reliable HTTP). También existen algunas tecnologías emergentes como WS-Reliable Messaging y BEEP. A continuación son descritas cada una de ellas así como sus ventajas y desventajas.



## **Hypertext Transfer Protocol-HTTP.**

Las ventajas que ofrece este protocolo son:

- Ampliamente adoptado ya que todas las organizaciones que cuentan con servidores Web tienen implementado HTTP. Además de esto, los clientes que utilizan navegadores web que soportan este protocolo, proveen la infraestructura necesaria que hacen que esta tecnología este ampliamente disponible.
- Desplegado para diversos tipos de sistemas, incluyendo dispositivos de computación no convencional, lo que lo hace portable y adaptable a diversos entornos.
- La mayoría de las empresas permiten que este protocolo pase libremente a través de firewalls, presentando menos barreras para extender el uso empresarial de HTTP como transporte para Servicios Web.

Sin embargo, este protocolo no fue diseñado para transportar datos de aplicaciones. Por esta razón, no es un protocolo de alta fiabilidad.

## **Java Message Service-JMS.**

La mediación de mensajes es una elección popular para acceder a sistemas empresariales existentes de forma asíncrona. La mensajería de comunicación es bajamente acoplada comparada con tecnologías como el Método de Invocación Remoto (Remote Method Invocation, RMI) y Llamada a Procedimiento Remoto (Remote Procedure Call, RPC). Quien envía el mensaje no necesita saber absolutamente nada sobre el destinatario. Otra ventaja es que el sistema origen y el destino no necesitan estar disponibles al mismo tiempo para poder comunicarse.

JMS provee caminos comunes para la creación, envío, recepción y lecturas de mensajes. Este estándar soporta dos modelos de mensajería: punto a punto y publicación/suscripción. Provee un transporte necesario de una forma más fiable que soluciones como HTTP, sin embargo como es basado en Java, no es totalmente accesible para sistemas que no están soportados por esta tecnología.

## **Blocks Extensible Exchange Protocol-BEEP.**

Es un protocolo de aplicación genérico para interacciones asíncronas par a par utilizando conexiones TCP/IP. A diferencia de HTTP, no tiene nociones de un servidor o cliente sino que, establece una sesión de comunicación basada en los mensajes cuando existe una solicitud para conectarse con un proveedor. Además, permite administrar la autenticación y los mensajes de transporte. Es orientado a la conexión, ya que la comunicación inicia y termina con cada interacción (BEEP, 2009).

Como fue descrito anteriormente, existe una amplia gama de protocolos que permiten el transporte de mensajes a través de una arquitectura. Sin embargo, como fue expuesto, el protocolo **HTTP** por su simplicidad y utilización en la industria además por las características de los ESB estudiados en la sección anterior, es empleado como protocolo estándar para el nivel de transporte, de tal manera que cumple con los requisitos establecidos para la arquitectura SOA-SIS propuesta.



### **3.1.2. NIVEL DE PROTOCOLOS DE COMUNICACIÓN DE SERVICIOS.**

Estos describen y definen las tecnologías y estándares necesarios para proveer mecanismos de transporte entre servicios integrados.

#### **Simple Object Access Protocol-SOAP.**

Le permite a un consumidor de servicio llamar a un proveedor de servicio remoto. Este mensaje tiene un formato XML. Está diseñado para ser simple y extensible también provee un marco de trabajo que describe contenidos de mensajes e instrucciones de procesos y un conjunto de reglas de codificación opcionales para representar tipos de datos definidos.

Es un protocolo de transporte independiente a pesar de que utilice HTTP como uno de sus soportes. Es por tal motivo que determina la utilización de SOAP sobre HTTP como estándar para el nivel de protocolos de comunicación de servicios, que responde a las necesidades expuestas por la arquitectura propuesta.

### **3.1.3. NIVEL DE DESCRIPCIÓN DE SERVICIOS**

Uno de los principales beneficios de los servicios web es permitir que las arquitecturas sean bajamente acopladas. Para lograr este objetivo, el proveedor de servicios y el consumidor deben estar lo más independientes posible. Una descripción estructurada del servicio es la clave para habilitar esa independencia. Los servicios pueden ser proveídos sin necesidad de que el proveedor o el consumidor estén pendientes de la plataforma tecnológica del otro. Esta descripción de servicios comprende dos niveles:

- Descripción operacional de servicios (XML y WSDL).
- Descripción completa de servicios.

#### **Lenguaje Extensible de Marcado (eXtensible Markup Language, eXML).**

Es la sintaxis de facto utilizada para el intercambio de datos de mensaje entre el consumidor y proveedor de servicios web. Permite un lenguaje de etiquetado personalizable con tags definidas en una Definición de Tipo de Documento (Document Type Definition, DTD) o XML Schema.

XML permite la representación de datos en un formato estandarizado y estructurado. Provee la sintaxis de un lenguaje pero no define el significado asociado con el dato.

Es una manera estándar de poner información en un formato que pueda ser procesada e intercambiada a través de diferentes dispositivos hardware, sistemas operativos aplicaciones de software y la web. Permite separar la representación de los datos de la presentación y formato de los datos para ser visualizados en un navegador u otro dispositivo.

#### **Lenguaje de Descripción de Servicios Web (Web Service Description Language, WSDL).**



Especifica las características operacionales de un servicio web utilizando un documento XML. Ofrece una forma estándar para describir abstractamente las operaciones y mensajes: definición de Interfaces de Servicio.

Un documento WSDL está definido en uno o más archivos físicos de acuerdo a la implementación. Como un requerimiento fundamental para la implementación de WS, WSDL requiere publicar el contrato de descripción de interfaces para que otros servicios la invoquen.

### **Negocios Electrónicos Mediante XML (Electronic Business Using XML, ebXML).**

Provee una suite modular de especificaciones que habilitan las empresas para conducir sus negocios sobre Internet. Es un método estándar para intercambiar mensajes de negocio, comunicar datos en términos comunes y definir y registrar procesos de negocio.

De acuerdo con las características descritas para los protocolos mencionados anteriormente, este trabajo de grado escoge **WSDL** como opción tecnológica más viable, el cual soporta los requerimientos de la arquitectura propuesta.

#### **3.1.4. NIVEL DE SERVICIO.**

Representa el software implementado que puede ser localizado e invocado basado en la publicación de la interfaz de descripción WSDL.

Para este nivel, así como la temática del presenta trabajo de grado, existen dos tecnologías para la implementación de servicios: .NET y JEE. Debido a que la arquitectura propuesta esta soportada por tecnologías basadas en software libre, así como en estándares abiertos, la implementación de los servicios es bajo JEE y sus especificaciones.

#### **3.1.5. NIVEL DE PROCESOS DE NEGOCIO.**

Permite crear y definir procesos y flujos de trabajo complejos. Los procesos son compuestos desde las operaciones provistas por los servicios web que pueden ser anidadas o secuenciales de acuerdo a los requerimientos de negocio.

Utilizando un enfoque de modelado consistente simplifica la comunicación entre todas las partes involucradas en los procesos de negocio. Este enfoque basado en estándares abiertos permite que las actividades del proceso estén bajamente acopladas del proceso mismo, minimizando el tiempo y los esfuerzos para implementar los cambios de acuerdo a los requerimientos y entorno.

### **Web Service Business Process Execution Language-WS-BPEL.**

Provee un estándar para describir procesos de negocio que están basados en servicios web. Estándares existentes como WSDL no proveen facilidades que son necesarias en los protocolos de negocio complejos, tal como la descripción de comportamiento que depende de los datos enviados entre servicios, descubrimiento y manejo de excepciones así como coordinación entre las partes involucradas en un proceso complejo y de larga ejecución.



Existen tecnologías como Web Service Flow Language-WSFL la cual cumple con funcionalidades y características similares descritas para WS-BPEL, sin embargo, por ser de naturaleza propietaria no son descritas con detalle en este trabajo.

De esta manera, BPEL es el lenguaje de ejecución de procesos que soporta las características y funcionalidades que requiere la arquitectura SOA-SIS propuesta en este trabajo. De esta manera, BPEL queda definido como estándar tecnológico para el nivel de procesos de negocio.

### **3.1.6. NIVEL DE REGISTRO DE SERVICIOS.**

Permite a los proveedores de servicios publicar la definición de los servicios que ofrecen haciendo uso de WSDL para que los consumidores puedan tener información sobre la disponibilidad de los mismos.

Existen algunas tecnologías como WS-Inspection, que cumplen funcionalidades como las descritas en el catálogo para la Descripción Universal, de Descubrimiento e integración (Universal Description, Discovery and Integration, UDDI), sin embargo, por ser de naturaleza propietaria no son descritas en detalle en este trabajo.

En este nivel, tecnologías como UDDI, permiten cumplir la funcionalidad requerida para la arquitectura en el ámbito de publicación de servicios web. De esta manera, UDDI es para este trabajo el estándar para el registro de servicios.

### **3.1.7. NIVEL DE POLÍTICAS.**

Este nivel cubre aspectos tanto funcionales como de calidad de servicio, como lo muestra la figura 16. Para este nivel, presenta el marco de trabajo de WS-Policy que permite describir los requerimientos y las capacidades asociadas a los servicios. Define la gramática para expresar las políticas, así como la forma en que adjuntan a los servicios web.

### **3.1.8. NIVEL DE SEGURIDAD.**

El estándar de seguridad para servicios web WS-Security define un modelo para servicios web comprensible, el cual soporta, integra y unifica muchos de los modelos de seguridad, mecanismos y tecnologías de forma que habilite una variedad de sistemas para “interoperar” seguramente en una plataforma.

Cubre un gran conjunto de especificaciones que recoge tecnologías de seguridad para autenticación, autorización, privacidad, integridad, confidencialidad, canales de comunicación seguros, así como auditoria dentro de un amplio espectro de aplicaciones y topologías de negocio. Provee un marco de trabajo extensible, flexible, y maximiza las inversiones existentes dentro de una infraestructura segura. Está fundamentada en los principios de tecnologías como SOAP, WSDL, Encriptación XML y SSL/TLS.



Provee un modelo de seguridad para mensajes. De esta manera, esta tecnología permite el establecimiento y cumplimiento de los requerimientos de seguridad de la arquitectura propuesta; por este motivo, WS-Security se toma como tecnología adecuada para el nivel de Seguridad.

### **3.2. BUS DE SERVICIOS EMPRESARIALES – ESB.**

Existen en el mercado un amplio número de ESB, algunos de naturaleza propietaria de probada robustez en otros entornos empresariales como WebSphere ESB (IBM, 2010) de IBM, que adicionalmente ofrece un entorno integrado a la metodología SOMA, para la construcción y desarrollo de aplicaciones orientadas a servicios. Oracle por su parte ofrece su producto: Oracle Service Bus (Oracle, 2010), con los componentes de JDeveloper<sup>18</sup> para la implementación de aplicaciones sobre una SOA. Sin embargo al no ser soluciones de libre distribución quedan fuera del marco de referencia para aplicarles un estudio detallado. No obstante las características mínimas deseables de un ESB son abstraídas de estos proyectos y enumeradas a continuación para considerarlas en la selección y debido a su relevancia en el proceso de selección de un ESB de libre distribución.

#### **3.2.1. CRITERIOS DE SELECCIÓN.**

Los criterios descritos a continuación son considerados suficientes para garantizar que el ESB ofrezca alta “escalabilidad” a la arquitectura.

##### **3.2.1.1. Enrutamiento de Mensajes y Servicios.**

Que administra comunicación entre los componentes de la arquitectura, actuando como un mediador que normaliza los mensajes que intercambian los componentes del Bus, permitiendo un acoplamiento débil entre ellos.

Es considerado de **ALTA** relevancia en la selección, por ser indispensable en la exposición y ejecución de los recursos que interactúan con la arquitectura.

##### **3.2.1.2. Herramienta de Administración.**

Permite la administración en general del ESB, como por ejemplo: la configuración, despliegue, nombrado y flujo de servicios.

Considerado de **ALTA** relevancia dado que esta característica facilita la administración.

##### **3.2.1.3. Soporte de Protocolos de Transporte Abiertos.**

Protocolos de alta disponibilidad tales como TCP/IP o FTP, entre otros. Es importante porque facilita el establecimiento de servicio de alta disponibilidad y es un medio para la interacción con otros sistemas basados en ellos.

---

<sup>18</sup> Entorno de desarrollo integrado desarrollado por Oracle Corporation.



De **ALTA** relevancia dado que provee los mecanismos para intercambio de procedimientos y datos, con los componentes que interactúan con la arquitectura.

#### **3.2.1.4. Transformación de los Datos.**

Capacidad para hacer una transformación de datos adecuada (mapeo de los datos de un formato a otro).

De relevancia **BAJA**, dado que nos interesa enfocarnos en formatos de mensaje HL7.

#### **3.2.1.5. Manejo Sincrónico y Asincrónico de los mensajes.**

El ESB debe permitir el control de ejecución síncrono y asíncrono de procesos. Esta característica es de **ALTA** relevancia porque es necesario considerar los tiempos que requieren para ejecución algunos servicios, en particular los de procesamiento de datos y adquisición de imágenes diagnósticas.

#### **3.2.1.6. Basado en Estándares.**

Es importante considerarlo ya que favorece el mantenimiento, adaptación, mejoramiento y crecimiento de las aplicaciones así como la integración con otros sistemas, por lo tanto es considerado de **ALTA** relevancia.

#### **3.2.1.7. Seguridad.**

En cuanto a protocolos de comunicación seguros como HTTPS, manejo de autenticación y autorización de usuarios, seguridad en los Servicios Web, etc.

Es considerado de relevancia **ALTA** dado que separa la relación o interacción de sistemas con otras empresas.

#### **3.2.1.8. Definición de Servicios.**

Que permita definir servicios mediante interfaces que abstraen al cliente de la implementación de éstos.

Es necesario que el ESB soporte esta característica dado que el departamento de TI del HSLV no está familiarizado con el desarrollo de Software con la Orientación a Servicios y esto permite reducir el impacto de apropiación a la metodología y mejoraría el nivel de producción, por lo tanto es de **ALTA** relevancia en las consideraciones.

#### **3.2.1.9. “Extensibilidad”.**

Que sea fácil la adición de nuevos componentes. Esta característica está entre las recomendaciones del HSSP cuando establece que debe mantenerse un equilibrio entre tecnologías emergentes y maduras. Considerado de **ALTA** relevancia.





#### **3.2.1.10. Orquestación de Servicios.**

De esta manera es posible definir procesos más complejos en el ESB basado en un lenguaje estándar que nos permita el control de la ejecución, por lo tanto es de **ALTA** relevancia.

#### **3.2.1.11. Calidad de Servicio.**

Que permita el establecimiento de lineamientos de QoS para diferenciar el tráfico que soportan determinados procesos y garantizar el óptimo uso de los recursos. Es considerado de **ALTA** relevancia.

#### **3.2.1.12. Soporte Técnico.**

Que se puedan encontrar en la comunidad de software libre la actividad suficiente para resolver dudas y problemas, además de la adecuada documentación. De **BAJA** relevancia ya que se considera que los protocolos a emplear sean abiertos es de alta relevancia.

#### **3.2.1.13. Entorno de Desarrollo Integrado (IDE).**

Disponer de un IDE que represente los componentes de la arquitectura ayuda al personal encargado de desarrollar sistemas y componentes en la arquitectura, aunque existen muchos en el mercado de libre distribución que puede ser empleado y en algunos casos ampliar sus capacidades por medios de extensiones para que soporten componentes de una SOA. Por lo tanto se considera de **BAJA** relevancia.

#### **3.2.1.14. Versión Estable.**

Para garantizar la correcta funcionalidad del ESB, es considerado de relevancia **ALTA**.

#### **3.2.1.15. Soporte de Mensajes HL7.**

Existen varias librerías en diversos lenguajes de programación para proveer la transformación hacia y desde mensajes HL7, transmisión y recepción de los mismos. Este es un criterio que se espera que soporte el ESB dado que es el estándar para brindar interoperabilidad semántica entre sistemas de información, por ende se considera de **ALTA** prioridad.

### **3.3. COMPARACIÓN DE ESB BASADOS EN SOFTWARE LIBRE.**

Según las características deseadas en todo ESB son analizados los siguientes productos sin considerar ningún orden particular:



- ChainBuilder.
- Apache ServiceMix.
- Apache Synapse
- JBossESB
- Petals ESB
- WSO2 ESB
- Mule ESB
- OpenESB
- FUSE ESB
- JeeESB
- Celtix

La siguiente tabla muestra un resumen de los principales ESB que son mas ajustados a los criterios de selección basado en la información detallada en el anexo A, es importante aclarar que los ESB basados en tecnologías privativas no fueron tenidos en cuenta para esta comparación.

**Tabla 4. Criterios de selección del ESB.**

Herramientas Consideradas	Criterios de Selección														
	Enrutamiento	Administración.	Protocolos de Transporte.	Transformación.	Mensajes Sincrónicos y Asincrónicos.	Basado en Estándares.	Seguridad.	Definición de Servicios.	Extensibilidad.	Orquestación de Servicios.	Calidad de Servicio.	Soporte Técnico.	Entorno de Desarrollo. Integrado.	Versión Estable.	Soporte para HL7.
<b>OpenESB</b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<b>ChainBuilder ESB</b>	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
<b>Fuse ESB</b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
<b>PEtALS</b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✗
<b>JBoss ESB</b>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗
<b>Apache Synapse</b>	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✗	✗	✓	✗
<b>Apache ServiceMix</b>	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	✓	✓	✓
<b>WSO2 ESB</b>	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✓	✓	✗	✓	✗
<b>Mule ESB</b>	✓	✗	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓
<b>JeeESB</b>	✓	✗	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
<b>Celtix</b>	✓	✗	✓	✓	✓	✓	✓	✗	✓	✓	✗	✓	✓	✓	✓

Algunas soluciones cumplen de una forma particular los requerimientos de la arquitectura, factor que influye de forma determinante en la selección de alguno de ellos, ver tabla 4.



### **3.4. SELECCIÓN DEL ESB.**

Herramientas como JeeESB carecen de elementos suficientes para responder a los requerimientos de una SOA para SIS. A pesar de ser basado en software libre y de soportar algunos protocolos de transporte y de servicios, posee características muy deficientes en enrutamiento de mensajes, lo que impide un alto desempeño como mediador entre sistemas dentro de un entorno heterogéneo. Tampoco cuenta con un IDE que agilice el desarrollo de sistemas que respondan a los continuos cambios del sector. Otro inconveniente, es que no soporta HL7, lo que implicaría evaluar una tecnología compatible como complemento para habilitar esta capacidad. Con estas apreciaciones, es descartada la utilización de este ESB, debido a la falta de soporte de características como sincronismo y asincronismo de mensajes así como la definición y orquestación de servicios.

Celtix ESB carece de la capacidad para la transformación de mensajes. Esta característica es importante en un entorno heterogéneo, lo que facilita que los SIS que hacen parte de este compartan información en un formato entendible. Sin embargo, soporta estándares como JBI que lo hace extensible e implementa JMS que le da la capacidad de sincronismo y asincronismo; a pesar de ello, no es una herramienta robusta en el enrutamiento de mensajes y no cuenta con soporte para HL7 lo que implicaría un estudio adicional en evaluación de tecnologías para habilitar esta capacidad. Este ESB, al igual que JeeESB, no cumple con las capacidades necesarias para soportar una SOA-SIS como propone este trabajo.

Apache Synapse y WSO2 ESB ofrecen una solución ESB orientada a la mensajería más que una solución ESB integral; aunque soportan diversos protocolos de transporte de mensajes y de transformación de datos, no soportan estándares para definición ni orquestación de servicios, lo que dificulta el establecimiento de una SOA. Para el caso particular de este trabajo, estos no soportan el estándar de mensajería HL7, lo que implica, una dificultad en el establecimiento de interoperabilidad bajo este estándar de salud. Al igual que JeeESB y Celtix ESB, son soluciones que no responden a los requerimientos para soportar una SOA robusta y extensible.

MuleESB es una solución bastante robusta para el enrutamiento de mensajes; sin embargo, es deficiente para el establecimiento de una SOA debido a que no soporta estándares para la definición de servicios; de igual manera, carece de soporte para HL7 y de una consola de administración, lo que dificulta establecer reglas de seguridad y de infraestructura.

ServiceMix es un caso particular, a pesar de que soporta la mayoría de los requerimientos para el establecimiento de SOA, carece de estándares para la definición y orquestación de servicios, aspecto crítico en la asignación de operaciones específicas de cada uno de estos así como en la definición de los procesos de negocio que soportan el carácter empresarial de las organizaciones; a pesar de que soporta HL7 esta herramienta no cuenta con lo necesario para responder ágilmente a los cambios en los objetivos y procesos de negocio.

PEtALS ESB es una solución que soporta estándares para mensajería sincrónica y asincrónica, así como protocolos para transporte; sin embargo, esta herramienta no soporta directamente el estándar HL7, lo que implica la utilización de herramientas adicionales como el componente HL7



de ServiceMix para habilitar esta función; además de esto, no fue encontrada alguna implementación de este ESB para soportar HL7.

JBossESB y FuseESB son herramientas que soportan enrutamiento basado en contenido y sincronismo/asincronismo de mensajes; implementan estándares de mensajería y de transporte así como estándares para la transformación de datos. Cuentan con la capacidad para la definición y orquestación de servicios, y entornos de desarrollo que facilita implementaciones ágiles para responder a los cambios en los procesos de negocio. El inconveniente de estas herramientas, radica en la utilización de funcionalidades de otras herramientas como Apache Camel para soportar capacidades como el soporte para HL7. Además de esto, para el caso particular de JBossESB, la documentación es bastante limitada, no hay un soporte o documentación que resuelva todos los inconvenientes en el manejo de la herramienta.

En este punto, las herramientas más opcionadas para soportar la SOA-SIS propuesta en este trabajo son ChainBuilder ESB y OpenESB.

OpenESB es una herramienta bastante completa, es intuitiva y tiene un entorno de desarrollo integrado que permite administrar muchos de sus recursos de una forma sencilla. Soporta todos los estándares de mensajería y de transformación de datos que permiten el establecimiento de canales de comunicación adecuados entre SIS dentro de un entorno heterogéneo. Es de importancia resaltar el soporte por parte de la comunidad de software libre, además de material adicional existente en la web.

Por otro lado, ChainBuilder ESB es una herramienta que aunque muy similar a OpenESB, tiene el inconveniente de carecer de estándares para la definición de servicios, ya que, está orientado a la integración de recursos existentes expuestos por medio de servicios, lo que supone que estos ya estén creados y disponibles en un entorno de red. Era de gran importancia considerar la utilización de este ESB debido a que soporta las especificaciones y recomendaciones del proyecto HSSP; sin embargo, la utilización de este implicaría hacer uso de herramientas adicionales para la definición de servicios así como su posterior migración al servidor de aplicaciones de ChainBuilder ESB para su despliegue en la arquitectura.

En orden a estas ideas, la herramienta seleccionada para la implementación de la SOA-SIS propuesta en este trabajo es OpenESB, la cual se ajusta adecuadamente a los requerimientos y criterios establecidos para la elección.

### **3.5. OPCIONES TECNOLÓGICAS PARA HL7.**

Este apartado describe las posibilidades tecnológicas para habilitar HL7 en una arquitectura SOA.

#### **3.5.1. HL7 BUILDING COMPONENT.**

Habiendo seleccionado el OpenESB que permite la implementación de HL7, el Anexo G describe el proceso de instalación del codificador HL7 Building Component y la configuración de los plugins necesarios en el IDE para permitir la creación de proyectos.



En el anexo E está descrito el proceso de instalación, configuración, creación, validación de un codificador HL7 v2 y la implementación de un Building Componente para hacer la transformación de v2 a v3 empleando los plugins de mapeo de HL7 GForse.

### **3.5.2. MOTOR DE INTERFACES HL7 (MIRTH).**

Definido el ESB con capacidades para HL7 podría cerrarse el estudio de otras alternativas de habilitación de HL7 dentro de una arquitectura SOA para salud, sin embargo son discutidas las capacidades de hacer interactuar a la arquitectura con un motor de interfaces HL7 con una lista de motores abiertos de libre distribución.

- InterSystems Ensemble
- Mirth Connect
- Siemens OpenLink
- TC ConnectMate Software

Para efectos demostrativos de esta posibilidad tecnológica fue seleccionado Mirth Connect por la amplia documentación que redundaba en facilidad de uso.

Mirth es un motor con interfaz HL7 de plataformas cruzadas de código abierto que permite el envío bidireccional de mensajes HL7 entre sistemas y aplicaciones sobre múltiples capas de transporte. Utilizando un ESB (Mule ESB) y una arquitectura orientada a canales, Mirth permite el filtrado de mensajes, el transformado, y el enrutamiento de los mismos con base en unas reglas definidas por el usuario. Crear interfaces HL7 para los sistemas es fácil utilizando la interfaz web y el asistente para crear canales que asocian las aplicaciones con los componentes del motor Mirth.

#### **Características:**

- Amplia variedad de conectores. Mirth puede configurarse para escuchar y enviar mensajes HL7 y conectar una variedad de protocolos:
- TCP/MLLP
- Bases de datos (MYSQL, Postgres, Oracle, MS SQL, ODBC)
- JMS
- FTP/SFTP
- SOAP (sobre HTTP)
- Plataforma cruzada. Mirth soporta la mayoría de sistemas operativos (aquellos que soporten la máquina virtual de Java en su versión 1.6).
- Creación o utilización de filtros y perfiles de validación. El sistema de filtrado de Mirth permite elegir el tipo de mensajes que son aceptados y encaminados. Múltiples destinatarios pueden seleccionarse automáticamente especificando los filtros HL7.
- Creación o utilización de transformadores. Una interfaz de Mirth permite la creación de transformadores y mapeos de datos HL7. Simplemente seleccionando y arrastrando con el puntero del ratón fragmentos de mensajes HL7 creamos mapeos, o utilizar una variedad de funciones para hacer consultas en la bases de datos, enviar correos electrónicos. Las transformaciones disponibles son las siguientes:



- Transformador de mapeo: mapea los datos desde los mensajes entrada hasta las variables.
- Transformador de script: ejecuta scripts definidos en los mensajes (por ejemplo, JavaScript, Python, Tcl).
- Generador de mensajes HL7: construye mensajes HL7 a partir de una fuente de datos.
- Transformador XSLT: ejecuta transformaciones XLS sobre mensajes de entrada HL7 v3 o XML.
- Todos los mensajes y transacciones son registrados en una base de datos interna.

El Anexo H contiene una guía práctica para la creación de un canal que permite realizar una transformación mapeando de una base de datos PostgreSQL a un archivo HL7 información básica de una epicrisis, y posteriormente la creación de un canal SOAP (servicio web), como interfaz que instancia el canal con la transformación base datos HL7. De esta forma queda demostrando que es posible emplearlo como un proveedor de servicios al ESB.

### **3.5.3. CHAINBULDER.**

De la misma forma como es posible emplear un motor de interfaces para la transformación de mensajes y obtener HL7 puede emplearse las capacidades de ChainBulder en el caso de que sea estimado necesario descentralizar el “parseo” de los mensajes. Sin embargo es estimado que subutilizaría las otras capacidades de esta tecnología, en caso de requerir tal funcionalidad particular es aconsejable el empleo de un motor de Interfaces ya que concentra sus recursos en la prestación de los servicios que provee. El Anexo I es una guía de implementación creada por la empresa Bostech Corporation de transformación de mensajes sobre ChainBuilder de forma similar como expone el Anexo D sobre OpenESB y el motor de Interfaces Mirth en el Anexo H, con la diferencia que la transformación es XML a XML empleado el “parser” HL7 que lleva consigo, pero son suficientes para demostrar que es factible emplearlo como motor de interfaces HL7.

## **3.6. CONCLUSIONES**

Analizando las diferentes tecnologías y protocolos para el despliegue de una SOA y considerando diferentes niveles dentro del dominio de la solución como lo son:

- **Nivel de transporte:** permita la comunicación entre servicios.
- **Nivel de protocolos de red y comunicación:** estándares que soportan la trasmisión de los mensajes entre servicios.
- **Nivel de descripción de servicios:** permite la descripción de completa de servicios para ser consultados, identificados, y consumidos.
- **Nivel de servicios:** tipo de software que ejecutara los servicios.
- **Nivel de procesos de negocio:** permita definir y ejecutar procesos y flujos de negocio.
- **Nivel de registros de servicios:** mecanismos de publicación y descubrimiento de servicios.
- **Nivel de Políticas:** define las políticas funcionales y no funcionales de acuerdo a las capacidades de los servicios.



- **Nivel de Seguridad:** encargado de aspectos de seguridad como autenticación, autorización, privacidad, integridad, confidencialidad, y comunicación segura.

Considerando los aspectos importantes descritos en los diferentes niveles de las opciones tecnológicas son establecidos los criterios de selección del ESB, teniendo en cuenta los aspectos que permiten considerar la relevancia de cada aspecto a tener en cuenta en la evaluación comparativa de las opciones para el bus de servicios empresariales.

Finalmente fue definida la selección de una plataforma para la implementación del SIS ESB, siendo escogida OpenESB, la cual cumple en forma adecuada los requerimientos y criterios establecidos en el proceso de selección y describe opciones tecnológicas de interoperabilidad de información usando el estándar HL7



## 4. CAPITULO IV. EVALUACIÓN DE LA ARQUITECTURA SOLUCIÓN.

Este capítulo describe los dos procesos de negocio (servicios de negocio) construidos sobre la arquitectura luego de seguir el proceso metodológico propuesto para una arquitectura orientada a servicio descrito por Erl (Anexo A). Primero presenta la estructura de la cabecera del mensaje HL7 empleado en uno de los servicios de aplicación. Posteriormente describe la metodología de prueba para evaluar los requerimientos no funcionales de flexibilidad y escalabilidad en la interoperabilidad de los dos procesos de negocio del Hospital Susana López de Valencia. Esta evaluación incluye la descripción de los escenarios de evaluación, resultados obtenidos y su respectivo análisis.

### 4.1. DESCRIPCIÓN DE LOS PROCESOS DE NEGOCIO.

En el establecimiento de una SOA es fundamental incluir la totalidad de los procesos de negocio de la institución, teniendo en cuenta lo anterior el modelo TOGAF es aplicado para la descripción de arquitecturas empresariales, y para este caso de estudio, emplea la metodología de referencia SOA sobre dos de los procesos de negocio del HSLV, ellos son:

- **Auditoría:** permite hacer una verificación de los servicios prestados por el HSLV y sus valores facturados.
- **Tele-consulta:** permite al personal médico del HSLV contratar servicios de especialistas de otras entidades conectadas a través de la Red Nacional de Tecnología Avanzada (RENATA), los cuales tienen el acceso a la información clínica de los pacientes del HSLV para realizar sus consultas de manera remota.

Estos procesos fueron seleccionados previa concertación con los ingenieros del departamento de Sistemas del HSLV, dado que el servicio de Tele-consulta no existía, y el proceso de auditoría estaba dividido en dos procesos: uno para consultar información clínica y otro para consultar facturación de servicios.

Luego de la aplicación de la metodología fueron obtenidos, para los servicios de negocio Auditoría y Tele-consulta, los servicios candidatos descritos en la tabla 5.

**Tabla 5 Servicios**

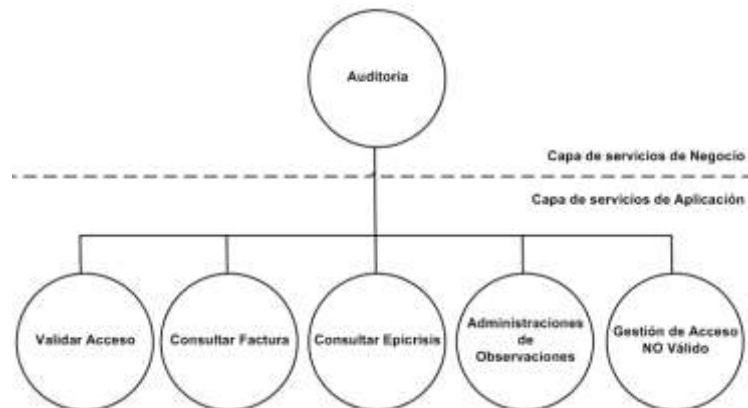
<b>Servicios</b>	<b>Lógica de Aplicación</b>
<b>Validar Acceso.</b>	No existe servicio legado, pero existen los datos de los repositorios.
	Identifica si hay acceso o no al proceso requerido.
	Retorna un mensaje de error y dispara el registro de errores en caso de que el acceso no sea válido.
	En caso de que el acceso sea válido, habilita el flujo normal del proceso.
<b>Administración de observaciones.</b>	Recibe un mensaje que contiene el tipo de operación a ejecutar: eliminar, crear, consultar y editar.
	Como resultado brinda el estado actual de las observaciones registradas.
<b>Consultar Historia Clínica.</b>	Recibe una solicitud que contiene el identificador de la consulta clínica requerida.
	Accede a la información de la consulta clínica.
	Retorno de la información en el formato adecuado.



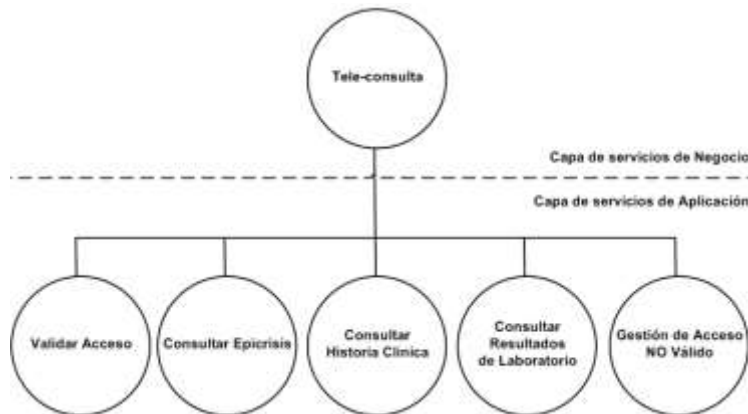


<b>Consultar Factura.</b>	Recibe una solicitud del consumidor del servicio con el identificador de la factura deseada.
	Accede a la información de la factura
	Retorna la información de la factura en formato adecuado.
<b>Gestión de acceso no valido.</b>	Recibe la información del servicio solicitado y el consumidor que lo requiere.
	Almacena la información de negación del acceso.
<b>Consultar Epicrisis.</b>	Recibe una solicitud que contiene el identificador de la epicrisis requerida.
	Accede a la información de la consulta de epicrisis.
	Retorno de la información en el formato adecuado.
<b>Consultar resultados de Laboratorio.</b>	Recibe la información que identifica los resultados solicitados.
	Accede a los resultados solicitados.
	Retorna los resultados en formato requerido.
<b>Consultar imágenes diagnósticas.</b>	Recibe la información que identifica las imágenes solicitadas.
	Accede a los recursos solicitados.
	Retorna los resultados en formato requerido.

A continuación las figuras muestran el modelo de la composición de los del proceso auditoria y tele consulta.



**Figura 17. Representación del la composicion del Proceso de Auditoria.**



**Figura 18. Representación del a composicion del Proceso Tele-consulta.**

Las figuras 17 y 18 representan una vista de los servicios de negocio definidos para llevar a cabo en los procesos de negocio y los servicios de aplicación requeridos para el proceso, el modelo de



orquestación y orientado por el proceso de negocio obtenido en la aplicación de la metodología de referencia SOA, detallado en el Anexo A.

## 4.2. ANALISIS DE SISTEMAS LEGADOS.

Uno de los pasos fundamentales en el establecimiento de una SOA es el establecimiento de la interoperabilidad con los sistemas legados. Para ello es expuesta una breve descripción de las herramientas con las que cuenta el HSLV y el nivel de interoperabilidad es descrito por medio de los protocolos definidos para la interoperabilidad (Anexo A).

**Tabla 6. Tabla analisis de sistemas legados**

<b>Aplicación</b>	<b>Descripción</b>	<b>Interoperabilidad</b>
<b>Dinámica Gerencia Versión Fox.</b>	Encargado de los procesos Operativo Administrativos y financieros de la institución. Base de datos en SQL Server. La arquitectura de la aplicación es Cliente servidor de nivel 2, es decir los clientes de la base de datos son aplicaciones de escritorio.	No soporta interoperabilidad a nivel de procesos con otros sistemas. Existe interoperabilidad con otros sistemas a nivel de SQL soportados en SQL Server.
<b>Dinámica gerencial version.net</b>	Actualmente este sistema está encargado de los procesos asistenciales de la institución (Historias clínicas y citas médicas). Al estar realizado en .NET soporta servicios web. Motor de bases de datos es SQL Server manejar estándares tales como XML, HL7 y DICOM.	Maneja interoperabilidad de algunos procesos internos, evidencia esto las conexiones a la base de datos cuando están centralizados algunos Querys comunes de los módulos que componen la aplicación, sin embargo las rutas de las WSL así como los archivos de configuración de todo el aplicativo (rutas de recursos, imágenes, etc.) están sofocados. De esta forma puede concluirse que el proveedor restringió o no consideró el acceso público a los servicios web.
<b>Validador de RIPS SAGITA</b>	Es una aplicación independiente a los demás sistemas de la institución pero que depende de la información suministrada desde el sistema Dinámica Gerencial a través de archivos planos. La información que retorna corresponde a los archivos RIPS (Registro Individual de Prestación de Servicios de Salud). Realizada en Delphi y base de datos en Firebird.	No existe ningún tipo de interoperabilidad técnica de procesos. Solo existe la interoperabilidad semántica necesaria para procesar la información suministrada.
<b>DataLab</b>	Sistema de laboratorio el cual administra la información generada desde los equipos de laboratorio, la comunicación entre los equipos y el sistema es establecido utilizando protocolos e interfaces propias. Actualmente están trabajando en una interfaz entre el sistema de laboratorio y la historia clínica cuyo funcionamiento basada en archivos script SQL.	Interfaz con otros sistemas a través de disparadores y script SQL.



<p><b>Carestream</b></p>	<p>Sistema encargado de la captura y administración de imágenes diagnósticas utilizando el estándar DICOM, específicamente esta institución maneja las imágenes de rayos X. Es un sistema modular con mucho potencial que es implantado de acuerdo a las necesidades del cliente. El hospital cuenta con un sistema que permite la captura, almacenamiento y visualización de las imágenes tomadas, pero para la interfaz con la historia clínica hay desarrollada una aplicación propietaria que realiza este proceso.</p>	<p>La forma en que son compartidos los resultados de este sistema es por medio del establecimiento de un directorio compartido para todos los equipos de la red.</p>
<p><b>Aplicaciones desarrolladas</b></p>	<p>Son aplicaciones desarrolladas con el fin mejorar los procesos institucionales. Normalmente estas aplicaciones son creadas basándose en la información contenida en las bases de datos alimentadas desde el sistema Dinámica Gerencial (las dos versiones). Actualmente trabajan en la construcción de un sistema unificado de aplicaciones llamado HSLV.NET cuyo objetivo es integrar las distintas aplicaciones desarrolladas en el hospital durante los últimos años.</p>	<p>Operación conjunta con los aplicativos a nivel de base de datos, es decir que la información que administra son las de la base de datos de los demás aplicativos.</p>
<p><b>Equipos de apoyo diagnóstico y cuidado del paciente</b></p>	<p>Actualmente el hospital cuenta con equipos médicos para el apoyo diagnóstico (endoscopia, colonoscopia, colposcopia, ecografía de 2do y tercer nivel) y equipos que están para el cuidado del paciente (respiradores, monitores, electrocardiógrafos). Todos estos equipos brindan información vital para el diagnóstico adecuado de un paciente</p>	<p>Debido a su uso básico, no existe interoperabilidad técnica con los demás sistemas de información con los que actualmente cuenta el hospital. Todos estos equipos operan de manera independiente y la administración de la información que procesan es integrada a los demás sistemas de información de manera periódica a través de procesos manuales.</p>

Como lo muestra la Tabla 6, el ambiente de interoperabilidad es hostil al no encontrar los protocolos necesarios y disponibles para habilitar la interoperabilidad de procesos basados en servicios web. Por tal motivo la única interoperabilidad posible es a nivel de datos y no de procesamiento. Para lograr interoperabilidad a nivel de datos puede emplearse el API de persistencia JAVA (JPA), facilita la construcción de objetos que controlan el acceso a datos y eliminar el alto consumo de las sesiones de conexión a los motores de almacenamiento al establecer y administrar pools de conexiones como describe el Anexo C (Sección C.4), que serán empleadas en todos aplicativos soportados por la arquitectura.

### **4.3. APLICACIÓN DE HL7 EN LA ARQUITECTURA SOLUCIÓN.**

Para efectos de validación del uso del estándar de interoperabilidad semántica (HL7) en la arquitectura de implementación es agregado en el servicio de aplicación consulta de epicrisis un campo de respuesta tipo cadena con el mensaje y la información con el estándar HL7. Para lograr



esto fue habilitado el Bussines Component de HL7 versión 3 como describe el anexo D (sección D.2, Instalación de Building Component).

Para definir la estructura del mensaje fue empleada la guía de implementación para resultados de laboratorio basada en el estándar HL7 para Arquitectura del Documento Clínico (Clinical Document Architecture, CDA) versión v21, propuesta por HL7 Colombia. Basados en esta especificación fue construido el mensaje tal como lo detalla el anexo D. A continuación, el encabezado del CDA es mostrado.

```
<?xml version="1.0"?>
<ClinicalDocument xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" moodCode="EVN"
xmlns="urn:hl7-org:v3 CDA.xsd">
  <typeId root="2.16.840.1.113883.1.3" extension="POCD_HD000040" />
  <id root="2.16.840.1.113883.19.4.x" extension="c266" />
  <code code="26436-6" codeSystem="2.16.840.1.113883.6.1"
codeSystemName="LOINC" displayName="Epicrisis" />
  <title>Epicrisis</title>
  <effectiveTime value="20081209001500" />
  <confidentialityCode code="R" codeSystem="2.16.840.1.113883.5.25" />
  <languageCode codeSystem="2.16.840.1.113883.6.121" />
  <recordTarget typeCode="RCT" contextControlCode="OP">
    <patientRole classCode="PAT">
      <id root="2.16.840.1.113883.19.5" extension="55140250" />
      <addr use="HP">
        <streetAddressLine>Carrera 29A1 No. 12B-105</streetAddressLine>
      </addr>
      <patient classCode="PSN" determinerCode="INSTANCE">
        <id root="2.16.840.1.113883.19.5" extension="55140250" />
        <name>
          <given>JUAN</given>
          <given>DANIEL</given>
          <family>MORALES</family>
          <family>GUTIERREZ</family>
        </name>
        <administrativeGenderCode code="M"
codeSystem="2.16.840.1.113883.5.1" />
        <birthTime value="20090203" />
      </patient>
      <providerOrganization classCode="ORG" determinerCode="INSTANCE">
        <id root="2.16.840.1.113883.19.5" extension="890.307.200-5" />
        <name>
          <suffix qualifier="LS">S.A.</suffix>Hospital Susana Lopez de Valencia
        </name>
        <telecom value="fax: 57 8381151" />
        <telecom value="tel: 57 8211721" />
        <telecom value="tel: 57 8382411" />
        <telecom value="mailto: informacion@hosusana.gov.co" />
        <telecom value="http: http://www.hosusana.gov.co " />
        <addr>
          <streetAddressLine>Calle 15 N° 17A - 196 La Ladera</streetAddressLine>
          <city>Popayán</city>
          <state>Cauca</state>
          <country>Colombia</country>
        </addr>
      </providerOrganization>
    </patientRole>
  </recordTarget>
  <author typeCode="AUT" contextControlCode="OP">
    <time value="20090203001746" />
    <assignedAuthor classCode="ASSIGNED">
      <id root="2.16.840.1.113883.19.5" extension="KP00017" />
      <assignedAuthoringDevice determinerCode="INSTANCE">
        <code code="dev1" />
        <manufacturerModelName>Peter Pinchao</manufacturerModelName>
        <softwareName>SOA SIS</softwareName>
      </assignedAuthoringDevice>
    </assignedAuthor>
  </author>
</ClinicalDocument>
```



```

</assignedAuthoringDevice>
<representedOrganization classCode="ORG" determinerCode="INSTANCE">
  <id root="2.16.840.1.113883.19.5" extension="890.307.200-5" />
</representedOrganization>
</assignedAuthor>
</author>

<componen>

</componen>
</ClinicalDocument>
    
```

A continuación es presentada la descripción de los elementos de conforman un CDA (solo fue considerado lo referente a la cabecera.)

**CONVENCIONES UTILIZADAS EN LA TABLA DE ELEMENTOS DEL CDA**

ELEMENTO	SIGNIFICADO
H	Indica que el elemento es obligatorio para HL7 (y por lo tanto para la guía)
X	Indica que elemento es obligatorio (según el contexto)
/	Indica que el elemento es opcional
+n	Indica que se debe escoger al menos uno de los elementos con el número n

**TABLA DE ELEMENTOS DEL CDA**

CAMPO	CONVENCIÓN	CARDINALIDAD	OBSERVACIONES	TIPO DE DATO
ClinicalDocument	H	0..1	Elemento raíz del CDA	ClinicalDocument
typeId	H	1..1	Default: @root="2.16.840.1.113883.1.3"; @extension="POCD_HD000040"	II
classCode	H	1..1	Indica la clase de documento Default: DDCCLIN	CS
moodCode	H	1..1	Indica la intención del documento Default: EVN	CS
id	H	1..1	Identificador único del CDA	II
Code	H	1..1	Tipo de doc Ej: "Resultado de Laboratorio", según el vocabulario LOINC.	CE
Title	H	1..1	Título del CDA	ST
effectiveTime	H	1..1	fecha de generación del documento original del acto clínico al que se asocia el CDA (yyyyMMddHHmmss)	TS
confidentialityCode	H	1..1	Indica el tipo de confidencialidad Normal, Alta, Muy Alta (CWE: Coded With Extensions)	CE
Code	H		[N, R, V]	
codeSystem	H		Sistema de codificación: (codeSystem 2.16.840.1.113883.5.25)	
languageCode	/	0..1	Indica el lenguaje del documento Default: @codeSystem="2.16.840.1.113883.6.121"	CS
recordTarget	H	1..1	Agrupación de información del paciente	RecordTarget
typeCode	H	1..1	Default: RCT	CS
contextControlCode	H	1..1	Default: OP	CS
patientRole	H	1..1	Rol del paciente	PatientRole
classCode	H	1..1	Default: PAT	CS
id	H	1..1	Identificación del paciente en el sistema Ej: número de historia clínica	II
Addr	X	0..1	Dirección del paciente	AD
Use (atributo)	X	0..*	HP (home), WP (work). Se pone solo USE="HP" y luego toda la dirección.	SET<CS>



Patient	X	1..1	Agrupación de información del paciente: PSN	Patient
classCode	H	1..1	Default: PSN	CS
determinerCode	H	1..1	Default: INSTANCE	CS
id (DEPRECATED)	H	1..1	Identificación del paciente EJ: Cédula	II
Name	X	1..1	Nombre del paciente	PN
Given	X	1..*	Los nombres del paciente (cada uno en un 'given')	ST
Family	X	1..*	Los apellidos del paciente (cada uno en un 'family')	ST
administrativeGenderCode	X	1..1	Genero (2.16.840.1.113883.5.1) (CWE)	CE
birthTime	/	0..1	Fecha de nacimiento (YYYYMMDD)	TS
providerOrganization	X	1..1	Agrupación de información del laboratorio	Organization
classCode	H	1..1	Default: ORG	CS
determinerCode	H	1..1	Default: INSTANCE	CS
id (root, extensión)	X	1..1	Identificación de la entidad (OID)	II
Name	X	1..1	Nombre (el status legal de la empresa SA, LTDA, etc. Se coloca como suffix con qualifier="LS")	ON

Telecom	X	1..1	Teléfono	TEL
Addr	X	1..1	Dirección	AD
Author	H	1..1	Agrupación de la información del autor del CDA (Persona o Dispositivo).	Author
TypeCode	H	1..1	Default: AUT	CS
contextControlCode	H	1..1	Default: OP	CS
Time	H	1..1	Hora a la que se produjo el resultado (por parte de la máquina de análisis o del bacteriólogo: según el AUTOR). (yyyyMMddHHmmss)	TS
assignedAuthor	H	1..1	Información del autor	AssignedAuthor
classCode	H	1..1	Default: ASSIGNED	CS
id (root, extensión)	H	1..1	Identificador del autor. El laboratorio debe tener su catálogo de máquinas de análisis con OID.	II
assignedAuthorChoice	X	1..1	Elige una (persona o maquina)	Person   AuthoringDevice
assignedPerson	+2H	1..1	Si es persona	Person
classCode	H	1..1	Default: PSN	CS
determinerCode	H	1..1	Default: INSTANCE	CS
Name	X	1..1	Nombre	PN
given	X	1..2	Primer y segundo nombre	
family	X	1..2	Primer y segundo apellido	
assignedAuthoringDevice	+2H	1..1	Si es maquina	AuthoringDevice
classCode	H	1..1	Default: DEV	CS

#### **4.4. IMPLEMENTACIÓN EMPLEANDO HL7 BUILDING COMPONENT.**

La implementación de esta cabecera es requerida para probar el componente que representa el Parser HL7 de la arquitectura definida, es decir el HL7 Building Component. Este componente está en la capacidad de codificar desde un medio de entrada que puede ser un archivo XML, base de datos, archivos planos, siempre que cuente con el archivo que describe el formato de



documento de salida definido por un descriptor (XML Schema Definición, XSD<sup>19</sup>), y los atributos (datos) de la fuente, para realizar el correspondiente mapeo.

El anexo D detalla el proceso de implementación, consta de los siguientes pasos:

- **Instalación de HL7 Encoder:** instala el codificador como una librería de componente común en el servidor de aplicaciones (es decir, disponible para todos los objetos del JNDI).
- **Instalación de los Plugins de NetBeans IDE:** instala los complementos y plugins que requiere Netbeans para la construcción de servicios y componentes con la funcionalidad proveída por el HL7 Encoder, para la creación de componentes de transformación de formatos y componentes para la transmisión de los mismos.
- **Instalación del HL7 Building Component:** detalla el despliegue en OpenESB de Building Component HL7 que al ser un componente JBI puede recibir y enviar mensajes a través de diversos niveles de transporte y protocolos particulares.
- **Prueba de WSD:** creando un mensaje codificando a un archivo describe los pasos para la reacción de un servicio de envío de mensaje.
- **Prueba del servicio:** desde una aplicación de composición (Compsite Application).
- **Creación de un componente para mapear el archivo de HL7v2 a HL7 v3:** empleando el plugin de HL7 Gforse (HL7 Mapping Tools) en Eclipse, describe la configuración del IDE para habilitar herramientas que permitan la creación de un componente JBI para la transformación del mensaje de versión 2 a versión 3 e integración en el JBI de OpenESB.
- **Construcción de un EJB para la publicación del mensaje HL7v3.** este aparte detalla los pasos con los que es consumido el componente y siendo el EJB encargado de publicar como servicio web (como es explicado en el Anexo F), queda habilitado para ser representado y empleado como describe el modelo de composición de la figura 17.

Como es evidente para el empleo del estándar en la versión 3 es necesario hacer una labor adicional de transformación de versiones, esto puede afectar el rendimiento de la aplicación. ChaingBuilder también soporta el mismo HL7 encoder por lo que requiere de hacer una transformación a formatos para la versión tres del estándar, es decir, necesita repetir los últimos dos ítems antes descritos. El anexo I describe un ejercicio similar a las secciones del Anexo D, D.3, D.4 y D.5, pero implementadas en ChaingBuilder. Por lo tanto, es aconsejable emplear un transformador que integre estas operaciones agilizando la labor de implementación y centraría la recepción y envío de mensajes HL7 en la versión 3 en el motor de interfaces como fue descrito en el Anexo H empleando Mirth.

#### **4.5. ESENAIO DE INTEROPERABILIDAD DE SERVICIOS.**

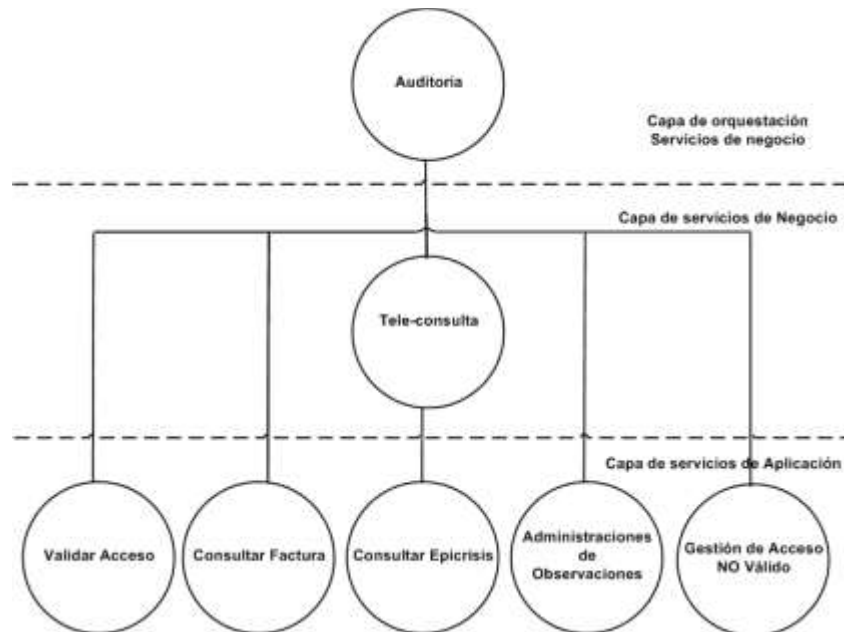
Es deseable poder habilitar la reutilización de los procesos de negocio existentes en las soluciones informáticas de la institución, sin embargo como muestra la Tabla 6 no es posible habilitar la interoperabilidad con estos sistemas a nivel de procesos.

---

<sup>19</sup> Creado por la W3C para la definición de documentos basado en XML.



El proceso Auditoria es un proceso de negocio que luego de la aplicación de la metodología de establecimiento de una SOA da origen a un servicio de negocio que a su vez está compuesto de un servicio de otro servicio de negocio como o es el de la Tele-consulta. Esta interacción determina la necesidad de orquestación de los servicios web.



**Figura 19. Jerarquía de composición del servicio de negocio de Auditoria.**

Como indica la figura 19, el servicio de auditoria es consumido del proceso Tele-consulta de esta forma a través de él, es posible hacer la consulta de la epicrisis, también es posible si son extendidas las capacidades del servicio solicitar información más específica como información de los resultados de los exámenes. Sin embargo, para este escenario de prueba será restringida a la consulta de la epicrisis a través el servicio de Tele-consulta, de esta forma queda especificada la interoperabilidad entre los procesos de negocio.

#### **4.6. CRITERIOS DE EVALUACIÓN (EVALUACIÓN DE LA ARQUITECTURA).**

El establecimiento de la vista de la ingeniería centra esfuerzos en proponer los componentes arquitectónicos que ofrezcan satisfacer los requerimientos no funcionales de los sistemas informáticos y de comunicación soportados sobre la arquitectura. En esta sección están descritos los criterios que pondrán a prueba los componentes seleccionados de la arquitectura.

##### **4.6.1. OBJETIVOS DE EVALUACIÓN.**

Los principales objetivos en la evaluación de la arquitectura orientada a servicios son los requerimientos no funcionales de flexibilidad y escalabilidad en la interoperabilidad de los dos procesos de negocio del Hospital Susana López de Valencia. Una definición precisa para cada uno de ellos es:





- **Flexibilidad:** la facilidad con que un sistema o componentes pueden ser modificados para su uso en aplicaciones o entornos que no sean aquellos para los que fue diseñado específicamente (IEEE, 2000).
- **Escalabilidad:** refiere a la capacidad de una aplicación para manejar carga de trabajo adicional, sin afectar negativamente el rendimiento, añadiendo recursos tales como el procesador, la memoria y la capacidad de almacenamiento (Meier, y otros, 2007).

#### **4.6.1.1. Evaluación Comparativa.**

La evaluación comparativa (benchmarking) es el proceso de comparar el rendimiento de un sistema en contra de una línea de base creada internamente o en contra de un estándar de la industria avalada por alguna otra organización (Meier, y otros, 2007).

Es posible emplear dos formas de establecer el benchmarking, la primera es definir las metas que deben cumplir los sistemas, descritas en los Acuerdos de Nivel de Servicios (Service Level Agreements, SLA), por ejemplo: el tiempo de respuesta de una consulta cuando el sistema este mas ocupado. Esto da como resultado indicadores de rendimiento claves (Key Performance Indicators, KPI) (Oracle, 2010).

La otra forma de establecer una evaluación comparativa es tomar dos o más sistemas soportados por arquitecturas comparables y realizar las pruebas de escalabilidad con el fin de establecer las diferencias.

#### **4.6.2. EVALUACIÓN DE ESCALABILIDAD.**

Para probar la escalabilidad son consideradas las métricas más importantes como el tiempo de respuesta, latencia, rendimiento, utilización de recursos, de manera que permita estimar la capacidad de la arquitectura de hacerse más grande sin perder la calidad de la prestación de los servicios. Este trabajo centra su estudio principalmente en métricas de latencia y rendimiento.

##### **4.6.2.1. Métricas de evaluación**

Las métricas de evaluación de escalabilidad seleccionadas son:

- **Tiempo de respuesta:** el tiempo transcurrido entre el inicio de una solicitud o comando a un sistema informático y el principio de la respuesta del sistema (IEEE, 2000).
- **Latencia:** el tiempo entre la emisión de una solicitud y el comienzo de obras reales de esa petición (transferencia de datos es comenzada).
- **Throughout:** número de unidades de trabajo que puede manejar por unidad de tiempo (rendimiento), por ejemplo, las solicitudes por segundo, las solicitudes por día, visitas por segundo informe, por año, etc. (Meier, y otros, 2007).
- **La utilización de recursos:** métrica de un consumo del recurso tecnológico (Los recursos principales son el procesador, memoria, disco E/S, y la red I/O).

Las anteriores métricas son básicas y fundamentales para la estimación de la escalabilidad de una arquitectura, ahora resta definir los tipos de pruebas a establecer, según las recomendaciones



Microsoft (Meier, y otros, 2007) y Oracle pero omitiendo que las pruebas de funcionalidad dado el especial interés del modelado y diseño a procesos de negocio y la simplicidad de los procesos diseñados.

**Pruebas de carga:** centradas en determinar o validar las características de rendimiento del sistema o la aplicación sometida a prueba cuando es enfrentada a diferentes volúmenes de cargas de trabajo prevista durante las operaciones de producción (Meier, y otros, 2007).

**Pruebas de stress:** las pruebas realizadas para evaluar un sistema o componente en y más allá de los límites de su requisitos especificados (IEEE, 2000).

Las pruebas de estrés por lo general miden los límites de un sistema en, máximo número de usuarios simultáneos, máximo volumen de datos o script procesados.

De esta forma quedan determinadas las métricas y pruebas de evaluación de escalabilidad en la arquitectura solución.

#### **4.6.2.2. Metodología Evaluación Escalabilidad.**

La metodología descrita por Oracle (Oracle, 2010) y Microsoft (Meier, y otros, 2007), son muy similares en sus procedimientos y en su descripción. Sin embargo por limitantes de recursos tecnológicos y tiempo la metodología a emplear no hace una comparación entre cada una de las iteraciones de las pruebas (como proponen Oracle y Microsoft), dejando el análisis para el conjunto completo de las pruebas. Las fases que componen la metodología empleada son:

1. **Definir la carga de trabajo esperada:** consiste en estimar el nivel de carga a aplicar para tomarla como base de referencia (números de conexiones, solicitudes etc).
2. **Definir los objetivos de rendimiento:** como son percibidos desde el punto de vista del usuario, puede especificarse en términos de acuerdos de nivel de servicio.
3. **Seleccionar los subsistemas para estudiar:** los procesos de negocio pueden tomar varios caminos en la ejecución es por esto que ha de estimar el flujo de los procesos de negocio a emplear durante la evaluación.
4. **Realizar una prueba para crear un punto de referencia inicial (Benchmark).** Tomado de la evaluación de la carga esperada en el sistema, si es comparada frente a otro sistema deben tomarse los valores de la evaluación de este último bajo las mismas características de evaluación que serán aplicadas al sistema a evaluar.
5. **Seleccione los parámetros para modificar:** concurrencia, y recurrencia de la carga.
6. **Realizar la prueba.**
7. **Repetir los pasos 5-6.**
8. **Analizar los resultados.**

#### **4.6.3. PRUEBAS DE FLEXIBILIDAD.**

El establecimiento de métricas de flexibilidad en software está estrechamente relacionado con la cantidad de cambios que han de hacerse en los códigos fuente y la complejidad en los mismos para poder lograr los nuevos objetivos requeridos (Eden, y otros, 2006). De esta forma puede



decirse que el sistema de A es más flexible que B si el número de cambios requeridos para cumplir con los nuevos requerimientos es menor en A que en B y la complejidad de estos cambios también es menor.

Para lograr una medida en el estudio de Eden el concepto Paso a la Evolución es definido como la cantidad de trabajo empleado en hacer los cambios y la complejidad de los mismos. El paso de Evolución está relacionado con el costo del proceso de evolución (proceso de hacer cambios a un sistema o ajustes). Para poder medir el costo de la evolución son empleadas las métricas de costo de evolución (Eden, y otros, 2006), Estas métricas son descritas en detalle a continuación.

#### **4.6.3.1. Paso a la evolución.**

Sea  $P$  un conjunto de problemas que se han resuelto con un conjunto  $I$  de implementaciones sobre el sistema de información. Para distinguir entre los cambios en el problema y cambios en la aplicación se ha de referir a los primeros como  $p_{modificada}$  y los ajustes sobre los códigos fuente como  $i_{modificada}$ .

La relación de la evolución se expresa como:

$$\varepsilon: P \times P \times I \rightarrow I$$

De esta forma la función de evolución será:

$$\varepsilon(p_{anterior}, p_{modificada}, i_{anterior}) = i_{modificada} \quad \text{Ecuación (1)}$$

La anterior función aplica solo cuando en la implementación anterior al cambio ( $i_{anterior}$ ) se abordaron los problemas iniciales ( $p_{anterior}$ ) y en la Implementación modificada ( $i_{modificada}$ ) se resolvieron los nuevos problemas ( $p_{modificada}$ ).

El paso a la evolución es entonces la pareja:

$$\varepsilon = \langle (p_{anterior}, p_{modificado}, i_{anterior}), (p_{anterior}, p_{modificado}, i_{anterior}) \rangle \quad \text{Ecuación (2)}$$

Lo anterior hace referencia al hecho de cumplir los nuevos requerimientos en la evolución del sistema. Sin embargo la complejidad de los cambios también es un aspecto a considerar en la medida de la flexibilidad por lo que es requerida una métrica que permita la medida de la complejidad de estas implementaciones. De esa forma las métricas están relacionadas con la cantidad de trabajo empleado en hacer los cambios y la complejidad de los mismos.

#### **4.6.3.2. Métricas de costo de evolución del software.**

Las métricas de evolución del software han sido aplicadas comúnmente para establecer diferencias entre distintos paradigmas desarrollo (programación estructurada y orientada a objetos principalmente). Son empleadas las métricas que aplican comúnmente a la programación orientada a objeto según el estudio de Eden (Eden, y otros, 2006).

- **Métrica costo de evolución**



Mide el costo de la ejecución de paso de la evolución en términos del número de clases afectadas por el ajuste.

$$C(\varepsilon) = |\Delta Clases(i_{anterior}, i_{modificada})| \quad \text{Ecuación (3)}$$

Donde  $\Delta Clases(i_{anterior}, i_{modificada})$  designa la diferencia simétrica de conjuntos del conjunto de clases en cada programa expresada por:

$$(Clases(i_{anterior})/Clases(i_{modificada}))U(Clases(i_{modificada})/Clases(i_{anterior})) \quad \text{Ecuación (4)}$$

De esta forma se tiene una métrica que permite establecer un criterio de comparación relacionado con la implementación de los cambios.

- **Métricas complejidad de evolución.**

Esta métrica provee un mecanismo para determinar la complejidad de la implementación de nuevos requerimientos funcionales sobre una aplicación.

Esta métrica centra sus valores en el número de operaciones (computaciones) necesarias para realizar los cambios en la implementación modificada ( $i_{modificada}$ ). Ello conlleva una relación lineal entre el tiempo computacional como evidencia el trabajo de Urquhart (Urquhart, 2004). De esta forma es estimada la complejidad como el número de operaciones  $OP$  para cumplir las funciones de los requerimientos modificados en las clases modificadas. Esto determinan el costo de operación en el paso a evolución de software ( $\varepsilon_{OP}$ ).

**Métrica costo de complejidad de evolución.**

Considerando los estudios de tiempos computacionales centrada en algoritmos sobre estructuras de datos (que generalmente son matrices, Data Structure - DS), Urquhart (Urquhart, 2004) establece una relación entre la modificación estas estructuras y el costo de operación. Sin embargo dado el grado de abstracción que ofrece la programación orientada a objetos y los recursos computacionales presentes en las bases de datos es considerada para efectos comparativos los cambios en el número de operaciones que implementan en las clases que son modificadas en la adición de una nueva operación funcional en una de las clases y su implementación sobre otras y mantiene la notación (DS) haciendo esta salvedad (Eden, y otros, 2006).

$$O(Clases(\varepsilon_{DS})) = \text{Numero de Clases Afectadas} \quad \text{Ecuación (5)}$$

**Métrica de costo de Operación**

Una vez los algoritmos están en funcionamiento debe considerarse el costo de operación ( $\varepsilon_{OP}$ ) de las estructuras modificadas que depende del número de datos que están alojados en estas estructuras.

$$O(Clases(\varepsilon_{OP})) = \text{Numero de operaciones agregadas} \quad \text{Ecuación (5)}$$



Las ecuaciones 5 y 6 aplican para un solo problema modificado. El costo de operación es determinado por las iteraciones presentes en los algoritmos (provenientes de los ciclos como for, while etc), esta medida está estrechamente relacionada con los datos provenientes de la fuentes de datos (Bases de Datos), por lo tanto es considerada en la evaluación y es definida de la siguiente forma cuando existe un conjunto de problemas:

$$O(\text{Clases}(\varepsilon_{OP})) = O(|DS|)$$

De esta forma podemos establecer la complejidad de operación sobre los elementos el sistema solución, simplificando el trabajo de considerar el costo de operación, sobre todo en los servicios que componen los servicios de negocio ya que esta construidos a partir de la implementación de un EJBs de sesión para publicarlos como servicios web.

#### **4.7. ESCENARIOS DE EVALUACIÓN.**

En esta sección son descritos los escenarios donde fueron aplicadas las evaluaciones de flexibilidad y escalabilidad a la arquitectura. Describe las pruebas realizadas sobre el servicio de negocio Auditoria médica, ya que este extiende del servicio de negocio tele-consulta en el momento en que es requerido la consulta de una epicrisis, de esta forma es establecida la evaluación de flexibilidad y escalabilidad en la interoperación de dos procesos de negocio del HSLV.

##### **4.7.1. ESCENARIOS DE EVALUACIÓN DE LA ESCALABILIDAD.**

Siendo la evaluación comparativa la esencia de una evaluación de escalabilidad como la comparación de dos sistemas informáticos. El caso más ideal sería comparar los servicios de negocio de la SOA solución con una aplicación existente en el HSLV bajo JEE que realice los mismos procesos de negocio. Desafortunadamente en el HSLV no existen tales sistemas que permitan hacer esta comparación, de esta forma será empleada una aplicación de la empresa ASMET Salud EPS de auditorías médicas, llamada AMedicas (Auditorias Médicas) con la salvedad que la consulta de epicrisis será obtenida de las bases de datos de Dinámica Gerencial (el mismo del HSLV) para que logre el objetivo de evaluar el servicio en un entorno similar al del HSLV. La consulta a bases de datos de Dinámica representa un cambio en las fuentes de AMedicas (permitiendo estos cambios establecer una base comparativa de la flexibilidad como esta expuesto más adelante). De esta forma es posible establecer una comparación de un ambiente SOA y otro No SOA.

Adicional a esta comparación son establecidos indicadores de rendimiento claves para establecer el “benchmarking”, como punto de referencia para hacer una evaluación comparativa y de esta forma tener dos comparaciones una entre ambientes SOA, no SOA y establecer acuerdos de nivel esperados.

Para la evaluación están disponibles de los recursos que nos permiten la implementación de cluster del ESB. La evaluación de la arquitectura fue desarrollada con los recursos tecnológicos y



logísticos de ASMET Salud EPS-S. El detalle de los escenarios de evaluación son descritos en el Anexo G.

#### **4.7.1.1. Pruebas de carga.**

La primera evaluación sobre los servicios será efectuada con un host donde los servicios de ESB son desplegados y un elemento de carga (otros hosts) con los parámetros de carga para tal fin. Antes de presentar los valores de carga son descritos los parámetros de configuración de los algoritmos de carga.

- **Hilo:** cada hilo configurado es un proceso de ejecución del servicio de negocio.
- **Periodo ente Hilos:** define cada cuantos segundos será disparado cada grupo de Hilos (conurrencia).
- **Número de veces de ejecución:** cuantas veces serán ejecutados determinado número de grupo de Hilos (recurrencia).

Fueron realizadas las pruebas de carga con el fin de estimar los niveles de servicio, ver la tabla 7.

**Tabla 7. Pruebas de carga para establecer el Benchmarking.**

<b>ID Prueba</b>	<b>Numero de Hilos por grupo</b>	<b>periodo entre grupos de Hilos (seg)</b>	<b>Número de veces ejecución</b>
P.1.1	60	30	3
P.1.2	90	20	6
P.1.3	120	10	9

Como puede observarse la relación de la configuración de cada parámetro es lineal entre cada iteración y en cada prueba ofrece un nivel de exigencia mayor que la prueba anterior, y con estos resultados el punto de referencia inicial (Benchmark) para la evaluación comparativa ha sido obtenido.

Posteriormente será incrementada la exigencia (la carga al sistema), para ello un cluster de ESB es utilizado como respuesta, ver la figura 20.

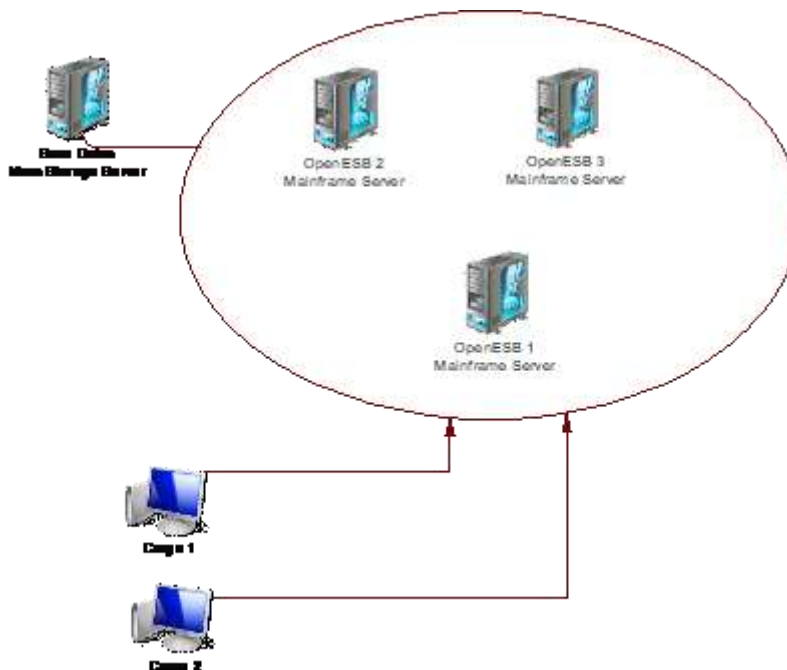


Figura 20. Escenario de prueba de escalabilidad.

En la tabla 8 describe los parámetros de carga y la configuración de clúster y de carga para la prueba:

Tabla 8. Pruebas de carga.

ID Prueba	Nodos del Cluster	Host de carga	Numero de Hilos por grupo	periodo entre grupo de Hilos (seg)	Número ejecuciones
P.2.1	1	1	120	30	3
P.2.2		1	180	20	6
P.2.3		1	240	10	9
P.2.4	2	2	240	20	3
P.2.5		2	380	10	6
P.2.6		2	500	5	9
P.2.7	3	2	600	20	3
P.2.8		2	900	10	6
P.2.9		2	1000	5	9

Con esas pruebas es posible establecer la evaluación de las pruebas de carga. Sin embargo como es requerido establecer una comparación entre ambientes SOA y no SOA será aplicado el mismo modelo de prueba a la solución AMedica de la Empresa ASMET Salud (considerando P.3.x las pruebas de punto inicial y P.4.x las pruebas de carga).

#### 4.7.1.2. Pruebas de stress.

Las siguientes progresiones matemáticas son empleadas para determinar la prueba de estrés sobre la arquitectura a fin de determinar los límites del sistema manteniendo constante el periodo entre hilos con un valor de 5 segundos.

- Numero de hilos por grupo:  $60 + \sum_{i=1}^n 60 * i$



- Número de ejecuciones:  $\sum_i^n 3i$

Tabla 9. Pruebas de Stress.

ID Prueba	Nodos del Cluster	Host de carga	Numero de Hilos por grupo	periodo entre grupo de Hilos (seg)	Número de veces ejecución
P.5.1	1	1	120	5	3
P.5.2		1	180	5	6
P.5.3		1	240	5	9
..		...	...	5	...
P.5.i		...	$60 + \sum_{i=1}^n 60 * i$	5	$\sum_i^n 3i$
...		...	...	5	...
P.5.n	...	$60 + \sum_{i=1}^n 60 * n$	5	$\sum_i^n 3n$	

La tabla 9 describe la prueba P.5.n donde más del 20% de la ejecución del proceso produce error de denegación de servicio. De esta misma forma puede procederse con las pruebas cuando en el clúster existen 2 nodos (para lo cual el ID de la prueba será P.6.x y pruebas P.7.x para 3 nodos).

El trabajo de diseñar y ejecutar pruebas de flexibilidad es una ardua labor que requiere de personal experto para determinar en el servidor de aplicaciones (Glassfish) diversas configuraciones de los nodos del clúster (número de sesiones concurrentes, número de conexiones del pool de conexiones, memoria de los objetos en el servidor, entre otros) para poder terminar las causas de la denegación de servicio. También requiere de la colaboración de Administradores de Bases de Datos (data Base Administrador, DBA) para gestionar los recursos técnicos de conexiones y disponibilidad de los sistemas de almacenamiento.

Por limitaciones de recursos y tiempo cada nodo del cluster solo presentará las siguientes configuraciones:

- **Memoria de dinámica:** configurada la memoria dinámica (Ramdon Access Memory, RAM) que es donde están establecidas los objetos del Directorio de Nombrado de Java (Java Naming and Directory Interface, JNDI) que inicia en 512Mb y podrá crecer hasta 1024Mb, Esta memoria es conocida comúnmente como base de Asignación de Memoria del Montón (Heap-Based Memory Allocation, o simplemente memoria heap).
- **Número de conexiones en el Pool de conexiones:** 60 Conexiones concurrentes por nodo.
- **Numero de sesiones permitidas:** "ilimitadas" (unlimited).

#### 4.7.2. ESCENARIOS DE EVALUACIÓN DE LA FLEXIBILIDAD.

Como fue expuesto anteriormente se han comparado la aplicación AMedicas de la empresa ASMET Salud EPS-S, frente al servicio de negocio Auditoria Médica soportado por la arquitectura prototipo. Para lograr estas actividades las fuentes de AMedicas han sido cambiadas de manera que este sistema pueda obtener los datos de las base de datos de Dinámica Gerencial. En estas





condiciones el documento de la factura será descargado en forma aleatoria de los disponibles. De igual forma con las modificaciones pertinentes en los servicios de negocio que inicialmente implementadas bajo un prototipo de fuente de información en PostgreSQL, para de esta forma establecer el número de cambios en las fuentes en ambos sistemas. Según lo descrito anteriormente como evolución de software, el problema modificado sería:

- Cambiar la fuente de información sobre los sistemas de información de los datos de epicirisis.

AMedicas está en proceso de ser migrada totalmente a Oracle Fusion Middleware 11g Build ADF (Application Developer Frametwork) Faces Clients for EJB and JPA. En este momento (febrero 2011) este ambiente de pruebas obtiene sus datos de los sistemas legados desarrollados en Firebird 2.1 y Delphi 7 y de imágenes de fax que encuentran el FTP del sistema de gestión documental y los nuevos registros de auditoria está en el marco de la construcción de los sistemas de información nuevos del proyecto SIA (Sistema de Información de Afiliación y Registro en el que uno de los autores de esta tesis es desarrollador activo) sobre tecnologías Oracle JAVA (ADF).

## **4.8. EVALUACIÓN DE LA ARQUITECTURA.**

En este punto están descritos los resultados de la evaluación de flexibilidad y escalabilidad.

### **4.8.1. PRUEBAS DEL ESCALABILIDAD.**

Las herramientas para realizar las pruebas de escalabilidad son:

JMeter: comportándose como un proxy para capturar y registrar solicitudes web y generar script XML que era convertido en algoritmos de carga. De esta herramienta son obtenidos los datos de cada solicitud y las metricas de Latencia (en milisegundos) y Throughout (en número de trasferencias por segundo que puede atender).

Quest Spotlight: permite monitorizar el uso de los recursos de un computador. Toma tres (3) medidas por prueba de carga para ser promediadas, de esta forma es obtenido el % actividad en el disco duro de acuerdo a la capacidad del bus de transferencia de datos y de memoria RAM.

#### **4.8.1.1. Establecimiento del punto de comparación (Benchmarking).**

Realizadas las pruebas descritas en la Tabla 7 sobre el primer nodo del cluster como esta en el Anexo A (en la fase de pruebas de servicio de la metodología SOA).

**Tabla 10. Resultado del Benchmarking.**

<b>ID Prueba</b>	<b>Latencia Prom. (mseg)</b>	<b>Throughout (TPS)</b>	<b>Disco Duro (%)</b>	<b>Memoria RAM (%)</b>
P.1.1	2358	158	34,2%	1.834MB (90,6 %)
P.1.2	2123	174	35,1%	1.724 MB (85,6 %)
P.1.3	1923	204	35,1%	1.867 MB (92,2%)



De esta forma es obtenido el nivel de aceptación de servicios que hacen parte de la los acuerdos de nivel de servicio (SLA) como muestra la siguiente tabla, sin considerar el uso de memoria RAM pues esta está configurada a no exceder 1024Mb y el consumo de disco por mostrar versiones despreciables.

**Tabla 11. Nivel de aceptación de Servicios (Benchmarking)**

Propiedad	Valor	Unidad
Latencia	2134	Milésimas de segundo
Throughout	600	Transferencias por segundo

Los datos de la tabla 11 obtenidos de promediar las latencias de la tabla 10 y el valor de Throughout es estimado para cargas grandes sobre un solo servidor ESB de 600 TPS.

#### 4.8.1.2. Pruebas de Carga.

Los resultados obtenidos luego de correr los algoritmos creado en JMeter y Quest Spotlight son presentados en la Tabla 12.

**Tabla 12. Pruebas de carga sobre la arquitectura.**

Nodos Cluster	ID Prueba	Latencia (ms)	Throughout (TPS)	Disco Duro %			Memoria RAM (%)		
1	P.2.1	1933	273	36,3			86,6 %		
	P.2.2	2223	452	36,7			88,3 %		
	P.2.3	2536	467	39.9			91,1%		
2	P.2.4	1702	587	46, 8%	38,3 %		91,2%	82,2%	
	P.2.5	1493	522	52,2%		38,7%	92,8%	74,6%	
	P.2.6	1587	656	46,2%	39.9%		84.2%	65,9%	
3	P.2.7	1202	684	66, 2%	45,6%	35,2%	92,2%	72,7%	82,5%
	P.2.8	1194	721	62,1%	58,2%	41,6%	96,5%	81,3%	61,2%
	P.2.9	1025	735	62,1%	56.2%	44.3%	93.3%	63,8%	62,1%

Estas mismas pruebas fueron realizadas en AMedica, no es considerado el empleo de más de un nodo de cluster dado que solo es posible hacer las pruebas sobre una sola instancia de dominio, es decir solo es posible emplear un nodo de cluster dado que está actualmente en el ambiente de desarrollo que consta de un solo dominio de Weblogic, no es viable instanciar servidores manejados dado que cada instancia de Weblogic cuesta alrededor de 25.000 dólares y en ambiente de desarrollo esto no está contemplado en los recursos de desarrollo y estas características están habilitadas a los servidores de producción de Asmet Salud EPS.

**Tabla 13. Pruebas de carga AMedicas.**

Nodos Cluster	ID Prueba	Latencia (ms)	Throughout (TPS)
1	P.2.1	3935	609
	P.2.2	2521	836
	P.2.3	2674	756

En la Tabla 13 no son presentados los datos de consumo de memoria y disco duro dado que otros aplicativos están actualmente sobre servidor de aplicaciones (SIA Afilados, SIA Sistema de Gestión documental, entre otros) y arrojarían datos no reales de consumo de recursos de



porcentaje de uso de disco duro y memoria RAM del sistema AMedicas. Es debido a que este equipo es un sistema de capacidades muy superiores a los que emplearon en las pruebas de la arquitectura de cluster de ESB.

#### **4.8.1.3. Pruebas de stress.**

Los resultados de las pruebas de stress aplicadas a la arquitectura para el cluster ESB con uno, dos y tres nodos son presentados en la tablas 14, 15 y 16 respectivamente.

**Tabla 14. Prueba de stres sobre la arquitectura un nodo en el cluster.**

<b>ID Prueba</b>	<b>Host de carga</b>	<b>Numero de Hilos</b>	<b>periodo entre Hilos</b>	<b>Número de veces ejecución</b>	<b>Denegación de servicio (%)</b>
P.5.1	1	120	5	3	0.0
P.5.2	1	180	5	6	0.
P.5.3	1	240	5	9	0.
P.5.4	1	300	5	12	0.0
P.5.5	1	360	5	15	0.0
P.5.6	1	420	5	18	0.0
P.5.7	2	480	5	21	0.02
P.5.8	2	540	5	24	0,3
P.5.9	2	600	5	27	2,1
P.5.10	2	660	5	30	2,4
P.5.11	2	720	5	33	3,1
P.5.12	2	780	5	36	3,3
P.5.13	3	840	5	39	12,1
P.5.14	3	900	5	42	13,7
P.5.15	3	960	5	45	16,6
P.5.16	3	1020	5	48	21,3

En la tabla 15 están consignados los resultados de la prueba de stress que fue realizada a partir de la iteración 20 (P.6.20) dado que en las pruebas iniciales no fueron encontrados indicios de saturación y como el objetivo de la prueba de stress es alcanzar los límites del sistema se procedió a adelantar las exigencias de las pruebas para lograr acercarse a los resultados esperados.

**Tabla 15. Prueba de stress sobre la arquitectura con 2 nodos en el cluster.**

<b>ID Prueba</b>	<b>Host de carga</b>	<b>Numero de Hilos</b>	<b>periodo entre Hilos</b>	<b>Número de veces ejecución</b>	<b>Denegación de servicio (%)</b>
P.6.20	3	1260	5	60	3,15
P.6.21	3	1320	5	63	2,45
P.6.22	3	1380	5	66	3,2
P.6.23	4	1440	5	69	3,45%
P.6.24	4	1500	5	72	4,16
P.6.25	4	1560	5	75	6,12
P.6.26	4	1620	5	78	6,12
P.6.27	4	1680	5	81	6,45
P.6.28	4	1740	5	84	8,65



P.6.29	4	1800	5	87	9,12
P.6.30	4	1860	5	90	7,34
P.6.31	4	1920	5	93	6,23
P.6.32	4	1980	5	96	5,13
P.6.33	4	2040	5	99	16,20
P.6.34	4	2100	5	102	14,20
P.6.35	4	2160	5	105	23,20

De igual forma como sucedió con el cluster ESB de 2 nodos fue realizada con el de tres nodos, iniciando las pruebas en la iteración 45 con los resultados de la tabla 16.

**Tabla 16. Prueba de stress sobre la arquitectura con 3 nodos en el cluster.**

ID Prueba	Host de carga	Numero de Hilos	periodo entre Hilos	Número de veces ejecución	Denegación de servicio
P.7.45	4	2760	5	135	4,20
P.7.46	4	2820	5	138	6,20
P.7.47	4	2880	5	141	6,20
P.7.48	4	2940	5	144	6,20
P.7.49	4	3000	5	147	7,70
P.7.50	4	3060	5	150	14,20
P.7.51	4	3120	5	153	10,60
P.7.52	4	3180	5	156	12,80
P.7.53	4	3240	5	159	16,65
P.7.54	4	3300	5	162	17,80
P.7.55	5	3360	5	165	21,70
P.7.56	5	3420	5	168	22,90

De esta forma son obtenidos los resultados de la prueba de stress:

**Tabla 17. Limites de la arquitectura.**

ID Prueba	Nodos del Cluster	Host de carga	Numero de Hilos	periodo entre Hilos	Número de veces ejecución	Denegación de servicio
P.5.16	1	3	1020	25	48	21,3%
P.6.35	2	4	2160	5	105	23,20%
P.7.55	3	5	3360	5	165	21,70%

En la tabla 17 está expuesta la parametrización de los algoritmos de carga empleados para alcanzar la denegación de servicio superior al 20% de las solicitudes realizadas contra el cluster ESB.

#### **4.8.2. PRUEBA DE FLEXIBILIDAD.**

Recoge los resultados de las pruebas de flexibilidad aplicadas al servicio de negocio Auditoria Medica y al aplicativo de AMedicas. El problema modificado es el siguiente:

Obtener los datos del epicrisis de las bases de datos del sistema de información Médica Gerencial.



Esto representa cambios comparables en las fuentes que componen la arquitectura orientada a servicios del sistema solución y en aplicativo AMedicas. Para reducir la complejidad de estas implementaciones fueron creadas Vistas<sup>20</sup> sobre la base de datos de Dinámica Gerencial de manera que correspondan a las entidades mapeadas con JPA como está en el Anexo F (Persistencia con JPA, F.3.2), es decir; creando vistas que corresponden al modelo físico de datos de la arquitectura solución de los servicios de tele-consulta y Auditoría. Estas clases son consideradas en las métricas a pesar que, como describe el Anexo F (Persistencia con JPA, sección F.3.2), son generadas por los IDE Netbeans y JDeveloper fácilmente, por lo tanto no fueron considerados como parte de la modificación. Para la medida de complejidad de la evolución no fueron consideradas las líneas de código que definen los POJOS (extender de la interfaces serializarle, constructor por defecto, métodos setter y getter, entre otros) por no representar un algoritmo como tal.

Los resultados del costo de evolución de los sistemas están consignados en la tabla 18. Como la evolución consta de un solo problema, el costo de evolución  $C(\epsilon)$  y complejidad de la evolución pueden obtenerse con operaciones escalares.

En los servicios de negocio de Auditoría no hace falta remapear las entidades si existe las vistas ajustadas al modelo de datos con el que fueron creadas, solo haría falta configurar un nuevo datasource y un pool de conexiones como es explicado en detalle en Anexo F, en tal caso son obtenidos los siguientes resultados.

**Tabla 18. Medidas del Costo de Evolución sin remapear las entidades usando JPA.**

Aplicativo	Numero de Clases Ajustadas		Número de Líneas de código en las clases ajustadas	Costo evolución	Complejidad de la evolución	
	Problema Anterior	Problema Modificado		$C(\epsilon)$	$O(Classes(\epsilon_{DS}))$	$O(Classes(\epsilon_{OP}))$
<b>Servicio de negocio Auditoria</b>	19	<b>2</b>	4	<b>9.6</b>	4	$O( DS )$
<b>AMedicas</b>	32	17	64	2.4	64	$O( DS )$

Volviendo a mapear las entidades existentes en los servicios de negocio (sin que sea necesario) son obtenidos los resultados de la tabla 19.

**Tabla 19. Medidas del Costo de Evolución remapeando las entidades usando JPA**

Aplicativo	Numero de Clases Ajustadas		Número de Líneas de código en las clases ajustadas	Costo evolución	Complejidad de la evolución	
	Problema Anterior	Problema Modificado		$C(\epsilon)$	$O(Classes(\epsilon_{DS}))$	$O(Classes(\epsilon_{OP}))$
<b>Servicio de negocio Auditoria</b>	19	<b>17</b>	4	<b>1.23</b>	4	$O( DS )$
<b>AMedicas</b>	32	17	64	2.4	64	$O( DS )$

Como es visible el costo de evolución cambia drásticamente a favor de AMedicas si son empleadas las utilidades de reutilización de código que ofrece JPA. Esto podría hacer pensar que

<sup>20</sup> En una base de datos es un resultado de una consulta SQL sobre una o varias tablas, también se puede considerar como una tabla virtual.



el modelo empleado no es adecuado y es analizada esta situación en la siguiente sección de este capítulo.

Con los resultados de la tabla 18 y 19 quedan concluidas las labores de medida de los requerimientos no funcionales de Escalabilidad y Flexibilidad.

#### **4.9. ANÁLISIS DE RESULTADOS.**

En esta sesión son analizados los resultados de las pruebas de escalabilidad y flexibilidad aplicadas descritas en la sección anterior.

##### **4.9.1. ANÁLISIS DE LOS RESULTADOS DE LAS PRUEBAS DE ESCALABILIDAD.**

###### **4.9.1.1. Análisis en las Pruebas de Carga.**

En este punto son analizados los resultados de las pruebas de carga, cabe mencionar que una forma exhaustiva de análisis de la información requiere de dejar variable uno de los parámetros de configuración de los algoritmos de carga y variar cada uno de estos parámetros para hacer un análisis de comparativo de cada algoritmo. Dado el tiempo y esfuerzo que demanda esta tarea fueron definidos escenarios de carga identificados por el Id de prueba, donde cada prueba representa un nivel de exigencia superior al anterior. Sin embargo para obtener una relación del nivel de exigencia de la prueba es calculado el número total de hilos ( obtenido de multiplicar el número de host de carga el número de hilos y el número de ejecuciones) de cada prueba y el tiempo total de la carga (período entre hilos por el número de ejecuciones). De esta forma para las pruebas de la Tabla 8 tendrá los siguientes valores.

**Tabla 20. Niveles de carga en la pruebas de carga.**

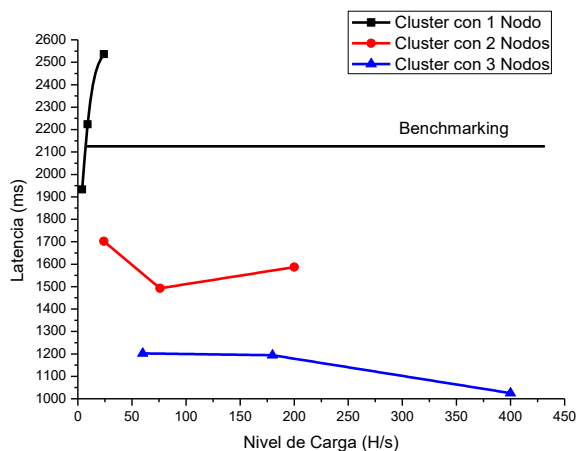
<b>ID Prueba</b>	<b>Nodos del Cluster</b>	<b>Host de carga</b>	<b>Numero de Hilos por grupo</b>	<b>periodo entre grupo de Hilos (s)</b>	<b>Número ejecuciones</b>	<b>Número total de Hilos</b>	<b>Tiempo de la prueba</b>	<b>Nivel de carga (hilos/seg)</b>
P.2.1	1	1	120	30	3	360	90	<b>4</b>
P.2.2		1	180	20	6	1080	120	<b>9</b>
P.2.3		1	240	10	9	2160	90	<b>24</b>
P.2.4	2	2	240	20	3	1440	60	<b>24</b>
P.2.5		2	380	10	6	4560	60	<b>76</b>
P.2.6		2	500	5	9	9000	45	<b>200</b>
P.2.7	3	2	600	20	3	3600	60	<b>60</b>
P.2.8		2	900	10	6	10800	60	<b>180</b>
P.2.9		2	1000	5	9	18000	45	<b>400</b>

De esta forma es posible establecer una comparación de la configuración de la arquitectura (número nodos en el cluster) y el nivel de exigencia de la prueba.

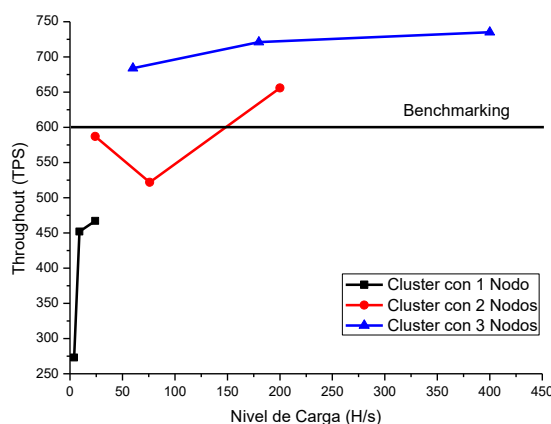
Como lo presenta la figura 21 el sistema carga con un nodo en el cluster ESB a mayor número de solicitudes de servicios mayor también el tiempo de latencia, como comúnmente es esperado este



tipo de sistemas. Sin embargo cuando es ampliado a dos el número de nodos en el cluster y el nivel de carga la curva presenta un comportamiento aparentemente extraño (gráfica de cluster con dos nodos), pues la curva a mayor carga tiende a responder más rápido y luego toma el comportamiento normal de retardar los tiempos de respuesta. En la curva del cluster ESB con tres nodos presenta un comportamiento aún más pronunciado a este respecto, estos comportamientos obedecen a que al estar basados en EJB los servicios de negocios, por cada solicitud de servicio no son instanciados nuevos recursos (EJB de sesión) sino que es el contenedor de EJB que administra los Beans que están en la memoria Heap. Un comportamiento similar puede ser observado en la figura 22, a excepción del throughput del cluster con tres nodos que ligeramente tiene a crecer.



**Figura 21. Latencia en las pruebas de carga**



**Figura 22. Throughput en las pruebas de carga.**

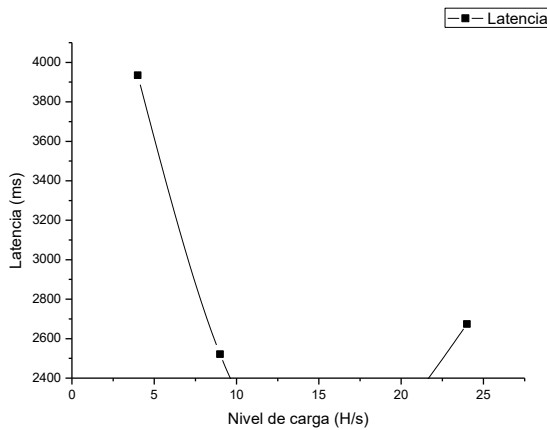
De esta forma puede inferirse que el sistema ESB presenta un comportamiento que lo hace hábil para atender crecientes número de solicitudes siempre que puedan ser mejoradas las configuraciones del mismo.

Las siguientes figuras muestran los resultados aplicados al sistema AMedicas aunque los valores que describen la curva son diferentes muestran en común que sus necesidades son ajustadas conforme aumentan los niveles de carga. Nótese que a diferencia del sistema de las curvas anteriores el sistema AMedicas con un solo nodo empieza a describir el comportamiento del cluster ESB tiene con dos o tres nodos, esto obedece a que las características de este equipo no son comparables con los equipos que componen el cluster ESB al ser de muy superiores capacidades (8 núcleos Xeon y 16 Gigas de RAM).

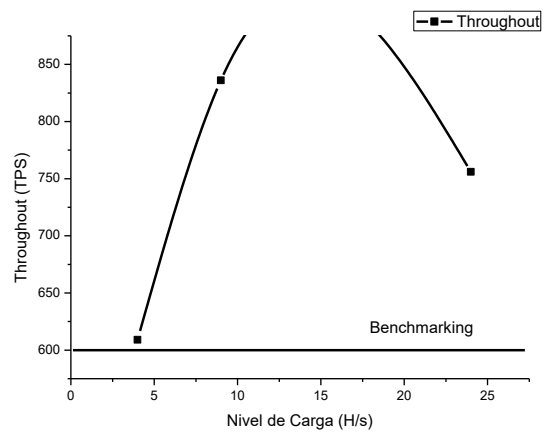
Las curvas de los resultados de las pruebas de carga en la arquitectura y en el sistema AMedicas muestran claramente que el sistema AMedicas está en capacidad de cumplir con un solo servidor los niveles de servicios esperados (Benchmarking) mientras que la arquitectura no, esto obedece a que las características de los servidores que componen el cluster ESB de la arquitectura no son comparables en capacidad de procesamiento de datos que tiene el servidor donde está desplegado el sistema AMedicas. De esta forma es de esperar que si mejoran las capacidades de



los nodos del cluster ESB también estaría en capacidad de alcanzar los niveles de servicio esperados.



**Figura 23. Latencia en las pruebas de Carga en AMedicas**

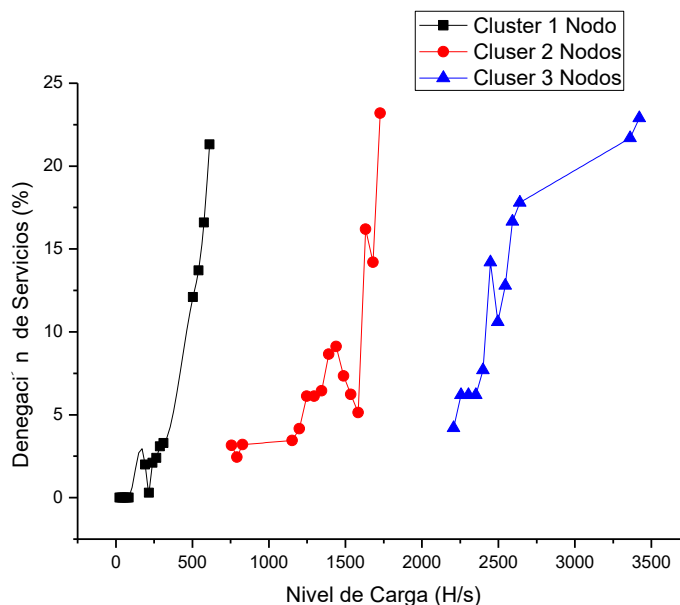


**Figura 24. Throughput en las pruebas de Carga en AMedicas.**

#### **4.9.1.2. Análisis en las Pruebas de Strees.**

Un análisis comparativo entre las distintas configuraciones del cluster ESB es presentado en la gráfica de la figura 25. En ella puede observarse los diferentes niveles de carga en los que es estresado el sistema donde más del 20% de las trasferencias de información son canceladas o denegadas.





**Figura 25. Pruebas de Stress.**

La Figura 25 muestra claramente que existen límites en la prestación de servicios y también pueden ser mejorados considerablemente estos límites si la configuración de cluster ESB es escalado a dos o tres nodos, de esta forma mientras el cluster ESB con un nodo es estresado con un nivel de carga de 612 hilos por segundo (156.060 hilos en 4,25 minutos) el clúster con dos nodos llega a saturación de igual forma con un nivel de carga de 1.728 hilos por segundo (933.120 hilos en 9 minutos), es decir un nivel de carga que equivale 2,8 veces el nivel de carga que el del cluster ESB con un nodo. De esta forma el cluster ESB con 3 nodos es estresado a un nivel de carga de 3.420 hilos por segundo (2'924.100 hilos en 14,25 minutos) que equivale a 5,5 veces y 1,98 veces el nivel de carga del cluster ESB con un nodo y el cluster ESB con dos nodos respectivamente.

De lo anterior puede concluirse que el sistema está capacitado para crecer, aumentando las características que mejoran el rendimiento y es considerado escalable, agregando nodos al cluster ESB o mejorando las características de los servidores.

#### **4.9.2. ANÁLISIS DE LOS RESULTADOS DE LA PRUEBA DE FLEXIBILIDAD.**

Los resultados de las tablas 18 y 19 muestran una situación interesante de analizar. Por una parte el empleo de JPA para mapear los modelos de datos ofrecen la capacidad de cambiar de fuente de datos (motor de almacenamiento) fácilmente si cumplen determinadas condiciones (que existan estructuras idénticas en los motores) que obtienen creando las vistas apropiadas. Como resultado de aplicar esta técnica en la arquitectura solución (creando un DataSource y nuevo pool de conexiones) en el problema modificado solo requiere de que sean ajustadas solo dos clases, mientras que en el sistema AMedicas requiere de mapear todas las entidades con lo que resulta que el número de clases ajustadas son 14 (las mapeadas) más 3 clases BeakingBean (las clases



BeakinBean son las clases de donde son instanciadas los Beans de sesión que comunican la vista y el controlador en el framework Oracle ADF) en total 17 clases.

De lo anterior puede concluirse que el costo de evolución es menor en el servicio de negocio Auditoria en la arquitectura SOA solución, sin embargo el número de clases que corresponden a la implementación del problema anterior  $p_{anterior}$  en AMedica es mayor que el del servicio de negocio Auditoria lo que da lugar a un costo de evolución mayor en el servicio de negocio Auditoria. para evidenciar esto nuevamente fueron mapeadas las entidades (sin ser necesario) del modelo en el servicio de negocio de manera que el número clases afectadas suman igualmente 17 como lo muestra la tabla 18 y en donde el costo de evolución muestra un menor valor en el servicio de negocio Auditoria que en el sistema AMedicas.

Lo anterior evidencia que el modelo empleado es ajustado a los pasos de la evolución del software que son posteriores a la construcción del aplicativo donde la implementación de ajustes anterior ( $i_{anterior}$ ) requiere menor número de clases ajustadas que en la construcción de desde cero de un sistema de información. Es decir que el modelo de Eden es apropiado para pasos de la evolución del software intermedio o posterior a la construcción inicial de un sistema informático (entre un ajuste del sistema y otro posterior).

De esta forma el hecho de que el número de clases ajustadas sea menor en el servicio de negocio auditoria nos permite concluir que es más flexible que el sistema AMedicas, que la estimación mediante la métrica empleada.

La métrica de complejidad de la evolución de software muestra claramente que en los algoritmos de ajuste en AMedicas presentó más complejidad que los del servicio de negocio Auditoria. De esta forma puede concluirse que el valor de la métrica cuantifica el esfuerzo empleado en los ajustes.

#### **4.10. CONCLUSIONES**

En este capítulo fueron descritos los procesos de negocio que soportan la arquitectura solución, incluyendo el mensaje de validación de uso del estándar HL7. También describe los sistemas legados del Hospital Susana López de Valencia que interactuaran con la arquitectura. Define los escenarios de prueba de los requerimientos no funcionales de flexibilidad e interoperabilidad basados en experiencia de empresas como Microsoft y Oracle. Las métricas para evaluar flexibilidad fueron latencia y throughout, mientras que las métricas para flexibilidad fueron empleadas métricas de costo de evolución del software. De los resultados puede concluirse que la arquitectura orientada a servicio basada en ESB es más flexible que Amedica, también que la arquitectura propuesta es altamente escalable.



## **5. CAPITULO V. CONCLUSIONES Y RECOMENDACIONES.**

Este capítulo presenta las conclusiones y recomendaciones de trabajos futuros sugeridos luego del desarrollo de este trabajo de grado.

### **5.1. CONCLUSIONES.**

A continuación son expuestas las conclusiones extractadas del trabajo realizado alrededor de la propuesta de interoperabilidad para procesos de negocio en sistemas de información hospitalarios bajo una Arquitectura Orientada a Servicios en el sector salud.

- Es importante que exista una conciencia de la necesidad de analizar estándares y recomendaciones al emprender proyectos en sistemas telemáticos, en particular en el ámbito de la salud, ya que es en un ambiente interinstitucional fortalecido por los medios de comunicación electrónicos y ellos necesitan operar información conjunta y procesos entre organizaciones de ámbito nacional e internacional.
- Establecer un enfoque metodológico es importante en las fases previas a la puesta en marcha de proyectos en el ámbito de las tecnologías de la información y comunicaciones, incluyendo en la implementación de arquitecturas SOA a pesar que para esta tecnología son poco utilizadas, ya que estos enfoques proveen mecanismos procedimentales y de control que recogen la experiencia colectiva y estimulan el empleo de las mejores prácticas en el despliegue de sistemas tecnológicos.
- Hacer una correcta selección de tecnologías, protocolos y estándares es una imperante necesidad en el establecimiento de sistemas informáticos altamente compatibles. En orden a esto, las tecnologías y protocolos abiertos ofrecen un amplio margen de alternativas; sin embargo, no todas las enfocadas pueden ser aplicables al dominio de aplicación, de esta forma es necesario establecer recursos para su estudio y análisis en las etapas iniciales de del desarrollo.
- Existe una amplia gama de productos ESB en el mercado para el establecimiento de una SOA de libre distribución que permita el empleo de estándares abiertos para tal fin, ello fue posible de evidenciar en el anexo B; sin embargo, hacer una selección rigurosa es una tarea ardua que demanda muchas actividades. Es deseable profundizar en el establecimiento de un ambiente de pruebas de campo en las diferentes posibilidades presentes, de manera que permitan una selección crítica y basada en evidencias medibles.
- Existen varias formas de desplegar nuevos servicios web sobre una arquitectura JEE, sin embargo la inclusión de las principales características de JEE como son los EJB y la persistencia mediante JPA, simplifican mucho la labor de elaboración de servicios de negocio, fundamentalmente por que permiten concentrarse en la lógica del servicio más que en los detalles de la implementación o el acceso a datos provenientes de diversas fuentes y arquitecturas de datos, de igual forma el rendimiento de los servicios web son afectados directa y positivamente al ser gestionados por contenedores de EJB, estos administran el encapsulado de los componentes de negocio.
- El uso de HL7 es viable en sistemas informáticos ya que permite compartir en forma sencilla la información entre sistemas informáticos de las instituciones en el sector salud.



- La labor de realizar pruebas de desempeño es fundamental en la selección de una arquitectura y según la experiencia adquirida consume un 40% del trabajo total del proyecto. Ello afecta directamente la selección adecuada de componentes arquitectónicos para grandes sistemas informáticos.
- Este trabajo demuestra que la selección de tecnologías de libre distribución es una buena alternativa para desarrollar sistemas altamente escalables y flexibles. Sin embargo es necesario evaluar de manera crítica la actividad de la comunidad libre ya que existen alternativas muy prometedoras en su especificación pero poco activas en su trabajo de desarrollo. Ello compromete su viabilidad y capacidad de sostenimiento.
- Este proyecto demuestra que el uso de un enfoque arquitectónico basado en Buses Empresariales de Servicios (ESB) es una buena alternativa para el establecimiento de una Arquitectura Orientada a Servicios.
- El uso de OpenESB como plataforma tecnológica de un Bus de Servicios Empresariales es una buena alternativa para empresas en el sector de la salud ya que es flexible, escalable y capaz de responder a las necesidades de las empresas de la industria de la salud.

## **5.2. RECOMENDACIONES Y TRABAJOS FUTUROS.**

Durante el desarrollo de este trabajo fueron identificadas diversas alternativas tecnológicas para alcanzar sus objetivos propuestos. Los siguientes trabajos futuros son sugeridos como consecuencia de estos estudios.

- **Estudio comparativo de las metodologías disponibles para el estableciéndote una Arquitectura Orientada a Servicios.**

Este trabajo abordó un estudio de las principales características de las metodologías de referencia SOA y SOMA, seleccionando una de ellas ya que ofrece recursos prácticos para los miembros del departamento de sistemas del HSLV. Sin embargo es deseable profundizar en las particularidades de cada una de ellas y contrastar estos enfoques metodológicos para poder tener un criterio objetivo de cuál puede ser la mejor alternativa metodológica para el HSLV.

- **Adaptación del marco de trabajo HIS-DF para el desarrollo de Arquitecturas Orientadas a Servicios.**

HIS-DF ofrece un conjunto de herramientas para diseñar sistemas de información y en este trabajo fue empleado para describir la arquitectura del sistema desde diferentes puntos de vista, desde un nivel de abstracción a las tecnologías de la implantación y también en los aspectos técnicos de la implementación; sin embargo, el enfoque de análisis y diseño orientado a servicios exige una adaptación de HIS-DF que le permita establecer los roles y artefactos de estos procesos.

- **Estudio de otras alternativas de Buses Empresariales de Servicios.**

Este trabajo emplea OpenESB, como Bus de Servicios Empresariales; sin embargo, existen otras herramientas maduras como son: ChainBulder, Petals ESB, JBoosESB, entre otras. Ellas ofrecen múltiples herramientas y soporte, la documentación estudiada a este respecto



demonstró que estas herramientas también pueden ser una alternativa válida para el establecimiento de una SOA en el sector salud.

- **Estudio comparativo de diferentes arquitecturas de procesamiento distribuido para dotar a los sistemas informáticos de alta disponibilidad.**

Los exigentes requerimientos no funcionales de las empresas en el sector salud son una realidad que impulsa la labor de establecer estudios de arquitecturas distribuidas para el procesamiento distribuido. Ello sugiere profundizar en el análisis de las diferentes técnicas de alta disponibilidad existentes en la actualidad.

- **Estudio de las métricas existentes para medir la flexibilidad en Arquitecturas Orientadas a Servicios.**

Las arquitecturas orientadas a servicio establecen un nuevo paradigma en cuanto al diseño e implementación de sistemas informáticos, la orientación servicios web, la reutilización de procesos y la composición de los mismos establecen mecanismos que hacen flexibles las soluciones soportadas por estas arquitecturas. En estos nuevos mecanismos de desarrollo de soluciones es posible estudiar el nivel de flexibilidad que ofrecen dadas las características de granularidad y acoplamiento de servicios, de manera que permitan influir en el análisis y diseño orientado a servicios.



## REFERENCIAS

**Aguilar, Ricardo Adolfo y Amanecer, Urruria. 2009.** Implementación del estandar HL7 para la notificación de eventos de interés en salud pública. Popayán : s.n., 2009. Trabajo de grado presentado como requisito para optar al título de Ingeniero en Electrónica y Telecomunicaciones.

**Ali Arsanjani. 2004.** IBM. [En línea] 9 de Noviembre de 2004. [Citado el: 12 de Noviembre de 2009.] <http://www.ibm.com/developerworks/library/ws-soa-design1/>.

**BEEP. 2009.** The Blocks extensible Exchange Protocol. [En línea] 1 de 04 de 2009. [Citado el: 26 de 04 de 2010.] <http://www.beepcore.org>.

**Bieberstein, Norbert, y otros. 2008.** *Executing SOA: A Practical Guide for the Service-Oriented Architect*. Crawfordsville : IBM Press, 2008. 978-0-13-714947-6.

**BLOBEL, Bernd. 2006.** EHR-Architectures-Comparisons and Trends. Regensburg, Alemania : s.n., 2006.

—. *EHR-Architectures-Comparisons and Trends*. Regensburg, Alemania : s.n.

**California Health Care Foundation. 2004.** *California Clinial Data Project: Setting Standards*. Oakland : s.n., 2004. pág. 2.

**Cámpoli, Marcela. 2005.** Sistemas de Información ene l sector salud. [En línea] Agosto de 2005. [Citado el: 16 de Junio de 2009.] <http://www.gestiopolis.com/canales5/mkt/simsalud.htm>.

**ChainBuilder. 2010.** ChainBuilder. [En línea] 2010. [Citado el: 23 de Mayo de 2010.] <http://www.chainforge.net/>.

**Chamberland, Luc, y otros. 2010.** *IBM Business Process Management Reviewer's Guide*. s.l. : ibm.com/redbooks, 2010. <http://www.redbooks.ibm.com/redpapers/pdfs/redp4433.pdf>.

**Colombia, Congreso de la República de. 2001.** *Ley 715 de Diciembre 21 de 2001*. 2001.

**de la Torre, César. 2008.** *Arquitectura SOA con tecnología Microsoft*. Madrid : Editorial Krasis Press, 2008. 978-84-935489-7-1.

**Delgado, Andrea, y otros.** *Metodologías de Desarrollo para Service Oriented Architecture con Rational Unified Process*. [www.isa.us.es/downloads/proceedings/0126.pdf] Montevideo : s.n.

**Eden, Ammon H. y Mens, Tom. 2006.** Measuring Software Flexibility. Junio de 2006, Vol. 153. [http://www.eden-study.org/articles/2006/measuring-sw-flexibility\\_ieesw.pdf](http://www.eden-study.org/articles/2006/measuring-sw-flexibility_ieesw.pdf).

*E-health is the way via SOA*. **Melrose, J Peter. 2007.** 3, Minneapolis : s.n., Marzo de 2007, Healthcare Financial Management, Vol. 61, pág. 120.



**Endrei, Mark, y otros. 2004.** Patterns: Service Oriented Architecture. [En línea] Abril de 2004. [Citado el: 08 de Julio de 2010.] [www.redbooks.ibm.com/redbooks/pdfs/sg246303.pdf](http://www.redbooks.ibm.com/redbooks/pdfs/sg246303.pdf). SG24-6303-00.

**Erl, Thomas. 2005.** *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR : Prentice Hall PTR, 2005. 0-13-185858-0.

**Firat Kart, Louise E. Moser, and P. Michael Melliar-Smith. 2009.** *Building a Distributed E-Healthcare System Using SOA*. s.l. : IEEE Xplore, 2009. 1520-920 /08/.

**García, Javier. 2009.** Javier García León. [En línea] 16 de Diciembre de 2009. [Citado el: 24-30 de Febrero de 2010.] <http://javiergarcialeon.wordpress.com/2009/12/16/interoperabilidad-y-salud/>.

**GlassFish ESB. 2010.** GlassFish ESB. [En línea] 2010. [Citado el: 21 de Enero de 2010.] <https://open-esb.dev.java.net>.

**Gobierno en Línea. 2008.** Gobierno en Línea. [En línea] 18 de June de 2008. [Citado el: 2 de October de 2009.] <http://lenguaje.intranet.gov.co/web/gelxml/queesgelxml>.

**HealthCare Services Specification Project-HSSP.** HSSP Project. [En línea] [Citado el: 15 de Octubre de 2009.] <http://hssp.wikispaces.com>.

**Heitmann, Kai U., Bernd, Blobel y Dudeck, Joachim. 1999.** *HL7 Communication standrd in medicine*. s.l. : Cologne, 1999. ISBN 3-933819-06-7.

**HL7. 2007.** hl7.org. [En línea] 2007. <http://www.hl7.org/about/index.cfm?ref=nav>.

**HL7; OMG. 2008.** HSSP - Guia Practica. [En línea] 2008. [Citado el: 12 de Enero de 2010.] <http://hssp.wikispaces.com/PracticalGuide>.

**HSSP. 2009.** HSSP Standards. [En línea] 8 de Diciembre de 2009. [Citado el: 12 de 2 de 2010.] <http://hssp.wikispaces.com/standards>.

—. 2009. *The Practical Guide for SOA in Health Care*. 2009.

**IBM Corp. 2007.** *Building SOA Solutions Using the Rational SDP*. [<http://www.redbooks.ibm.com/abstracts/sg247356.html>] s.l. : RedBooks, 2007.

**IBM. 2010.** IBM WebSphere Enterprise Service Bus (ESB). [En línea] 2010. [Citado el: 20 de Mayo de 2010.] <http://www-01.ibm.com/software/ar/info/middleware/applications/>.

**IEEE. 2000.** IEEE Recommended Practice for Architectural Description of Software-Intensive Systems. [En línea] 21 de Septiembre de 2000. [Citado el: 12 de Enero de 2010.] <http://www.win.tue.nl/~johan/educ/2II45/Lit/software-architecture-std1471-2000.pdf>. ISBN 0-7381-2519-9.

**Intel. 2010.** Intel SOA Expressway. [En línea] 2010. <http://www.intel.com/about/companyinfo/healthcare/products/soa/index.htm>.



—. **2010.** Inter.com. [En línea] 2010. [Citado el: 23 de Mayo de 2010.] <http://www.intel.com/about/companyinfo/healthcare/products/soa/index.htm>.

**ITÄLÄ , Timo, y otros.** *SOA approach for integration of departmental systems. eHealth Beyond the Horizon.*

**López, Diego Mauricio.** **2009.** *Interoperable Architectures for Advanced Health Information Systems. PhD thesis.* University of Regensburg, Germany : s.n., 2009.

**Matsumara, Miko, Brauel, Bjoern y Shah, Jignesh.** **2009.** *Adopción de SOA para Dummies.* Indianápolis, Indiana : Wiley Publishin, 2009. 978-0-470-48334-3.

**Meier, J.D., y otros.** **2007.** *Performance Testing Guidance for Web Applications.* 2007.

**Microsoft Corporation.** **2006.** *Enabling Real World SOA through the Microsoft Platform.* Diciembre : s.n., 2006. Disponible en: <http://download.microsoft.com/download/b/4/d/b4db580a-0361-4907-9a6e-9d2866d8b581/Real%20World%20SOA.doc>.

**Ministerio de Educación Nacional.** **2002.** Ministerio de Educación. [En línea] 2002. [Citado el: 23 de January de 2010.] [http://www.mineducacion.gov.co/1621/articles-86098\\_archivo\\_pdf.pdf](http://www.mineducacion.gov.co/1621/articles-86098_archivo_pdf.pdf).

**Ministerio de la Protección Social.** **2008.** SISPRO Sistema Integral de Información de la Protección Social. [En línea] 2008. [Citado el: 15 de May de 2009.] <http://www2.sispro.gov.co/default.aspx>.

**Minoli, Daniel.** **2008.** *Enterprise Architecture A to Z.* s.l. : Taylor & Francis Group, 2008. 978-0-8493-8517-9 .

**Oasis.** **2007.** oasis-open. [En línea] 02 de Mayo de 2007. [Citado el: 21 de Marzo de 2010.] <http://www.oasis-open.org/committees/download.php/23964/wsbpel-v2.0-primer.htm>.

**Object Management Group - OMG.** **2010.** OMG Certified Expert in BPM™ Overview. [En línea] 06 de Febrero de 2010. [Citado el: 23 de Marzo de 2010.] <http://www.omg.org/oceb/>.

**OMG.** **2010.** About The Object Management Group™ (OMG™). <http://www.omg.org>. [En línea] 27 de 08 de 2010. [Citado el: 12 de 08 de 2010.] <http://www.omg.org/gettingstarted/gettingstartedindex.htm>.

—. **2010.** OMG. [En línea] Febrero de 2010. <http://www.omg.org/news/releases/pr2010/02-04-10.htm>.

**Oracle.** **2010.** *Introduction to Performance Monitoring.* s.l. : Oracle, 2010. Tomada de las presentaciones del curso Oracle University de administración de weblogic.

—. **2010.** oracle.com. [En línea] 2010. <http://www.oracle.com/us/technologies/soa/service-bus-066459.html>.

**OSGi.** **2010.** OSGI. [En línea] 2010. <http://www.osgi.org>.





**osESB. 2007.** *osESB Documentation, Part 1: The SOA Stack and the ESB.* 2007.

**OW2 Middleware Consortium.** OW2. [En línea] [Citado el: 8 de June de 2010.] <http://www.ow2.org>.

**Proyecto GEL-XML. 2008.** GEL-XML. [En línea] 2008. [Citado el: 2006 de Octubre de 2009.] <http://gelx.l.igob.gov.co/web/gelxml/inicio>.

**Ryan, A, P., Eklund y B., Esler. 2007.** *Toward the Interoperability of HL7 v3 and SNOMED CT: a case study modeling mobile clinical treatment.* Amsterdam : MedInfo, 2007.

**Ryan, Amanda. 2008.** *A Framework for Semantic Interoperability in Healthcare: A Service Oriented Architecture based on Health Informatic Standards.* s.l. : Organizing Committee of MIE, 2008.

**SRI. 2008.** [En línea] 2008. [www.sri.com](http://www.sri.com).

**The Open Group. 2008.** [theopengroup.org](http://www.theopengroup.org). [En línea] 2008. <http://www.theopengroup.org>.

**TIBCO. 2008.** *El papel de un Bus de Servicios Empresariales (ESB) en una SOA.* 2008.

**United Nations Economic and Social Commission for Asia and the Pacific. 2007.** [En línea] October de 2007. [Citado el: 10 de September de 2009.] <http://www.unescap.org/esid/hds/lastestadd/eHealthICTpaper041007.pdf>.

**Urquhart, A. 2004.** *The Blackwell Guide to Philosophy of Computing and Information.* Oxford: Blackwel : L. Floridi, 2004.

**Vallecillo, Antonio. 2001.** RM-ODP: The ISO Reference Model for Open Distributed Processing. [aut. libro] Thomas E. Vollman. *An Overview of Standards for CASE Tool and Process Automation.* Madrid : DINTEL Edition on Software Engineering, 2001. Disponible en: <http://www.lcc.uma.es/~av/Publicaciones/00/odpeng.pdf>.

**W3C. 2010.** Guia Breve de Servicios Web. [En línea] 06 de Junio de 2010. [Citado el: 16 de Agosto de 2010.] <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>.