

**UNIVERSIDAD CENTROCCIDENTAL
“LISANDRO ALVARADO”**

**ARQUITECTURA ORIENTADA A SERVICIOS PARA LA
INTEROPERABILIDAD CON EQUIPOS MEDICOS.
CASO DE ESTUDIO LABORATORIO CLINICO MASCIA.**

CARMARY ANDREA MENDEZ TORRES

Barquisimeto, 2011

**UNIVERSIDAD CENTROCCIDENTAL “LISANDRO ALVARADO”
DECANATO DE CIENCIAS Y TECNOLOGIA
POSTGRADO EN CIENCIAS DE LA COMPUTACION**

**ARQUITECTURA ORIENTADA A SERVICIOS PARA LA
INTEROPERABILIDAD CON EQUIPOS MEDICOS.
CASO DE ESTUDIO LABORATORIO CLINICO MASCIA.**

Proyecto de Trabajo de Grado presentado para optar al título de Magister Scientiarum
en Ciencias de la Computación

Por: **CARMARY ANDREA MENDEZ TORRES**

Barquisimeto, 2011



UNIVERSIDAD CENTROCCIDENTAL
"LISANDRO ALVARADO"
DECANATO DE CIENCIAS Y TECNOLOGIA
COORDINACION DE POSTGRADO

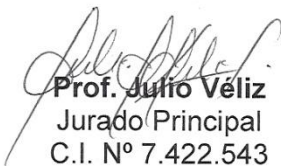
ACTA VEREDICTO TRABAJO DE GRADO


Nosotros, Miembros del Jurado Examinador del Trabajo de Grado titulado: "ARQUITECTURA ORIENTADA A SERVICIOS PARA INTEROPERABILIDAD CON EQUIPOS MÉDICOS. CASO DE ESTUDIO LABORATORIO CLÍNICO MASCIA", presentado por la Ingeniera **CARMARY ANDREA MÉNDEZ TORRES**, titular de la Cédula de Identidad N° 15.875.668, como requisito para optar al grado académico de **MAGÍSTER SCIENTIARUM EN CIENCIAS DE LA COMPUTACIÓN**, ofrecido por el programa de Maestría en Ciencias Computación del Decanato de Ciencias y Tecnología de la Universidad Centroccidental "Lisandro Alvarado", hacemos constar que hoy siete de julio del año dos mil once (07/07/2011) a las tres y treinta de la tarde (3:30 p.m.), se realizó examen Público de Defensa de Trabajo de Grado, de acuerdo a lo establecido en la Normativa de Trabajos de Grado de la UCLA. Una vez rendido el examen, este Jurado emite el siguiente veredicto:

El Trabajo de Grado fue:

APROBADO

Dando fe de ello, levantamos la presente acta en la ciudad de Barquisimeto a los siete días del mes de julio del año dos mil once.


Prof. Julio Véliz
Jurado Principal
C.I. N° 7.422.543


Prof. Jesús Guédez
Tutor - Presidente
C.I. N° 7.426.621




Prof. Jorge Pérez
Jurado Principal
C.I. N° 7.444.662

INDICE GENERAL

	PAG.
ÍNDICE GENERAL.....	iv
ÍNDICE DE TABLAS.....	v
ÍNDICE DE FIGURAS.....	vi
RESUMEN.....	viii
INTRODUCCIÓN.....	1
 CAPÍTULO	
I EL PROBLEMA.....	4
Planteamiento del Problema.....	4
Objetivos de la Investigación.....	14
Objetivo General.....	14
Objetivos Específicos.....	14
Justificación e Importancia de la Investigación.....	14
Alcances y Limitaciones.....	16
 II MARCO TEÓRICO.....	 17
Antecedentes de la Investigación.....	17
Bases Teóricas.....	23
Bases Legales.....	49
Sistema de Variables.....	51
Operacionalización de las Variables.....	51
 III MARCO METODOLÓGICO.....	 54
Naturaleza de la Investigación.....	54
Estudio de Factibilidad.....	55
Procedimiento de la Investigación.....	57
Técnicas e Instrumentos de Recolección de Datos.....	57
 IV PROPUESTA.....	 59
Introducción.....	59
Exploración.....	61
Planificación de Entregas.....	69
Iteración 1.....	72
Iteración 2.....	84
Iteración 3.....	89
Iteración 4.....	99
 V CONCLUSIONES Y RECOMENDACIONES.....	 104
Conclusiones.....	104
Recomendaciones.....	106
 REFERENCIAS BIBLIOGRÁFICAS.....	 108

ÍNDICE DE TABLAS

TABLA		PAG
1	Definición Conceptual de la Variable.....	52
2	Operacionalización de las Variables.....	53
3	Planificación de Entregas.....	71
4	Patrones requeridos ESB para Arquitectura SOA.....	79
5	Descripción de Paquetes Componente de Datos.....	83
6	Descripción de Paquetes Componente de Integración.....	83
7	Descripción de Paquetes Componente de Lógica de Negocio.....	84
8	Definición de Servicio.....	102

ÍNDICE DE FIGURAS

FIGURA		PAG
1	Recursos de Redes.....	26
2	Arquitectura de un SIL.....	29
3	Flujo Unidireccional	31
4	Flujo Bidireccional	31
5	Flujo Consulta Bidireccional	32
6	Jerarquía de un Mensaje ASTM E1394	34
7	Definición del Mensaje ASTM E1394 – Encabezado.....	34
8	Definición del Mensaje ASTM E1394 - Paciente.....	35
9	Definición del Mensaje ASTM E1394 - Orden.....	35
10	Definición del Mensaje ASTM E1394 - Resultado.....	36
11	Definición del Mensaje ASTM E1394 - Terminal.....	36
12	Modelo 4 + 1 Vistas.....	44
13	Diagrama de Contexto.....	62
14	Esquema de la Propuesta de Solución.....	66
15	Diagrama de Casos de Uso de SOA-Médico.....	67
16	Diagrama de Casos de Uso de SOA-Médico.....	67
17	Diagrama de Casos de Uso de SOA-Médico.....	70
18	Aproximación Arquitectural	73
19	Arquitectura en Capas de SOA-Médico.....	74
20	Escenario de Interacción Arquitectura.....	76
21	Diagrama de Despliegue SOA-Medico.....	78
22	Diagrama de Despliegue con Tecnologías SOA-Médico.....	82
23	Casos de Uso Gestión para Información básica SOA-Médico.....	85
24	Diagrama de Clases paquete com.soamedico.dato.dto.....	86
25	Diagrama de Clases paquete com.soamedico.dato.dao y com.soamedico.dato.utilidades.....	88

26	Diagrama de Clases paquete com.soamedico.dato.....	89
27	Casos de Uso para Recepción de Resultados SOA-Médico.....	90
28	Integración Transferencia de Archivo usando patrones de integración empresarial.....	92
29	Esquema del contenido para xmlSchema_Astm.xsd.....	94
30	Esquema del contenido para xmlSchema_Xml.xsd.....	94
31	Código obtenido Iteración 3.....	95
32	Configuración Mule Integración Transferencia de Archivo (FileReader).....	96
33	Configuración Mule Integración (RecibirResultadoEquipo)	96
34	Integración Sistema Mensajería usando patrones de integración empresarial.....	97
35	Configuración Mule Integración (EnrutarResultadoJmsAFile).....	98
36	Esquema de Funcionamiento SOA-Médico Recepción de Resultados.....	98
37	Casos de Uso Envío de Resultados SOA-Médico.....	99
38	Integración Servicio Web usando patrones de integración empresarial.....	100
39	Código obtenido Iteración 3.....	101
40	Configuración Mule Integración (Enviar Resultado Servicio).....	102

**UNIVERSIDAD CENTROCCIDENTAL “LISANDRO ALVARADO”
DECANATO DE CIENCIAS Y TECNOLOGIA
POSTGRADO EN CIENCIAS DE LA COMPUTACION**

**ARQUITECTURA ORIENTADA A SERVICIOS PARA LA
INTEROPERABILIDAD CON EQUIPOS MEDICOS.
CASO DE ESTUDIO LABORATORIO CLINICO MASCIA**

Autor (a): Carmary Méndez Torres

Tutor (a): Jesús Guedez Maldonado

RESUMEN

Los sistemas de información dentro de los laboratorios clínicos, representa un punto crítico por la cantidad de información de carácter sensible que gestionan. Este fenómeno ha generado la necesidad de que estas organizaciones se encuentren trabajando bajo un modelo colaborativo, donde las mejoras en los procesos de administración y gestión de los resultados clínicos, permitan acceder a la información real de manera rápida y sin errores humanos, que pudieran comprometer la calidad de los servicios médicos. Para cubrir esta necesidad, los laboratorios clínicos deben impulsar sus capacidades de innovación mediante el uso de las tecnologías de información, al proporcionarles mecanismos de acceso y consulta a la información, reducción de costos y tiempos, crecer la participación en el mercado y mejorar la satisfacción de los clientes. La presente investigación tiene como propósito desarrollar una arquitectura orientada a servicios (SOA) basada en un bus de servicio empresarial (ESB), que como parte de la arquitectura tecnológica empresarial del laboratorio clínico Mascia, permita la interoperabilidad de su sistema de información con los resultados de los exámenes clínicos obtenidos desde los equipos médicos. La metodología utilizada se basó en un proyecto factible, descriptivo con revisión bibliográfica. Para lograr el objetivo, se identifican características funcionales y de desempeño necesarias para el intercambio de información entre equipos médicos y el sistema de información de laboratorio, estos requerimientos y restricciones guiados bajo la metodología Programación Extrema (XP), permitieron el diseño y desarrollo del modelo arquitectural para la construcción de software de este tipo. Las conclusiones del estudio recomienda la aplicación de la arquitectura propuesta al propio laboratorio clínico Mascia como un canal de innovación para la solución a la problemática planteada, pues al alinear el negocio con la tecnología, se obtienen capacidades que beneficiaran los procesos organizacionales.

Descriptor: Arquitectura Orientada a Servicios, Sistema de Información de Laboratorios, Bus de Servicio Empresarial, Metodología Programación Extrema.

INTRODUCCIÓN

La Organización “Socio Política”, está concebida en la sociedad del siglo XXI como varios sistemas que a su vez cada uno de ellos está integrado por varios otros sub-sistemas y todos ellos conforman organizaciones complejas que interaccionan entre sí. Dentro de esta gran organización, todos los sistemas y subsistemas encontrados tienen en común, que la unidad menor y objeto mayor es el hombre.

Puede definirse entonces que el Sistema de Salud es un sistema primordial que formando parte de la organización Socio Política, debe velar por la salud y calidad de vida del hombre. Del conjunto de entes que forman parte de este sistema, se hace referencia a los laboratorios clínicos, quienes se encargan de realizar los análisis clínicos de los exámenes médicos. Al respecto, el Colegio de Patólogos Americanos, estima que más del 80% de las decisiones médicas, tanto para el diagnóstico como para establecer terapias o pronósticos, se realizan en base a los resultados de dichos análisis, incluso en algunos casos, no hay sustituto para las pruebas de laboratorio, por ejemplo, un colesterol elevado sólo puede detectarse por análisis de laboratorio y mucho antes de que aparezcan los síntomas.

Como los médicos confían en los resultados que el laboratorio clínico produce para ayudarles a diagnosticar muchas situaciones y para seguir la eficacia del tratamiento, toda la informática sanitaria, contiene información sensible, para la cual deben establecerse mecanismos que garanticen la confiabilidad para todo tipo de datos producido, y disponer a múltiples profesionales información para los fines que fueron solicitados, asegurando una correcta toma de decisiones.

En la era actual y dentro de un contexto económico y empresarial, que se caracteriza por fenómenos como la globalización, los crecientes grados de competitividad, la evolución de las nuevas tecnologías y la naturaleza dinámica de los nuevos mercados, no parece existir ninguna duda acerca de que la innovación en los

procesos, representa uno de valores más críticos para lograr el éxito sostenible dentro de estas organizaciones.

Para conseguir un mayor nivel de agilidad en los procesos organizacionales, es necesario poder combinar rápidamente los distintos componentes de las tecnologías de información, capital humano, sistemas de información y otros, que permitan aumentar sus capacidades para la innovación. Por lo que, en los últimos años la automatización computacional ha tenido un crecimiento significativo en la industria de Laboratorios Clínicos, esta automatización consiste en sistemas informáticos para el manejo de los procesos del laboratorio: recepción de pacientes, toma e identificación de muestras, programación automática de analizadores, obtención y publicación de resultados, entre otros.

En este sentido, la integración de la información de los sistemas de información de los laboratorios clínicos con la información generada por las áreas diagnósticas y clínicas representadas por los equipos médicos, con el uso exhaustivo de los sistemas de codificación, representan un valor añadido no sólo como base de investigaciones epidemiológicas, sino para el establecimiento real de elementos básicos para la práctica de la medicina. De esta manera, una arquitectura orientada a servicios separa los procesos de negocio de las funciones automatizadas y organiza estas últimas en módulos individuales catalogados en un diccionario de servicios que permiten su utilización por parte de toda la organización.

Por tal razón, la presente investigación tiene como propósito fundamental desarrollar una arquitectura orientada a servicios que permita la interoperabilidad de equipos médicos y el sistema de información de laboratorio como parte de la arquitectura tecnológica empresarial del laboratorio clínico Mascia, sede Central, Barquisimeto, Estado Lara.

De igual forma, la investigación está estructurada de la siguiente manera:

Capítulo I, contenido del Planteamiento del Problema, los Objetivos de la Investigación, la Justificación y Alcances y Limitaciones.

Capítulo II, se describen los Antecedentes de la Investigación, Bases Teóricas, Bases Legales, Sistema y Operacionalización de las Variables.

Capítulo III, se definen la Naturaleza de la Investigación, Tipo de Investigación, Estudio de Factibilidad, Procedimiento de la Investigación y Técnicas para la Recolección de Datos.

Capítulo IV, se despliega la Propuesta para Desarrollar una Arquitectura Orientada a Servicios que permita la interoperabilidad de equipos médicos y el Sistema de Información de Laboratorio como parte de la arquitectura tecnológica empresarial del laboratorio clínico Mascia, sede Central, Barquisimeto, Estado Lara

Capítulo V, se describen las Conclusiones y Recomendaciones. Finalmente, se esbozan las Referencias Bibliográficas.

CAPÍTULO I

EL PROBLEMA

Planteamiento del Problema

En un medio globalizado y competitivo como el actual, es necesario que el recurso humano de toda organización, cuente con competencias personales de calidad que les permitan mantenerse a la vanguardia en sus áreas de especialización, pues de otra manera los conocimientos adquiridos pronto se volverán obsoletos como consecuencia de los cambios predominantemente tecnológicos.

En este sentido, las organizaciones para aprovechar al máximo de su capital humano, deben establecer estrategias que permitan obtener ventajas competitivas en el desarrollo de sus operaciones y en sus resultados, mediante el fortalecimiento de sus capacidades para la innovación al identificar aquellos factores que logren integrar funciones productivas con la creación de conocimientos e información. Las entidades capaces de asumir los riesgos que la innovación supone, deben entender que ésta es vista como un fenómeno social, asociada a una serie de aspectos estructurales, nacionales y tecnológicos que le son propias en el contexto de la sociedad del siglo XXI. Al respecto, Barry y Barry (2003), afirman:

“La sociedad del siglo XXI no es ya un paradigma ideológico puede en realidad concebirse como varios sistemas que a su vez cada uno de ellos está integrado por varios otros sub-sistemas y todos ellos conforman organizaciones complejas que interaccionan entre sí y que son permanentemente modificadas y reorganizadas por estas interacciones que las someten a todas a un 'sistema organización' mayor que podría llamarse Organización 'Socio Política' que no es más que un todo de las partes”.

De lo anterior, es propio mencionar que dentro de esta gran Organización Socio Polítca, todos los sistemas y subsistemas encontrados: biológico, de salud, económico, político, cultural, ecológico, medio ambiente, entre otros, están permanentemente interactuado por fenómenos directos, indirectos y emergentes o sumergentes, y tienen en común, que la unidad menor y objeto mayor es el hombre.

Por tanto, se concibe que en el Sistema de Salud, se identifica en primer término al hombre como objeto primario y hegemizador del mismo, quien a su vez integra los distintos sub-sistemas de la Organización Socio Polítca, y en segundo término su calidad de vida, lo cual, lleva a definir al sistema sanitario como una organización para el monitoreo permanente de la salud y su conservación, tanto en la prevención como en el tratamiento.

En este sentido, fortalecer las capacidades para la innovación en el Sistema de Salud, requiere del uso de las tecnologías de información y comunicación (TIC's) en la cotidaneidad de las relaciones sociales, culturales y económicas, para la creación, distribución y manipulación de la información en el seno del hombre y su salud, generando motores para la competitividad, el desarrollo y el progreso desde el contexto local hacia un contexto globalizado.

Al respecto, Campoli (2003), hace referencia que "la sociedad de la información constituye un marco para solucionar problemas de forma innovadora empleando tecnologías para atender las necesidades básicas aún no resueltas de forma tradicional". Es por esto, que las organizaciones sanitarias, como sector clave para la protección de la salud y calidad de vida, han asimilado el cambio que acompañan las nuevas tecnologías, viéndose optimistas en su integración a la sociedad de la información, generando un desafío a los sistemas de información como vía hacia la excelencia organizativa.

Observando el sistema sanitario, Barry y Barry (2003), fundadores de una compañía global Latinoamericana, donde su principal actividad es la de brindar soluciones tecnológicas en el ámbito de la salud, han observado que debido a la gran cantidad de actores del sistema sanitario, ha existido una evolución tecnológica desordenada, lo cual en consecuencia ha producido los siguientes inconvenientes:

“Múltiples repositorios de datos: Pacientes, beneficiarios, centros de atención, etc.; Circuitos y procedimientos administrativos orientados y alineados con los sistemas informáticos, las organizaciones alineadas al servicios de los sistemas de información y no al revés; Inexistencia de centros codificadores (vocabularios, glosarios e identificadores); Integración de aplicaciones punto a punto, propietarias y específicas para dicha integración; Falta de patrones de diseño de sistemas informáticos sanitarios, modelos de datos y atributos comunes: Lenguaje común; Falta de estándares y de políticas de administración e intercambio de información sanitaria”.

De todos los actores participantes en el sistema sanitario, esta investigación se centrará en los centros de diagnósticos o laboratorios clínicos, lugar donde profesionales del área (Tecnólogo Médico, Técnicos Superiores de Laboratorio Clínico, Bioquímicos, Químicos Fármaco Biólogos y Médicos) realizan análisis clínicos que contribuyen al estudio, prevención, diagnóstico y tratamiento de los problemas de salud de los pacientes. En este escenario, Cedeño (2008), determina tres grandes áreas en los laboratorios clínicos: “área administrativa, área de toma de muestras y recepción, y área de análisis, procesamiento de muestras, donde se realiza las funciones propias para llevar a cabo el correcto funcionamiento de la actividad específica por desarrollar”.

La misión de un laboratorio clínico, según Aguirre y otros (2002), es la de "servir de apoyo a la clínica, proporcionándole información fiable y útil para el correcto diagnóstico de las enfermedades, para la prevención y el seguimiento evolutivo de las mismas y para el control en la eficacia de la terapia aplicada". Para cumplir dicha misión, los laboratorios clínicos deben mover sus vértices en función

del tiempo, el coste y la calidad, bajo dos grandes características, la fiabilidad de los resultados de las pruebas diagnosticas solicitadas y la puntualidad en la entrega de dichos informes, al respecto, Barreiro (2002) expresa:

“la demora en la entrega de los resultados del Laboratorio produce una calidad percibida baja en los Clínicos y los pacientes, pero sobre todo, produce grandes costes de ineficiencia que repercuten en el retraso de altas de hospitalización, inicio de otras pruebas diagnósticas o terapéuticas, número de pacientes a tratar en el hospital de día, o número de visitas del paciente en consultas externas para proceder a su analítica y todos los cuales se pueden evaluar económicamente”.

Bajo esta perspectiva, Cedeño (2008) indica que "el desarrollo de los sistemas de información en el ámbito de los laboratorios clínicos ha supuesto un gran impacto en todos los aspectos, comparable al que en su día hubo la introducción de los analizadores automáticos". Por ello, los Sistemas de Información de Laboratorio (SIL), vistos como el conjunto de hardware y software que da soporte a la actividad de un laboratorio clínico, cuentan con un papel importante para la gestión y administración de estas organizaciones, el cual normalmente se refieren al manejo de la información derivada de todo el proceso analítico, desde que el médico formula una solicitud de análisis hasta que el laboratorio emite un informe.

Dentro del proceso clínico asistencial, los laboratorios clínicos han cambiado notablemente, al respecto, Aguirre y otros (2002), confirman que “esto ha sucedido como consecuencia del desarrollo tecnológico, la automatización de los procedimientos analíticos y pre analíticos mediante equipos médicos, el perfeccionamiento de los sistemas de información del laboratorio y la introducción de las nuevas tecnologías de la comunicación (telemedicina)".

De igual forma, Zubillaga y Acevedo (2001), afirman que “la Informatización del Laboratorio es un aspecto imprescindible en la interconexión de los equipos de laboratorio con la red informática”. Asimismo, Cedeño (2008), menciona que:

“Los laboratorios clínicos generan diariamente una cantidad de información inmensa, por lo que deben de dotarse de un sistema informático que permita un trámite expedito del trabajo, conectando los diferentes equipos automatizados al Sistemas de Información de Laboratorio (SIL), con lo que se logra el envío de los resultados de cada muestra”.

De lo antes expuesto y considerando, que un laboratorio clínico puede influir de forma determinante en los procesos asistenciales médicos, no sólo por la exactitud de los resultados clínicos, sino también, por el tiempo que transcurre desde el momento cuando se realiza la petición, hasta que el médico que ha solicitado la prueba recibe el resultado, concepto que se conoce como tiempo de respuesta. Se ve la necesidad de que los SIL, siendo un tipo de software que recibe, procesa y almacena información generada por procesos de laboratorios clínicos, establezca comunicación automatizada con el conjunto de instrumentos y equipamiento médico necesarios para el correcto desarrollo de la actividad descrita en su propia cartera de servicio.

En este sentido, Zubillaga y Acevedo (2001), señalan “son pocos los laboratorios que, aún teniendo un sistema, trabajan con los equipos clínicos conectados con las computadoras, ya sea por desconocimiento o por el costo adicional de esta actividad”. En la actualidad, el desarrollo tecnológico de los equipos médicos permite establecer comunicación con un ordenador a través del puerto serie y muy recientemente puertos de red, Siemens Medical Solutions Diagnostics (2006), menciona que al igual que "las computadoras son interconectados en una red, los instrumentos pueden ser conectados mediante una interfaz con un SIL. Para lograr esto, una interfaz debe ser creada para que el computador del instrumento pueda 'hablar' con el computador del SIL".

Sin embargo, la adopción de una interconexión automatizada entre el equipamiento clínico y los Sistemas de Información de Laboratorio (SIL) obliga a un desembolso importante para la organización, no siendo factible adquirir el software

de interfaz del fabricante o inclusive en algunos de estos equipos no hay referencia de la existencia de los mismos. Al respecto, Siemens Medical Solutions Diagnostics (2006), menciona que una interfaz de interconexión, "es un pequeño programa que traslada lo que el instrumento 'dice' en un lenguaje que el SIL entienda. La interfaz es generalmente creada por el proveedor del SIL y tiene un costo para el laboratorio (\$5.000 o más por conexión)".

Por otra parte, debido a diferencias tecnológicas entre los diferentes fabricantes (Roche, Hitachi, Stago, DPC, BeckMan Coulter, Siemens, entre otros), no se cuenta con un sistema estándar para la adquisición, procesamiento, almacenamiento y análisis de los datos contenido en dichos equipos. En consecuencia, Zubillaga y Acevedo (2001), mencionan, que "lo normal es que cada equipamiento tenga su propio protocolo y por tanto, una comunicación realizada para un equipo, generalmente, no puede ser aprovechada en otro".

Es importante resaltar, que desde hace varias décadas la necesidad de interoperabilidad y estandarización de las comunicaciones entre los datos, aplicaciones y componentes de Tecnología de Información (TI) de las empresas, ha originado que el área de la integración este cobrando una gran importancia. Al respecto, Frantz (2008), menciona que:

“Un problema frecuente en estos ecosistemas es integrar dos o más aplicaciones de forma que los datos que manejan por separado estén sincronizados o que puedan colaborar para ofrecer nueva funcionalidad o nuevas vistas de datos. Según un reciente informe de IBM los gastos de integración superan en una proporción de entre cinco y veinte a los de desarrollo de nueva funcionalidad”.

Lo antes expuesto, muestra que la integración de aplicaciones empresariales supone un reto importante, es un objetivo clave para las empresas operar en un mercado global altamente competitivo mediante la agilidad en los negocios, hacerlo posible, requiere que los procesos de TI presten mayor atención a los métodos para

acoplar de manera flexible diversas aplicaciones y suministrar así funciones empresariales que puedan convertirse en servicios reutilizables en toda la empresa.

En este contexto, TIBCO Software Inc. (2008), proveedor de servicios para acelerar procesos de negocio TI, menciona que "una arquitectura orientada a los servicios (SOA) es un método para construir una infraestructura de TI a partir de componentes acoplados de manera flexible, denominados 'servicios', que desempeñan funciones específicas", donde las aplicaciones empresariales, representan los componentes que serán acoplados mediante la creación y orquestación de múltiples servicios, eventos y modelos, de manera que el colectivo representa una función empresarial de alto nivel. Dicha arquitectura, permite reutilizar componentes, mantener la escalabilidad y calidad, reducir tiempos y costes para el desarrollo de nuevas funciones e incrementar la agilidad empresarial de los departamentos de TI.

Por otro lado, en la actualidad es un hecho aceptado la necesidad que sistemas de información clínica contengan semántica interoperable para la consecución de la continuidad asistencial, siendo una forma de conseguir dicha meta la normalización de los significados relacionados con la transferencia de información en el sector salud, como historias clínicas, análisis clínicos, entre otros. Cedeño (2008), sostiene que distintos organismos de normalización, conscientes de esta necesidad, se embarcaron en la creación de normas, "introduciendo en su elaboración los nuevos paradigmas aparecidos en este campo: separación de responsabilidades, separación de puntos de vista y separación de información y conocimiento".

Al respecto, la ASTM (American Society for Testing and Materials - Sociedad Americana para Pruebas y Materiales) organismo mundial que desarrolla normas por consenso aplicables a los materiales, productos, sistemas y servicios, define dos estándares para la comunicación entre los instrumentos de laboratorio y los sistemas de información el ASTM E-1381 protocolo de bajo nivel y el ASTM E-1394 protocolo de alto nivel.

De igual forma, la HL7 (Health Level Seven - Nivel Siete en Salud), que es una organización de connotación internacional que define estándares dirigidos a permitir la interoperabilidad entre aplicaciones heterogéneas en el ámbito de la salud, ha desarrollado especificaciones que utilizando un metalenguaje extensible de marcado con etiquetas XML (eXtensible Markup Language) permite el intercambio electrónico de datos de salud basada en el RIM (Reference Information Model - Modelo de Referencia de Información), el estándar de mensajería HL7 en la actualidad se encuentra en la versión 3.

Los resultados de estos procesos de estandarización, generan que aplicaciones empresariales en el sector salud totalmente heterogéneas, puedan ser semánticamente interoperables entre ellos, si son conformes a una misma norma, pero que no pueden comunicarse cuando están basados en normas distintas. Además, ambas normas están inmersas en el proceso necesario para convertirse en normas internacionales ISO (International Organization for Standardization - Organización Internacional para Estandarización), por lo que se prevé que los sistemas conformes a las mismas van a tener que convivir durante un tiempo.

Actualmente en el laboratorio clínico Mascia, el proceso de obtención de los análisis clínico arrojados de los equipos hacia el Sistema de Información de Laboratorio (SIL), se realiza mediante una boleta impresa con los exámenes a realizar por paciente donde el bioanalista o auxiliar llena manualmente con los resultados obtenidos desde los equipos médicos, los cuales se transcriben al SIL por un personal destinado para esto, seguidamente se imprimen los resultados y de ser necesario son comparados nuevamente con la boleta por el bioanalista al momento de firmar los resultados. Por ello, este método manual está sujeto a errores humanos y demora en la entrega de resultados, que pudieran comprometer la calidad de los servicios médicos y el prestigio de la institución, a la vez que representa un maltrato al paciente si se le entregan los resultados fuera del horario establecido o con errores en los mismos.

Por otro lado, existen diversos fabricantes de equipos médicos que ofrecen una interconexión punto a punto para automatizar los procesos de transcripción, el cual siendo software propietario no ofrecen una interfaz para comunicarse con el SIL de la organización, como el laboratorio clínico Mascia cuenta con muchos equipos médicos de alta tecnología, irse por esta vía de automatización requeriría de muchos equipos de computación e interfaces de comunicación, tantos como equipos médicos y unidades de transcripción existen en el laboratorio, ocasionando grandes gastos de hardware y software.

En atención a lo anteriormente expuesto, la investigación propuesta se centra en el desarrollo de una Arquitectura Orientada a Servicios que permita la interoperabilidad de equipos médicos y el Sistema de Información de Laboratorio como parte de la arquitectura tecnológica empresarial del laboratorio clínico Mascia, sede Central, Barquisimeto, Estado Lara. El diseño de dicho modelo arquitectural, identifica los componentes y sus relaciones, que permitirá el desarrollo de una solución de integración donde las funcionalidades son expuestas como servicios que siendo enrutados por un bus troncal de comunicación, logran la interoperabilidad de los equipos médicos de un laboratorio clínico con la gestión de su SIL de mejor forma, más rápidamente y en lenguaje universal. Para este diseño, se hará el uso de las utilidades ofrecidas por la Ingeniería de Software, como: metodología de desarrollo, patrones arquitecturales, patrones de diseño, entre otros, las cuales ofrecen soluciones prácticas reconocidas para problemas de software.

Sobre la base de la problemática planteada se formulan las siguientes interrogantes de investigación:

¿Qué requerimientos funcionales y no funcionales permiten una integración de equipos médicos y el Sistema de Información de Laboratorio (SIL)?

¿Cuáles son los componentes y mecanismos de integración incluidos en el diseño del modelo arquitectural que permita la interoperabilidad de equipos médicos y el Sistema de Información de Laboratorio (SIL)?

¿El desarrollo del modelo arquitectural integrará los equipos médicos y el Sistema de Información de Laboratorio (SIL) del laboratorio clínico Mascia?

Para responder las interrogantes anteriores, se ve la necesidad de crear un modelo arquitectural que basado en un enfoque orientado en servicios bajo un bus troncal de comunicación empresarial, logre la interoperabilidad de los equipos médicos con los sistemas de información de laboratorio.

Dicho modelo arquitectural, estará en la capacidad de ofrecer diferentes canales, protocolos y estándares de comunicación, que permitirá obtener la información desde los equipos médicos, para su posterior transformación y almacenamiento en un repositorio intermedio, desde el cual, el SIL podrá obtener los resultados de las pruebas diagnosticas a manera de servicios y en un formato estándar.

Objetivos de la Investigación

Objetivo General

Desarrollar una Arquitectura Orientada a Servicios que permita la interoperabilidad de equipos médicos y el Sistema de Información de Laboratorio como parte de la arquitectura tecnológica empresarial del Laboratorio Clínico Mascia, sede Central, Barquisimeto, Estado Lara.

Objetivos Específicos

1. Identificar los requerimientos funcionales y no funcionales que permiten una integración de equipos médicos y el Sistema de Información de Laboratorio (SIL).
2. Diseñar el modelo arquitectural que permitirá la selección de los componentes y los mecanismos de integración para la interoperabilidad de equipos médicos y el Sistema de Información de Laboratorio (SIL).
3. Desarrollar el modelo arquitectural en un entorno de integración de equipos médicos y el Sistema de Información de Laboratorio (SIL) del laboratorio clínico Mascia.

Justificación e Importancia

Siendo el hombre el eje principal dentro del sistema sanitario, los laboratorios clínicos con los Sistemas de Información de Laboratorio (SIL), representa un punto crítico por la cantidad de información de carácter sensible que gestionan. Por ello, la automatización computacional ha tenido un crecimiento significativo en la industria de Laboratorios Clínicos, pues esta automatización consiste en sistemas informáticos

para el manejo de los procesos del laboratorio: recepción de pacientes, toma e identificación de muestras, programación automática de analizadores, obtención y publicación de resultados, entre otros.

Por ello, la presente investigación tiene como propósito fundamental Desarrollar una Arquitectura Orientada a Servicios que permita la interoperabilidad de equipos médicos y el Sistema de Información de Laboratorio como parte de la arquitectura tecnológica empresarial del laboratorio clínico Mascia, sede Central, Barquisimeto, Estado Lara. Asimismo, se ofrecerá una visión de la estructura y de los aspectos de funcionamiento de los elementos determinantes, concurrentes e influyentes en la arquitectura orientada a servicios, para dar respuesta a la problemática que actualmente se observa, mediante el reenfoque de los procesos que se manejan en dicha función, y con ello resolver y mejorar con efectividad los cambios que se susciten en el entorno organizacional, donde se desarrollará dicha arquitectura.

Además, el desarrollo de una arquitectura que permitirá la interoperabilidad de equipos médicos con los SIL, tiene un impacto social, ofreciendo un mecanismo confiable para obtención automática de análisis de resultados, mejorando en manera sustancial los tiempos de repuestas, atención a un mayor número de pacientes y así otorgar a múltiples profesionales del sector salud información oportuna y coherente para la toma de decisiones.

Por otro lado, los laboratorios clínicos, al prescindir de personal humano para la transcripción manual de resultados, logran optimizar la gestión de los recursos humanos y de esta manera enfocar los esfuerzos a la gestión de calidad concerniente al proceso de los análisis de resultados.

Finalmente, el estudio tiene su importancia metodológica, pues fortalecerá líneas de investigación en el ámbito de la Tecnología de Información y los Sistema de

Información, al utilizar modelos y métodos científicos que permitirán comprender la naturaleza del objeto de estudio y la resolución de problemas con una nueva visión que conceda mayor capacidad de respuesta en las organizaciones tanto a nivel nacional como regional. Además, las metodologías que se aplicarán para los estudios del futuro, permitirán dar mayor libertad de análisis en cuanto a tendencias y construcción de posibles escenarios en los que se moverá la tecnología en la sociedad, conformará las bases para la integración de conocimientos y desarrollo de innovación.

Alcances y Limitaciones

La investigación se orientará al desarrollo de una arquitectura de integración de los equipos Immulite 2000 de Siemens y Coulter LH 500 de Bekman Coulter con el Sistema de Información de Laboratorio (SIL). Su diseño y desarrollo se basará en una Arquitectura Orientada a Servicios (SOA) sobre un Bus de Servicio Empresarial (ESB) ejecutándose en un servidor, permitirá una comunicación unidireccional desde los equipos para obtener los informes de resultados y serán expuestos a los Sistemas de Laboratorio Clínico (SIL) por un protocolo para el intercambio de mensajes sobre redes basado en XML como el Objeto para Protocolo de Acceso Simple (SOAP).

En este sentido, para el Bus de Servicio Empresarial (ESB) se utilizará la herramienta Mule en su versión de licencia libre, lo cual permitirá medir el desempeño y comportamiento de la misma para soluciones de integración, y determinar si logra de una manera fácil y sencilla, poner a disposición un pool de servicios que basados en los protocolos de comunicación de los equipos médicos (ASTM E-1394 y HL7 v3) previamente encapsulados, puedan obtener los informes de resultados y ser enrutados de manera transparente a quien lo solicite mediante un mensaje estandarizado, y en un futuro esta investigación pueda convertirse una arquitectura de integración tan general que logre la conexión de un sin fin de equipos de hardware de distinta marcas y modelos.

CAPÍTULO II

MARCO TEÓRICO

El Marco Referencial está basado en los antecedentes y las bases teóricas. En los antecedentes, se describen todos los estudios y proyectos previos que inspiran el desarrollo de la presente investigación. Las bases teóricas están constituidas por las teorías, definiciones y enfoques que se consideran pertinentes en el contexto objeto a estudio.

Antecedentes de la Investigación

En relación con el propósito perseguido en esta investigación, como es la interoperabilidad de equipos médicos con sistemas de información de laboratorios, se analizan los siguientes trabajos:

En la investigación realizada por Zubillaga y Acevedo (2001), exponen que existe desconocimiento o un costo adicional alto para que los equipos de laboratorios clínicos se interconecten con las computadoras a fin de que los resultados sean ingresados automáticamente, esto debido, a que la forma y el protocolo utilizado para la comunicación es escogida por el fabricante del equipo, siendo lo normal que cada equipamiento tenga su propio protocolo y por tanto, una comunicación realizada para un equipo, generalmente no puede ser aprovechada en otro.

En dicha investigación, luego de identificar los métodos de comunicación predominantes para extraer información: puertos de serie, utilización de disquetes y el menos utilizado puerto de red, y los protocolos de comunicación más utilizado en el área de Análisis de Laboratorios Clínicos: Protocolo Kermit, y al Protocolo ASTM

descrito en las normas E 1381 y E 1394, generan una interfaz de interconexión del equipo ACCESS de la Firma Sanofi Pasteur con el Sistema Informático, basada en Delphi 2 sobre una plataforma Windows 95 y opciones de comunicación mediante el uso del programa ASYNC versión 2 de la firma Turbo Power, que permite al usuario la obtención de los resultados de los exámenes por pacientes, para que sean luego enviados a un servidor SQL Windows NT.

Con esta novedosa interfaz dentro de la informatización, Zubillaga y otros (2001), como conclusión, determinan que la interconexión computadora-equipo médico, representó una reducción promedio de 8 a 2 días en la recepción de resultados, papel preponderante que garantizó una prestación adecuada a las necesidades del usuario.

En el mismo orden de ideas, Ramos y Simón (2007), quienes plantean que el problema se presenta en la transcripción manual de los resultados, lo cual acarrea errores humanos y demora el proceso, que pudieran comprometer la calidad de los servicios médicos, intentaron solucionar estos problemas, con un diseño de un sistema de interface que permite adquirir, procesar, analizar, almacenar y representar los datos entregados por los equipos de laboratorio clínico HITACHI® 902, PENTRA® 120 RETIC y START ®4. Este sistema se basa en la comunicación de estos equipos con un ordenador a través del puerto serie utilizando el protocolo RS232C y conectado físicamente con un cable diseñado para esta aplicación. El control, adquisición y procesamiento de la información se realiza mediante un programa diseñado en MATLAB.

Esta investigación tiene relación con el estudio efectuado, pues concluye que la implementación del sistema al permitir almacenar los datos automáticamente en un fichero en EXCEL, minimiza la probabilidad de errores en los resultados de los pacientes y garantiza la entrega de los mismos en el horario establecido y con la calidad de las mismas.

De igual forma, en el estudio realizado por Ramos y Simón (2007), indican que a pesar que los modernos equipos de los laboratorios clínicos, soportan generalmente el protocolo de comunicación asíncrono por puerto serie RS232C que permite automatizar el almacenamiento de los resultados en un servidor de base de datos, se requiere de tarjetas comerciales de múltiples puertos series instalada en la placa madre del servidor si se desean conectar varios equipos, lo cual aumenta la probabilidad de conflictos de interrupciones internas, limitan la flexibilidad de la interconexión, y a su vez algunos equipos de laboratorio permiten sólo una distancia de tres (3) metros hasta el puerto, en este caso el servidor tendría que estar en un departamento específico afectando la funcionalidad del departamento.

Una forma de simplificar este inconveniente, los autores antes mencionados, proponen conectar los equipos a través de un concentrador que implementa una tarjeta multipuerto serie basada en un microcontrolador PIC (Programmable Intelligent Computer), el diseño reduciría la complejidad y costos comparado con los diseños comerciales, y la utilización de un dispositivo programable, dedicado para esta aplicación, permite controlar y almacenar el flujo de información de forma bidireccional. Así, el volumen de muestras procesadas por los equipos de diferentes fabricantes, se despliegan sobre una plataforma única que garantiza la calidad y fiabilidad de los resultados en el menor tiempo posible.

En este contexto, las investigaciones anteriormente descritas, evidencia la factibilidad de la interconexión de los equipos de laboratorios con computadoras, pero existe otro reto, el de permitir la interoperabilidad de los datos obtenidos desde los equipos con los Sistemas de Información de Laboratorio (SIL), entre otros. Gómez y otros (2003), en el marco de la integración de los sistemas de información de la Historia Clínica Electrónica, el Laboratorio Central y los Servicios al Paciente, el Departamento de Información Hospitalaria del Hospital Italiano de Buenos Aires, desarrollaron un sistema de consultas e impresión de resultados de exámenes

complementarios en línea por Intranet, utilizando HL7 para lograr la interoperabilidad de los sistemas. La solución, consiste en una interfaz que se encarga de generar y enviar al Laboratorio central el mensaje HL7 de consulta, este recibe la solicitud, accede a su base de datos, arma un documento HTML con el informe del estudio el cual es enviado para su visualización e impresión.

El modelo tecnológico para el desarrollo de dicha aplicación WEB está basado en tecnología Java Enterprise Bean (J2EE) y clientes web con tecnología JSP y Servlets. La implementación se realizó sobre un entorno libre, que utiliza Web Containers y servidores de aplicación del proyecto Apache Jakarta. El modelo de información se montó en una base de datos SQL. En cuanto al intercambio de mensajes se utilizó el estándar HL7 Versión 2.3 montado sobre un servidor de mensajería IBM MQSeries como manejador de mensajes: esto permitió simplificar la capa física de implementación de HL7 respetando el marco conceptual de Evento / Mensaje / Respuesta de HL7 sin la necesidad de una conexión sincrónica obligatoria, pero gozando de la posibilidad de obtenerla si fuera necesaria.

Para Gómez y otros (2003), el uso de HL7 en combinación del uso de la tecnología WEB como estándar de comunicaciones, le permitió mantener la independencia de cada uno de los sistemas sin necesidad de estar ligado a una plataforma de hardware o de software y de este modo potenciar la comunicación institucional.

Al respecto, en el ámbito mundial existen gran cantidad de proveedores de software que proponen diversas arquitecturas de integración de aplicaciones, TIBCO Software Inc. (2008), menciona que ha sido un reto clave para los desarrolladores la integración de múltiples sistemas que emplean distintos lenguajes y formatos, para la popularización de los Servicios Web ha proporcionado a SOA un formato estandarizado de mensajería SOAP (Simple Object Access Protocol - Protocolo Simple de Acceso a Objetos) que hace posible la interacción de diversos sistemas, que en conjunto, al uso de los WSDL (Web Services Description Language -

Lenguaje de Descripción de Servicios Web) para describir el final de la interfaz del programa de aplicación (API), hacen posible que aplicaciones programadas en diferentes plataformas, se comuniquen usando interfaces comunes.

Sin embargo, los mecanismos de integración basados en los Servicios Web y la SOA, debido a sus principios de descomposición inherentes, se enfrentan a retos en mayor escala, que a los enfrentados por las generaciones anteriores de sistemas distribuidos, el desafío principal a resolver será la escalabilidad de las conexiones punto a punto, lo que se conoce también como el “problema de conexión M*N”.

En busca de esta solución, TIBCO Software Inc. (2008), inventó el paradigma del Information Bus™, que idea una arquitectura basada en un bus, mediante el cual, cada aplicación se conecta a esa única infraestructura troncal común: esto reduce al mínimo las conexiones y proporciona una ubicación centralizada para la administración y gestión de sistemas integrados. Esta infraestructura troncal fiable proporciona un bus de servicios a escala empresarial (ESB), que vinculada con SOA precisa una infraestructura troncal capaz de ir más allá de la mensajería distribuida tradicional para gestionar la complejidad de cómo un servicio cliente se conecta y se comunica con el proveedor del servicio, para así, proveer transformación compleja, enrutamiento y conectividad acoplada libremente en un entorno TI heterogéneo, independientemente de las plataformas usadas.

En este sentido, dicha compañía, implementa mediante su producto BusinessWorks™, un ESB maduro, dotado de toda la funcionalidad necesaria para diseñar una SOA. Este producto, según TIBCO Software Inc. (2008), "es una plataforma SOA extensible que permite la integración de aplicaciones empresariales y el desarrollo y el despliegue de servicios Web. Su arquitectura basada en un bus puede ser ampliada para albergar un gran rango de capacidades de integración". Este producto, es una herramienta organizacional, eficaz para resolver problemas de integración complejos, la cual, teniendo como principal característica el enfoque de

configurar en lugar de codificar, contribuye a la escalabilidad y la reducción de los costos futuros.

Seguidamente, Fuenmayor y Morales (2008), realizaron un estudio titulado “Arquitectura de Servicios de e-salud para Salud Chacao sobre Redes Inalámbricas Metropolitana”, cuyo objetivo fue desarrollar una arquitectura de servicios de e-salud a través de una Red Banda Ancha Inalámbrica Móvil para el Municipio Chacao en la ciudad de Caracas Venezuela, permitiendo a las instituciones encargadas de la salud del sector incorporar un conjunto de aplicaciones para mejorar su gestión administrativa y de servicio mejorando la calidad de vida de los habitantes del Municipio.

En dicha investigación, se proponen una arquitectura básica para los servicios de Telemedicina e E-Salud cuyos componentes principales están conformados por: puntos de captura/consulta de datos, procesamiento y almacenamiento de los mismos, permitiendo que las historias médicas y datos personales de los pacientes se encuentran almacenados en un servidor centralizado, sobre el cual se actualizan los datos obtenidos de la captura de los equipos medidos de diagnóstico de los pacientes, los cuales pueden transmitir la información a dicho servidor centralizado bien sea de forma cableada ó inalámbrica.

Por otra parte, consta de otro componente importante dentro de ésta arquitectura, que es la Gestión Pre-Hospital, el mismo permite la transmisión de datos de pacientes (es decir, signos vitales: Ritmo cardíaco, Presión arterial, imágenes en casos de accidentes, etc.) en casos de emergencia fuera del centro hospitalario, así como la aplicación de Consultas médicas fuera del mismo (por ejemplo, unidades de asistencia móvil).

Esta investigación tiene relación con el estudio efectuado, pues a través del uso de las tecnologías de información y comunicación, busca integrar equipos médicos a

un sistema de gestión hospitalaria remota, mejorando y agilizando el tratamiento médico, lo que puede elevar considerablemente las posibilidades de vida de un paciente que es trasladado hacia el centro hospitalario, si durante el traslado está recibiendo el tratamiento adecuado en tiempo real de parte de el especialista remotamente. De igual forma, se logrará un gran impacto social, puesto que buscan aumentar la calidad de prestación de los mismos, tratando de satisfacer las necesidades de todos los habitantes de un sector (municipio) y que podría expandirse hacia otros evitando la dispersión geográfica permitiendo el acceso oportuno a la atención médica o servicios a todos los ciudadanos.

Los esfuerzos realizados por diversos investigadores y organizaciones anteriormente descritos, sirven de inspiración para la presente investigación, la cual, plantea el desarrollo de una arquitectura orientada a servicios bajo un bus troncal de comunicación como parte de la arquitectura tecnológica empresarial de un laboratorio clínico, que permita la interoperabilidad de equipos médicos con sus sistemas de información.

Bases Teóricas

Tecnologías de Información

Las organizaciones para el fortalecimiento de sus capacidades de innovación, hacen uso intensivo de las tecnologías de información, a fin de que estas le ayuden a enfrentar la competitividad presente en el mundo globalizado y lograr tener más éxito en el mercado.

La Asociación de Tecnología de Información de América (ITAA), citada por Hernández (2008), afirma que la tecnología de información es “el estudio, diseño,

desarrollo, implementación, soporte o dirección de los sistemas de información computarizados, en particular de software de aplicación y hardware de computadoras”. Asimismo, Bologna y Walsh, (1997), señalan que la tecnología de la información se entiende como "aquellas herramientas y métodos empleados para recabar, retener, manipular o distribuir información, pues la tecnología de la información se encuentra generalmente asociada con las computadoras y las tecnologías afines aplicadas a la toma de decisiones”.

Desde hace varias décadas, la Tecnología de la Información (TI) está cambiando la forma tradicional de hacer las cosas, las personas que trabajan en gobierno, en empresas privadas, que dirigen personal o que trabajan como profesional en cualquier campo utilizan la TI cotidianamente mediante el uso de Internet, las tarjetas de crédito, el pago electrónico de la nómina, entre otras funciones; es por eso que la función de la TI en los procesos de la empresa como manufactura y ventas se han expandido grandemente. Por ello, utilizando eficientemente la tecnología de la información se pueden obtener ventajas competitivas, pero es preciso encontrar procedimientos acertados para mantener tales ventajas como una constante, así como disponer de cursos y recursos alternativos de acción para adaptarlas a las necesidades del momento, pues las ventajas no siempre son permanentes.

Así, las tecnologías de información, es el conjunto de herramientas y métodos que permiten gestionar la información en pro del desarrollo de las organizaciones mediante la mezcla de la computación y las comunicaciones. Al respecto, Bologna y Walsh (1997), describen las aplicaciones de la tecnología de información, las cuales son: el almacenamiento de datos, información para la toma de decisiones y control, sistema de toma de decisiones organizacionales, planeación de recursos empresariales y administración del conocimiento.

Características de las tecnologías de la información

Bologna y Walsh (1997), afirman que las características de la tecnología de información se fundamentan en la Inmaterialidad, donde la digitalización hace posible almacenar gran cantidad de información en dispositivos físicos de pequeño tamaño, la Instantaneidad, que permite la transmisión de la información entre lugares totalmente distantes de manera instantánea y las aplicaciones multimedia.

Una de las áreas de los profesionales de TI está relacionada con el desarrollo de sistemas de información, que en relación con las características de las tecnologías de información, permiten a las organizaciones mejorar los procesos, reducir tiempos, centrarse en tareas que agreguen valor y así obtener información fiable e inmediata que permitan la oportuna toma de decisión.

Sistemas de Información

Laudon y Laudon (1996), definen los sistemas de información (SI) como “un conjunto de componentes interrelacionados que reúne (u obtiene), procesa, almacena y distribuye información para apoyar la toma de decisiones y el control en una organización”, y sus actividades están determinadas por: entrada, procesamiento y salida de datos.

En este sentido, puede entenderse que los sistema de información, sirven de apoyo para las actividades de una organización, utilizan recursos de personas, hardware, software, datos y redes para realizar actividades de entrada, procesamiento, salida, almacenamiento y control que transforman los recursos de datos en salidas de información. A continuación, O'brien (2001), ilustra un modelo de sistema de información que expresa un marco conceptual fundamental para los principales componentes y actividades de los sistemas de información.

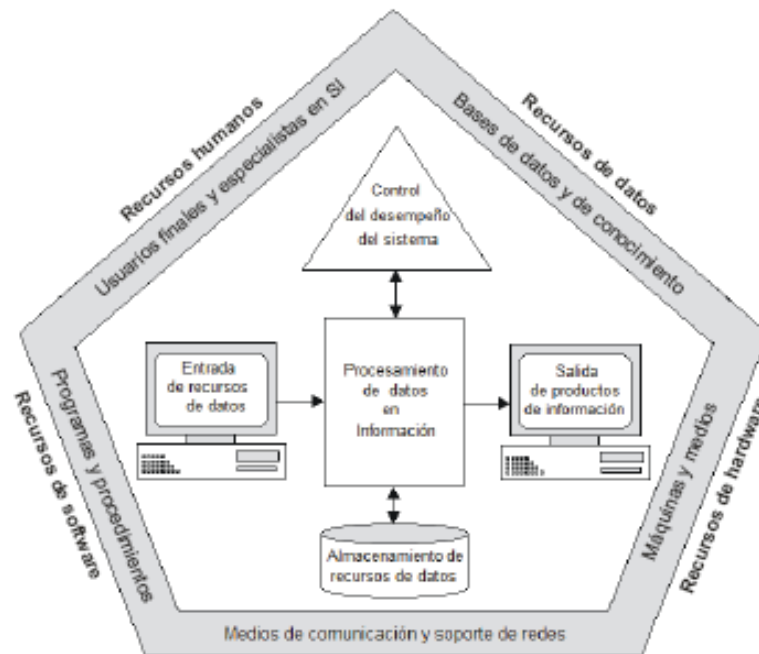


Figura 1. Recursos de Redes, Fuente: O'brien (2001).

Este modelo, pone en relieve las relaciones entre los componentes y las actividades de los sistemas de información y proporciona un marco que hace énfasis en las personas, el hardware, el software, los datos y las redes como recursos básicos. De donde se extrae, que los sistemas de información representan las siguientes funciones:

- Un área funcional principal dentro de la empresa, que es tan importante para el éxito empresarial como las funciones que en ella devienen.
- Una colaboración importante para la eficiencia operacional, la productividad y la moral del empleado, y el servicio y satisfacción del cliente.
- Una fuente importante de información y respaldo importante para la toma de decisiones efectivas por parte de los gerentes.
- Un ingrediente importante para el desarrollo de productos y servicios competitivos que den a las organizaciones una ventaja estratégica en el mercado global.

Sistemas de Información de Laboratorios

Marroquin (2009), plantea que los sistemas de información de laboratorios (SIL), son sistemas complejos altamente configurables, que adaptándose a distintos modelos de flujo, administran toda la información que se genera en un laboratorio clínico. Esta información contiene la demografía de los pacientes, los análisis clínicos que los médicos le solicitan y los resultados obtenidos para estos análisis.

Por otro lado, Rosales (2009), sostiene que para hacer posible un ejercicio asistencial diferente mediante modernas funciones, se deben identificar los cuatro pilares del laboratorio clínico-asistencial, que son el equipo analítico clínico, la gestión integral del laboratorio, el aseguramiento de la calidad y el sistema de información del laboratorio. Este último, considerado un pilar fundamental, ya que si la información es el producto final del laboratorio clínico - asistencial, el SIL debe ser el centro y el eje sobre el que se impulse todo laboratorio.

Siguiendo con este orden de ideas, el mismo Rosales (2009), indica que el SIL será el banco y fuente de datos para la información diagnóstica de las pruebas, por lo tanto, las prestaciones del SIL, el modo en que se use, el volumen y la explotación que se haga del banco de datos, van a determinar la cantidad y la calidad de los servicios y los resultados finales del laboratorio. Estos, requieren de tres herramientas informáticas independientes, pero interrelacionadas, que son:

- La base de datos que con el tiempo se constituye en el banco de datos y en el motor del sistema informático que da velocidad de procesamiento al sistema,
- El programa o aplicación informática específica de gestión e información del laboratorio clínico y finalmente,
- La aplicación general de consulta e impresión que permite medir y estandarizar la base y banco de datos, por lo que el SIL en definitiva

condiciona la labor asistencial y la integración del laboratorio en su entorno clínico y sanitario.

Arquitectura de los Sistemas de Información de Laboratorios

Para Marroquín (2009), regularmente los sistemas de información de laboratorio son aplicaciones de 3 capas, debido a que la información que manejan este tipo de sistemas es confidencial y de vital importancia para la salud de los pacientes, se hace necesario agregar una capa de seguridad por medio de la cual se respete dicha confidencialidad. Las capas que integran las aplicaciones son:

- **Capa de presentación o interfaz de usuario**, esta capa está formada por los formularios y los controles en los formularios. Sirve para la interacción con el usuario final.
- **Capa de lógica del negocio**, en esta capa se definen las clases que implementan los procesos del laboratorio: recepción de órdenes, procesamiento de muestras, impresión y entrega de resultados, entre otros.
- **Capa de acceso a datos**, finalmente está la capa de acceso a datos, son clases que permiten la conexión con la base de datos a través de consultas SQL ó procedimientos almacenados.
- **Capa de seguridad**, esta capa define métodos de validación de usuarios e inscripción de información para casos específicos del sistema que lo requieran.

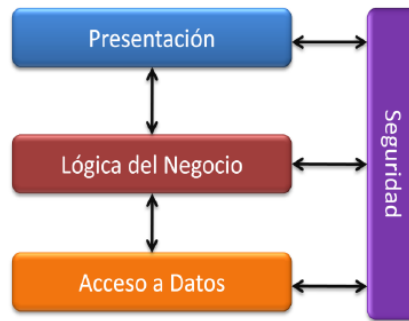


Figura 2. Arquitectura de un SIL. Fuente: Marroquin (2009).

Operación Básica de los Sistemas de Información de Laboratorios

Marroquin (2009), la operación básica de un sistema de información para laboratorios consiste en el ingreso de órdenes de trabajo, luego se recoge la muestra del paciente y se realizan los análisis en los diferentes equipos, los resultados son revisados y publicados en el sistema y finalmente se obtiene el informe final o reporte de laboratorio. Cada una de estas etapas de la operación tiene algunas actividades que a continuación se describen:

- ***Ingreso de órdenes***, las órdenes son ingresadas por un asistente, enfermera o un técnico de laboratorio, por lo que la orden de laboratorio contiene una lista de estudios a ser efectuados en una o más muestras de un paciente.
- ***Impresión de etiquetas con código de barras***, los sistemas de información de laboratorios deben tener la capacidad de emitir etiquetas de códigos de barras para identificar las muestras de cada orden y cada paciente, de esta forma se puede darle seguimiento a la muestra durante todo su proceso en el laboratorio hasta la obtención de resultados.
- ***Obtención de resultados***, la obtención de resultados se puede dar de dos formas, de forma manual y de forma automatizada mediante los analizadores.

- ***Captura de resultados con analizadores automatizados***, en los laboratorios automatizados la programación de los equipos es automática, el técnico de laboratorio únicamente coloca el tubo con la etiqueta con código de barras en el equipo y éste interroga al SIL para saber que estudios debe realizar. Luego de procesar los estudios, el analizador le envía los resultados al sistema de laboratorio. En este punto, ya pueden ser revisados y validados por un superior.
- ***Captura de resultados manual***, en estos laboratorios el técnico programa los equipos de forma manual. Establece que se le debe trabajar a cada muestra y luego pone a trabajar al equipo. Al obtener los resultados, el técnico los ingresa al SIL para que un superior los pueda revisar y validar. En este proceso hay cabida al error humano, tanto al programar el equipo, como al transcribir los resultados al sistema.
- ***Publicación de resultados***, luego que se obtienen los resultados de los analizadores, ya sea de forma manual o automatizada, estos son publicados en el sistema de laboratorio, de forma provisional, es decir, no son resultados finales. Por ello, es necesario que un supervisor, químico biólogo en la mayoría de los casos, revise los resultados y los valide. Después de validados ya se pueden entregar los resultados o ser enviados al sistema hospitalario, según sea el caso.

Captura de resultados con equipos médicos

Según Siemens Medical Solutions Diagnostics (2006), los sistemas de información en un laboratorio pueden alcanzar un alto grado de automatización, esta automatización se refiere en especial a los analizadores clínicos, los cuales tienen la capacidad de entablar conversaciones con un SIL, mediante el uso de una

computadora denominada Host o Interface. El flujo del uso de los analizadores para el caso del Immulite es el siguiente:

1. Las órdenes de trabajo por pacientes son creadas en el SIL.
2. El SIL asigna un número de orden al paciente y genera etiquetas de código de barra y la colección de exámenes a aplicar.
3. Las órdenes de trabajo son descargadas al instrumento.

3.1. Si el instrumento está configurado para comunicación unidireccional, el operador debe introducir la orden en el instrumento mediante el código de barra. Los resultados serán luego enviados al SIL. Figura 3.

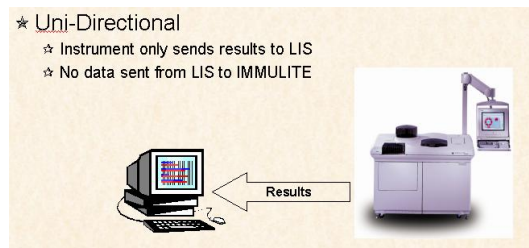


Figura 3. Flujo Unidireccional. Fuente: Siemens Medical Solutions Diagnostics (2006).

3.2. Si se configura comunicación bidireccional en lote, el operador debe requerir que el SIL envíe las órdenes de pacientes al instrumento, luego envía todos los resultados al SIL. Figura 4.

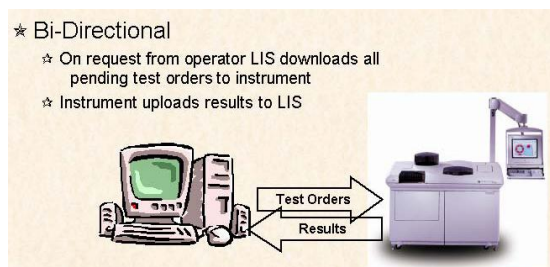


Figura 4. Flujo Bidireccional. Fuente: Siemens Medical Solutions Diagnostics (2006).

3.3. Si es configurada consulta bidireccional, el operador ubica los códigos de barra que identifican las muestras. Cuando el instrumento las escanea, envía un mensaje al SIL solicitando las ordenes, luego envía todos los resultados al SIL. Figura 5.

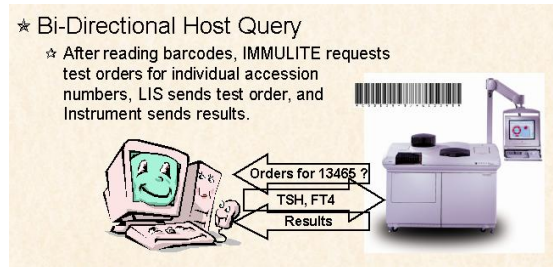


Figura 5. Flujo Consulta Bidireccional. Fuente: Siemens Medical Solutions Diagnostics (2006).

Protocolos estándar de comunicación con equipos médicos

Gaya (1999), expresa que la transferencia de datos entre diferentes máquinas, sean éstas sistemas analíticos u ordenadores, se realiza mediante protocolos estándar establecidos por consenso internacional con este fin. La estandarización ha hecho posible que las máquinas se entiendan entre sí, obviando la creación de interfaces específicas para cada sistema, como debía hacerse hace poco más de una década.

Al respecto, el mismo autor indica, que comúnmente, en los laboratorios clínicos se utiliza principalmente la norma ASTM E1394, que hace referencia al intercambio de datos clínicos y a la comunicación de instrumentos de laboratorio a ordenadores, y posteriormente, el protocolo HL7 (Health Level Seven), como estándar para intercambio de información entre las distintas instancias del sector salud.

Estándar ASTM

Siemens Medical Solutions Diagnostics (2007), indica que sus equipos Immulite e Immulite 2000/2500 operan bajo las especificaciones del estándar ASTM E1394.

ASTM international (2002), establece que esta norma ASTM E1394 se aplica a la transmisión digital en ambas direcciones de las solicitudes remotas y los resultados clínicos entre los instrumentos y sistemas informáticos, su objetivo es documentar las convenciones comunes que se requieren para el intercambio de los resultados clínicos y datos de los pacientes entre los instrumentos clínicos y sistemas informáticos.

Esta norma especifica las convenciones para la estructuración del contenido del mensaje y la representación de los datos contenidos dentro de esas estructuras, que se requieren para transferir información entre un instrumento clínico y de un sistema informático, permitiendo establecer una relación lógica de comunicación de texto para enviar resultados, solicitud o información demográfica en un formulario estándar e interoperable.

Siemens Medical Solutions Diagnostics (2007), explica detalladamente como los mensajes son enviados por el Immulite mediante el estándar ASTM E1394, a continuación se muestran un conjunto de figuras tomadas del autor para facilitar su entendimiento.

En la Figura 6 se observa la jerarquía de la estructura del mensaje que tiene convenido esta norma, se identifican entidades en el sector de laboratorio clínicos como paciente, orden y resultados entre las marcas de inicio con un encabezado y fin con un terminal.

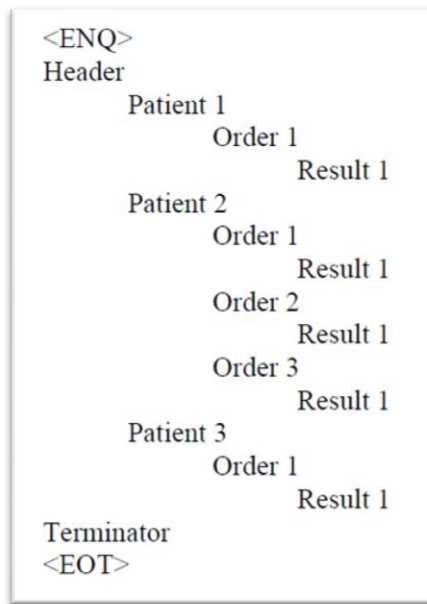


Figura 6. Jerarquía de un Mensaje ASTM E1394.
Fuente: Siemens Medical Solutions Diagnostics (2006).

El mensaje inicia con un encabezado (Header), se encuentra conformado por el campo inicial tipo de registro 'H', y otros que identifican información básica del Equipo que emite el resultado, en la Figura 7 se muestra su estructura.

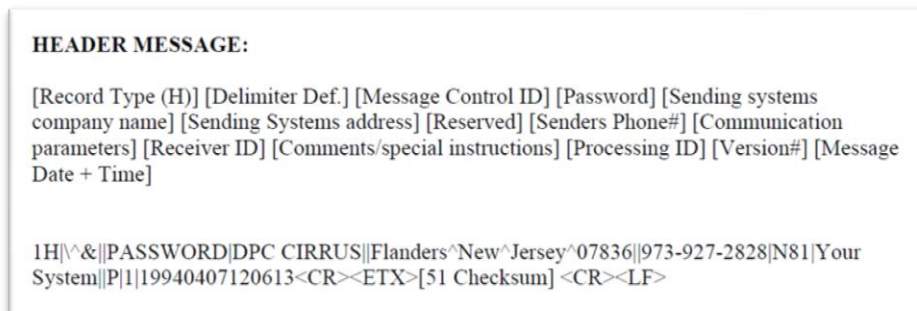


Figura 7. Definición del Mensaje ASTM E1394 - Encabezado.
Fuente: Siemens Medical Solutions Diagnostics (2006).

Luego del encabezado se presenta la información del paciente, se encuentra conformado por el campo tipo de registro 'P', y otros campos que determinan su número de identificación dentro del laboratorio, nombre, dirección, fecha de nacimiento, entre otros que se pueden observar en la Figura 8.

PATIENT MESSAGE:

[Record Type (P)][Sequence #][Practice Assigned Patient ID][Laboratory Assigned Patient ID][Patient ID][Patient Name][Mother's Maiden Name][BirthDate][Patient Sex][Patient Race][Patient Address][Reserved][Patient Phone #][Attending Physician ID][Special Field 1][Special Field 2][Patient Height][Patient Weight][Patients Known or Suspected Diagnosis][Patient active medications][Patients Diet][Practice Field #1][Practice Field #2][Admission and Discharge Dates][Admission Status][Location][Nature of Alternative Diagnostic Code and Classification][Alternative Diagnostic Code and Classification][Patient Religion][Marital Status][Isolation Status][Language][Hospital Service][Hospital Institution][Dosage Category]

2P|1|101||Riker^A|19611102|F||||Bashere<CR><ETX>[2AChecksum] <CR><LF>

Figura 8. Definición del Mensaje ASTM E1394 - Paciente.

Fuente: Siemens Medical Solutions Diagnostics (2006).

La información de la orden inicia por el campo tipo de registro 'O', y otros campos como examen clínico a realizar, fecha y hora de recolección de los datos, y fecha y hora de la obtención de resultados entre otros que son identificados en la Figura 9. Cabe destacar, que esta línea se repite tantos exámenes posea la orden.

ORDER MESSAGE:

[Record Type (O)][Sequence#][Specimen ID (Accession#)][Instrument Specimen ID][Universal Test ID][Priority][Order Date/Time][Collection Date/Time][Collection End Time][Collection Volume][Collector ID][Action Code][Danger Code][Relevant Clinical Info][Date/Time Specimen Received][Specimen Descriptor, Specimen Type, Specimen Source][Ordering Physician][Physician's Telephone Number][User Field No.1][User Field No.2][Lab Field No.1][Lab Field No.2][Date/Time results reported or last modified][Instrument Charge to Computer System][Instrument Section ID][Report Types][Reserved Field][Location or ward of Specimen Collection][Nosocomial Infection Flag][Specimen Service][Specimen Institution]

3O|1|1550623||^LH|R|19931011091233|19931011091233<CR><ETX>[6C Checksum] <CR><LF>

Figura 9. Definición del Mensaje ASTM E1394 - Orden.

Fuente: Siemens Medical Solutions Diagnostics (2006).

La sección con la información de los resultados de la orden inicia por el campo tipo de registro 'R', y otros campos como secuencia, código del examen clínico, resultado obtenido, unidad, rangos de referencia, fecha y hora de inicio y completado, entre otros identificados en la Figura 10.

RESULT MESSAGE:

[Record Type (R)][Sequence #][Universal Test ID][Data (result)][Units][Reference Ranges]
[Result abnormal flags][Nature of Abnormality Testing][Result Status][Date of change in
instruments normal values or units][Operator ID][Date\Time Test Started][Date\Time Test
Completed][Instrument ID]

```
4R|1|^^LH|8.2|mIU/mL|.7\7^400\400|N|N|F||test|19931011091233|19931011091233|DPCCIRR  
US<CR><ETX>[8FChecksum] <CR><LF>
```

Figura 10. Definición del Mensaje ASTM E1394 - Resultado.

Fuente: Siemens Medical Solutions Diagnostics (2006).

Por último el delimitador de fin de archivo que inicia por el campo tipo de registro 'L' y otros identificados en Figura 11.

MESSAGE TERMINATOR:

[Record Type ID (L)][Sequence Number][Termination Code]

```
5L|1|N<CR><ETX>[Checksum]<CR><LF>
```

Figura 11. Definición del Mensaje ASTM E1394 - Terminal.

Fuente: Siemens Medical Solutions Diagnostics (2006).

Estándar HL7

Gaya (1999), expone que "el desarrollado del protocolo Health Level Seven (HL7), surge como una superación del ASTM, y se está trabajando para la integración de los códigos SNOMED (Nomenclatura Sistematizada de Medicina - Systematized Nomenclature of Medicine) en él".

Al respecto, SNOMED International, una división de la CAP (College of American Pathologists - Colegio Americano de Patólogos), define SONOMED como la terminología clínica integral, multilingüe y codificada de mayor amplitud, precisión e importancia desarrollada en el mundo, terminología internacional usada

para el desarrollo de registros médicos electrónicos integrados mediante la presentación de un subconjunto de la terminología de referencia.

Health Level Seven Inc (HL7), se define como una organización sin ánimo de lucro, acreditada por la ANSI (American National Standards Institute - Instituto Nacional Americano de Estándares) para el desarrollo de estándares en el área de la salud, específicamente en el dominio de los datos clínicos y administrativos, aunque sin excluir otros como los de farmacia, equipos de diagnóstico médico y de imagen, entre otros. Colectivamente, los estándares que desarrolla se conocen como el estándar HL7.

Villegas y Tamura (2008), especifican que el estándar HL7 surgió como respuesta a la necesidad de resolver el problema de interoperabilidad entre los distintos y numerosos sistemas informáticos que se requieren en las áreas o unidades al interior de una entidad o centro de salud, y por parte de los diversos tipos de actores que intervienen en un sistema de salud.

Entre las diversas versiones existentes del estándar HL7, Villegas y otros (2008), mencionan que la versión 3 de HL7, se basa en un lenguaje de especificación cuya definición se apoya en lenguajes bien definidos XML y UML (Unified Modeling Language - Lenguaje Unificado de Modelado) cuyas reglas léxicas y sintácticas están formalmente definidas, aunque sus reglas semánticas lo están semiformalmente. Los elementos claves en el estándar HL7 del origen y evolución del estándar versión 3, se identifican varios aspectos clave, en cuanto a la estructura:

- La adopción de modelos y estándares ya establecidos y ampliamente reconocidos en el área de las tecnologías de la información y comunicaciones.
- Un fuerte énfasis en definiciones y vocabulario controlado en todas las especificaciones, al punto de existir un comité técnico específico de

vocabulario para la codificación del significado preciso y “libre de ambigüedades” de los datos representados y transmitidos.

- El énfasis en lograr una semántica de alto nivel (metamodelamiento) como base para mejorar la interoperabilidad.
- Establecimiento del modelo de información de HL7, el cual actualmente se compone del modelo de información de referencia (HL7 Reference Information Model - RIM).

Y en cuanto a los aspectos dinámicos del manejo del estándar:

- Adopción de guías de implementación para definición y aplicación de los estándares.
- Adopción de estándares por votación, garantizando tres aspectos: (1) Consenso; (2) Apertura en la participación; (3) Balance en la representación y de los intereses de todos los tipos de actores del sector salud.

Arquitectura Tecnológica Empresarial

García (2007), Definimos arquitectura tecnológica empresarial como el conjunto de componentes, servicios y procedimientos que dirigen y soportan el desarrollo y funcionamiento de una solución de negocio para una organización, garantizando la calidad, completitud y operatividad del resultado. Desde esta perspectiva, una arquitectura debería contemplar:

- Los componentes software orientados a satisfacer requerimientos de negocio o tecnológicos (arquitectura de ejecución).

- Las herramientas, estándares y metodologías para el óptimo desarrollo de software (arquitectura de desarrollo).
- La infraestructura técnica (hardware y software) para soportar el proceso de desarrollo y operación del sistema (arquitectura técnica).
- Los mecanismos, técnicas y componentes que garanticen la operación, nivel de servicio, mantenimiento y seguridad de la solución software (arquitectura de operación).

Integración de Aplicaciones Empresariales

Según Frantz (2008), la integración de aplicaciones empresariales EIA (Enterprise Application Integration) son soluciones de integración cuyo objetivo es mantener sincronizadas un conjunto de aplicaciones, aportar nuevas funcionalidades y que se ejecutan en el lado del servidor dentro de la misma empresa, con el objetivo de mantener las aplicaciones funcionando en sincronía y de forma exógena poder integrar sus funcionalidades. Este tipo de solución suele conectar dos o más aplicaciones por medio de un flujo exógeno de datos y/o comandos, capaz de integrar funcionalidades de las aplicaciones involucradas sin que las aplicaciones conozcan la solución. Es decir, una solución EAI aporta de forma exógena funcionalidad de una aplicación a la otra, o incluso, nuevas funcionalidades, además de mantener las aplicaciones independientes una de las otras y coordinarlas.

Por lo tanto, y en primer lugar, Frantz (2008), insiste que un proyecto de EAI implica implementar una arquitectura bajo la cual las distintas aplicaciones se comuniquen entre sí. En consecuencia, esto conlleva el desarrollo de conectores que posibilitan la interfaz de aplicaciones mediante el uso de distintos protocolos de comunicación.

Middleware

Es un módulo intermedio que actúa como conductor entre sistemas permitiendo a cualquier usuario de sistemas comunicarse con varias fuentes de información u otros sistemas que se encuentran conectadas por una red.

Para Bernstein (1996), middleware es un software que ayuda a una aplicación interactuar o comunicarse con otras aplicaciones, redes, hardware y/o sistemas operativos. Este software ayuda a los programadores de librarlas de ellos complejas conexiones necesarias en un sistema distribuido. Ya que, proporciona herramientas para mejorar la calidad, la seguridad, el paso de mensajes, servicios de directorio, archivo servicios, entre otros, que pueden ser invisibles para el usuario.

Arquitectura de Software

Presman (2002), hace referencia que "la Arquitectura de Software de un sistema de programa o computación es la estructura de las estructuras del sistema, la cual comprende los componentes del software, las propiedades de esos componentes visibles externamente, y las relaciones entre ellos".

Para Reynoso (2006), a grandes rasgos la Arquitectura de Software es una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. La vista arquitectónica es una vista abstracta, aportando el más alto nivel de comprensión y la supresión o diferimiento del detalle inherente a la mayor parte de las abstracciones.

Igualmente la IEEE 1471 (2004), en el cual se describen estándares para la arquitectura de un sistema de software por la IEEE (Institute of Electrical and Electronics Engineers - Instituto de Ingenieros en Electricidad y Electrónica), menciona que "la Arquitectura de Software es la organización

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

En este orden de ideas, la arquitectura de software, está referida al diseño y la implementación de las estructuras de software de alto nivel, que como resultado, se obtienen un conjunto de elementos arquitectónicos, que ensamblados de manera adecuada logran satisfacer la totalidad de las funcionalidades y requerimientos de desempeño de un producto software.

Estilo

En el contexto de arquitectura de software, se observan estilos, los cuales pueden estar asociados a los distintos software, cada forma arquitectónica tiene gran relación y se asocia con herramientas, conceptos, experiencias y problemas. Esas cualificaciones arquitectónicas se denominan estilos, o alternativamente patrones de arquitectura, para Reynoso y otros (2004), "los estilos sólo se manifiestan en arquitectura teórica descriptiva de alto nivel de abstracción; los patrones, por todas partes".

Perry y otros (1992), define que "un estilo es un concepto descriptivo que define una forma de articulación u organización arquitectónica. El conjunto de los estilos cataloga las formas básicas posibles de estructuras de software, mientras que las formas complejas se articulan mediante composición de los estilos fundamentales".

Según Pressman (2002), un estilo describe una categoría de sistema que abarca un conjunto de componentes que realizan una función requerida por el sistema, un conjunto de conectores que posibilitan la comunicación, la coordinación y cooperación entre los componentes, las restricciones que definen como se integran los componentes para conformar el sistema, y los modelos semánticos que facilitan al

diseñador el entendimiento de todas las partes del sistema. El estilo arquitectónico es también un patrón de construcción.

Patrón

Reynoso (2004), propone que "un sistema bien estructurado está lleno de patrones", se podría ver a un patrón como una solución a un problema determinado, una codificación de algo específico basado en un conocimiento específico acumulado por la experiencia.

Según Fowler (2002), "cada patrón describe un problema que ocurre una y otra vez en nuestro ambiente, y luego describe el núcleo de la solución a ese problema, de tal manera que puedes usar esa solución un millón de veces más, sin hacer jamás la misma cosa dos veces".

Si se hace una primera clasificación de los patrones software atendiendo al nivel de abstracción de éstos, pudieran quedar: los patrones arquitectónicos, de diseño o centrados en el código, entre otros.

Patrón Arquitectónico

Los patrones de arquitectura representan el nivel más alto en el sistema de patrones propuesto en Bruschmann (1996) y ayudan a especificar la estructura fundamental de una aplicación. Según este autor, los patrones de arquitectura se pueden ver como la descripción de un problema en particular y recurrente de diseño, que aparece en contextos de diseño arquitectónicos específicos, y representa un esquema genérico demostrado con éxito para su solución. El esquema de solución se especifica mediante la descripción de los componentes que la constituyen, sus responsabilidades y desarrollos, así como también la forma en que estos colaboran entre sí.

Patrón de Diseño

Bruschmann (1996), define que un patrón de diseño provee un esquema para refinar los subsistemas o componentes de un sistema de software, o las relaciones entre ellos. Describe la estructura comúnmente recurrente de los componentes en comunicación, que resuelve un problema general de diseño en un contexto particular.

Son menores en la escala de abstracción que los patrones arquitectónicos, además tienden a ser independientes de los lenguajes y paradigmas de programación. Su aplicación tiene efectos sobre subsistemas, debido a que especifica a un mayor nivel de detalle, sin llegar a la implementación, el comportamiento de los componentes del subsistema.

Marco de trabajo (Framework)

El concepto de framework se emplea en muchos ámbitos del desarrollo de sistemas software, se pueden encontrar frameworks para el desarrollo de aplicaciones web, aplicaciones de escritorio, aplicaciones médicas, para el desarrollo de juegos, y para cualquier ámbito que se le pueda ocurrir a uno.

Carrascoso y Chaviano (2008), expresan que en general, el término framework, se refiere a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para desarrollar una aplicación concreta.

Modelo Arquitectural

Reynoso (2006), al proponer una definición de Arquitectura de Software, en una perspectiva de modelo arquitectural, esta concierne a decisiones sobre organización, selección de elementos estructurales, comportamiento, composición y estilo arquitectónico susceptibles de ser descriptas a través de las cinco vistas clásicas del modelo 4+1 de Kruchten.

Modelo 4+1 vistas

El conjunto de diagramas requeridos para representar el diseño y distribución de los componentes que se encuentran incluidos para el desarrollo de un software pueden estar enmarcada bajo el modelo de 4 + 1 vistas, propuesto por Kruchten (1995), quien define "es una propuesta que establece las diferentes perspectivas a través de las cuales se puede presentar el diseño y arquitectura de un sistema de software".

El modelo 4+1 describe la arquitectura del software usando cinco vistas concurrente, tal como se muestra en la Figura 12.

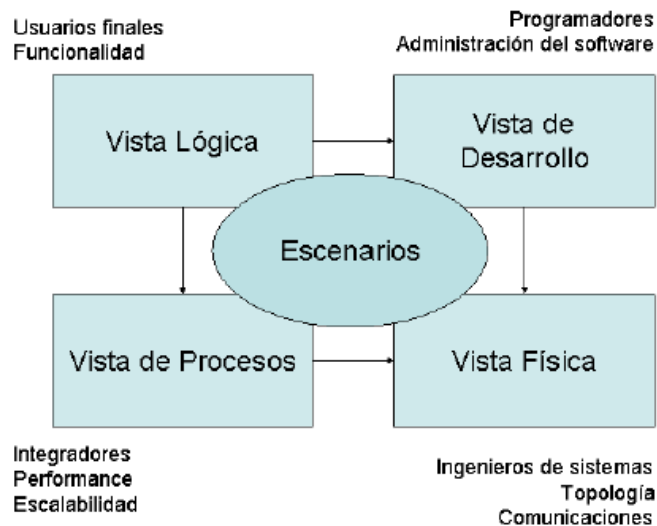


Figura 12. Modelo 4 + 1 Vistas. Fuente: Kruchten (1995).

- La vista lógica describe el modelo de objetos del diseño cuando se usa un método de diseño orientado a objetos. Para diseñar una aplicación muy orientada a los datos, se puede usar un enfoque alternativo para desarrollar algún otro tipo de vista lógica, tal como diagramas de entidad-relación.
- La vista de procesos describe los aspectos de concurrencia y sincronización del diseño.
- La vista física describe el mapeo del software en el hardware y refleja los aspectos de distribución.
- La vista de desarrollo describe la organización estática del software en su ambiente de desarrollo.

Arquitectura Orientada a Servicios

Carrascoso y Chaviano (2008), La Arquitectura Orientada a Servicios es un concepto de arquitectura de software que define la utilización de servicios como construcciones básicas para el desarrollo de aplicaciones. Es una arquitectura de una aplicación donde las funcionalidades se definen como servicios independientes, con interfaces accesibles, bien definidas, que pueden ser llamadas en secuencias dadas para formar procesos de negocios.

Reynoso, 2006. SOA ha surgido como la mejor manera de afrontar el desafío de hacer más con menos recursos. Promete hacer la reutilización y la integración mucho más fáciles, ayudando a reducir el tiempo de desarrollo y aumentando la agilidad organizacional. No sorprendentemente, el 80% de las organizaciones de IT están implementando aplicaciones usando SOA con web services subyacentes. SOA proporciona mayor flexibilidad para afrontar los cambios tanto en el ambiente de negocios como en la infraestructura tecnológica”

SOA es un estilo de arquitectura en el cual se exponen los procesos de negocio del sistema a construir como servicios independientes de alta cohesión y bajo

acoplamiento que encapsulan dichos procesos y pueden ser invocados a través de interfaces bien definidas.

Carrascoso y Chaviano (2008), mencionan que los principales componentes de una arquitectura orientada a servicios son:

- **Servicios**, un servicio de negocio es un componente reusable de software, con significado funcional completo, y que está compuesto por:
 - Contrato: especificación de la finalidad, funcionalidad, forma de uso y restricciones del servicio.
 - Interfaz: mecanismo de exposición del servicio a los usuarios.
 - Implementación: debe contener la lógica o el acceso a datos.

Tipos de servicios.

- Servicios básicos: pueden estar centrados en datos o en lógica y encapsulan funcionalidades como cálculos complejos, acceso a datos y reglas complejas de negocio.
 - Servicios intermediarios: servicios adaptadores, façades, entre otros. Suelen ser servicios sin estado.
 - Servicios de proceso: servicios de negocio que encapsulan la lógica de proceso. Suelen conservar estado y pueden residir en herramientas BPM.
 - Servicios públicos: servicios accesibles por terceros (fuera de la organización).
- **Repositorio de servicios**, proporciona facilidades para descubrir servicios y adquirir la información necesaria para su uso, en particular fuera del alcance temporal y funcional del proyecto en el que se crearon.

Además de la propia información de contrato, los repositorios pueden proporcionar información acerca de:

- Localización

- Personas de contacto.
 - Restricciones Técnicas
 - Service Level Agreements (SLAs).
- **Bus de servicios**, el bus de servicios es el elemento de las arquitecturas SOA que conecta los servicios con sus consumidores y que proporciona:
 - Conectividad: el propósito principal de un bus de servicios es interconectar a los participantes de una arquitectura SOA.
 - Soporte a la heterogeneidad de tecnologías: debe ser capaz de conectar a participantes basados en distintos lenguajes de programación, sistemas operativos, entornos de ejecución y protocolos de comunicación.
 - Soporte a la heterogeneidad de paradigmas de comunicación: debe ser capaz de mantener distintos modos de comunicación (por ejemplo comunicaciones síncronas y asíncronas).
 - **Consumidores de servicios**, definimos consumidores de servicios como aquellos elementos de una arquitectura SOA que:
 - Pueden descubrir servicios a través de un repositorio.
 - Realizan llamadas a los mismos de acuerdo al contrato y a través del interfaz definido a tal efecto.

Servicios Web

Las Arquitecturas Orientadas a Servicios, en contraste con las Arquitecturas Orientadas a Objetos, están formadas por servicios de aplicación altamente interoperables y débilmente acoplados. Donde estos servicios se pueden ver como la evolución en complejidad de un componente distribuido, diferenciándose en que son Alves y otros (2006):

- Mucho menos acoplados con sus aplicaciones cliente que los componentes.
- Menor granularidad que los componentes.

- No son diseñados e implementados necesariamente como parte de una aplicación end-to-end.
- Son controlados y administrados de manera independiente.
- Expone su funcionalidad a través de protocolos abiertos e independientes de plataforma. Incluso arriesgando el rendimiento y consumo de recursos.
- Son transparentes de su localización en la red, de esta manera garantizan escalabilidad y tolerancia a fallos.

Así mismo, Alves y otros (2006), expresan que los servicios web comunican aplicaciones, lo cual permite que se comparta información independientemente de cómo se hayan creado, cuál sea el sistema operativo o la plataforma en que se ejecutan y cuáles sean los dispositivos utilizados para obtener acceso a ellas. La comunicación se caracteriza por el intercambio de mensajes a través de lenguaje de etiquetado extensible XML y por ser independientes del protocolo de comunicación. Para lograr esta independencia, el mensaje XML se envuelve de manera apropiada para cada protocolo gracias a la creación del protocolo de transporte SOAP (Simple Object Access Protocol: Protocolo de Acceso a Objetos Simples).

El Lenguaje de Descripción de los Servicios Web (WSDL: Web Services Description Language), expresado en XML, describe cómo acceder al servicio, de qué funciones dispone, qué argumentos necesita y qué devuelve cada uno.

La otra tecnología fundamental implicada es la de Descripción Universal, Descubrimiento e Integración (UDDI: Universal Description, Discovery, and Integration). UDDI es un directorio de servicios web donde se puedan publicar los servicios ofrecidos, dar características del tipo de servicio, y realizar búsquedas.

En resumen, SOAP define un protocolo XML para la interoperabilidad básica entre servicios, WSDL introduce un lenguaje común para describir servicios y UDDI provee la infraestructura requerida para publicar y descubrir servicios

programáticamente. Juntas, estas especificaciones permiten a las aplicaciones interactuar siguiendo un modelo débilmente acoplado e independiente de la plataforma subyacente.

Bus de Servicios Empresariales

TIBCO Software Inc (2008), un bus de servicios empresariales (ESB) es una solución de integración distribuida, basada en los mensajes y en estándares abiertos. La función de un ESB es proporcionar una comunicación fiable entre los distintos recursos tecnológicos tales como aplicaciones, plataformas y servicios, que están distribuidos en múltiples sistemas por toda la empresa. A medida que los departamentos de TI se centran cada vez más en el diseño de SOA para reducir los costes de desarrollo y para aumentar la agilidad del negocio, los ESB se están convirtiendo en un primer paso clave para el establecimiento de una SOA empresarial. Los ESB constituyen los cimientos de una SOA y pueden complementarse con capacidades de productividad adicionales, como la orquestación de servicios y los registros. Este documento trata sobre los requisitos para el establecimiento de un ESB capaz de satisfacer las necesidades de una SOA lista para operar en un marco empresarial.

Bases Legales

Constitución de la República Bolivariana de Venezuela

La Constitución de la República Bolivariana de Venezuela (1999), establece en su artículo 110 lo siguiente:

El Estado reconocerá el interés público de la ciencia, la tecnología, el conocimiento, la innovación y sus aplicaciones y los servicios de información necesarios por ser instrumentos fundamentales para el desarrollo económico, social y político del país, así como para la

seguridad y soberanía nacional. Para el fomento y desarrollo de esas actividades, el Estado destinará recursos suficientes y creará el sistema nacional de ciencia y tecnología de acuerdo con la ley. El sector privado deberá aportar recursos para los mismos. El Estado garantizará el cumplimiento de los principios éticos y legales que deben regir las actividades de investigación científica, humanística y tecnológica. La ley determinará los modos y medios para dar cumplimiento a esta garantía.

El gobierno de Venezuela dando cumplimiento a la norma constitucional transcrita anteriormente, aprobó la Ley Orgánica de Ciencia, Tecnología e Innovación (2001), que tiene por objeto tal y como lo señala su artículo 1:

El presente Decreto-Ley tiene por objeto desarrollar los principios orientadores que en materia de ciencia, tecnología e innovación, establece la Constitución de la República Bolivariana de Venezuela, organizar el Sistema Nacional de Ciencia, Tecnología e Innovación, definir los lineamientos que orientarán las políticas y estrategias para la actividad científica, tecnológica y de innovación, con la implantación de mecanismos institucionales y operativos para la promoción, estímulo y fomento de la investigación científica, la apropiación social del conocimiento y la transferencia e innovación tecnológica, a fin de fomentar la capacidad para generación, uso y circulación del conocimiento y de impulsar el desarrollo nacional.

Sin embargo, para que este esfuerzo de incorporar a Venezuela en la era de la tecnología y de la información, alcance un nivel adecuado, se hace necesario la promulgación de un conjunto de instrumentos legales que proporcionen el marco institucional al desarrollo armonioso del sector y a su democratización y, que precisamente para lograr los objetivos tanto de la norma constitucional como de la Ley Orgánica de Ciencia, Tecnología e Innovación, se hizo necesario promover al mismo tiempo las condiciones de seguridad que inspirarán suficiente confianza tanto a los administradores de las plataformas que brindan servicios tecnológicos como al usuario en general.

Sistema de Variables

Según Hernández, Fernández y Baptista (2006), “una variable es una propiedad que puede fluctuar y cuya variación es susceptible de medirse u observarse”. Por ello, las variables adquieren valor para la investigación científica cuando llegan a relacionarse con otras variables, es decir, si forma parte de una hipótesis o una teoría. En este caso se les suele denominar constructor o construcciones hipotéticas.

Del mismo modo, Arias (2006), señala que una variable “es una características o cualidad; magnitud o cantidad, que puede sufrir cambios, y que es objeto de análisis, medición, manipulación o control en una investigación”.

Operacionalización de las Variables

Definición Conceptual

En el proceso de investigación, una vez planteado y esclarecido el asunto o problema y toda vez que se hayan establecido las preguntas de investigación, es imprescindible definir los términos o variables que están siendo incluidos. Al respecto, se debe clarificar lo que es la definición conceptual de la variable.

La variable conceptual es de índole teórica, debe referir qué se entiende, qué es, en qué consiste. La variable conceptual, presentada en la Tabla 1, presenta la variable conceptual referida al centro del estudio de la presente investigación.

Tabla 1. Definición Conceptual

Variable	Definición
Arquitectura Orientada a Servicios que permita la interoperabilidad de equipos médicos y el Sistema de Información de Laboratorio	Esta constituido como el conjunto de componentes, interfaces y relaciones, que a través del uso de patrones, abstracciones y metodologías, ofrecen un marco de referencia para el desarrollo de un modelo arquitectural que permita la interoperabilidad de equipos médicos y el sistema de información de laboratorio.

Fuente: El Autor de la Investigación

Definición Operacional

De acuerdo con Méndez (2008), “implica desglosar la variable por medio de un proceso de deducción lógica en indicadores, los cuales se refieren a situaciones específicas de las variables”. Es decir, descender a un nivel de abstracción de la misma. Son las que especifican los procedimientos que describe las actividades que un observador debe registrar para poder medir la variable.

En tal sentido, la variable definida en la investigación, se operacionaliza en consideración de cuatro dimensiones: (1) Requerimientos Funcionales, (2) Requerimientos No Funcionales, (3) Diseño de Arquitectura de Software y finalmente (4) Desarrollo de Modelo Arquitectural. En la Tabla 2 se muestra la operacionalización de las variables, donde se reflejan los indicadores respectivos de cada una de las dimensiones.

Tabla 2. Operacionalización de las Variables

Objetivo Específicos	Dimensión	Indicadores	Técnica
<p>Identificar los requerimientos funcionales y no funcionales que permiten una integración de equipos médicos y el Sistema de Información de Laboratorio (SIL).</p>	<p>Requerimientos Funcionales</p>	<p>Tecnología de Información. Sistema de Información. Sistema de Información de Laboratorios.</p>	<p>PROYECTO FACTIBLE - DESCRIPTIVA - REVISIÓN BIBLIOGRÁFICA</p>
	<p>Requerimientos No Funcionales</p>	<p>Operaciones Básicas. Arquitectura Tecnológica Empresarial. Protocolos estándar de comunicación.</p>	
<p>Diseñar el modelo arquitectural que permitirá la selección de los componentes y los mecanismos de integración para la interoperabilidad de equipos médicos y el Sistema de Información de Laboratorio (SIL).</p>	<p>Diseño de Arquitectura de Software</p>	<p>Estilos arquitecturales. Patrón Arquitectural. Elementos Arquitectónicos. Servicios. Bus de Servicio Empresarial. Exploración.</p>	
<p>Desarrollar el modelo arquitectural en un entorno de integración de equipos médicos y el Sistema de Información de Laboratorio (SIL) del laboratorio clínico Mascia.</p>	<p>Desarrollo de Modelo Arquitectural</p>	<p>Patrón de diseño. Middleware. Metodología de Desarrollo. Iteraciones.</p>	

Fuente: Autor de la investigación.

CAPITULO III

MARCO METODOLOGICO

Naturaleza de la Investigación

La presente investigación de una arquitectura orientada a servicios para la interoperabilidad con equipos médicos. Caso de estudio laboratorio clínico Mascia, estará enmarcada de acuerdo a los objetivos planteados, en una investigación de campo con carácter descriptivo bajo la modalidad de proyecto factible. De acuerdo con la definición presentada por el "Manual para la Presentación de Trabajo de Grado de la Universidad Centroccidental " Lisandro Alvarado" UCLA(2002), se entiende por investigación de campo “la aplicación del método científico en el tratamiento de un sistema de variables y sus relaciones, las cuales conducen a conclusiones y al enriquecimiento de un campo del conocimiento o disciplina inherente a la especialidad, con la sustentación de los experimentos y observaciones realizadas”.

Con respecto al tipo descriptivo, según Hernández, Fernández y Baptista (2006), “los estudios descriptivos buscan especificar las propiedades importantes de personas, grupos, comunidades o cualquier otro fenómeno que sea sometido a análisis”. En este sentido, se describirá con detalle el conjunto de abstracciones necesarias para diseñar y desarrollar el modelo arquitectural para la integración de equipos médicos y los sistemas de información de laboratorio.

El estudio se apoya en una investigación bajo la modalidad proyecto factible, por cuanto representa el diseño de un modelo arquitectural viable para la solución de una problemática planteada en una organización específica, tal como lo especifica el "Manual para la Presentación de Trabajo de Grado" UCLA (2002), "se entenderá por

estudios de proyecto una proposición sustentada en un modelo viable para resolver un problema práctico, tendente a satisfacer necesidades institucionales o sociales".

Estudio de Factibilidad

Factibilidad Tecnológica

El desarrollo del modelo arquitectural que permitirá la integración de los equipos médicos con un Sistema de Información de Laboratorio, requiere de recursos de hardware y software.

Enmarcados en el software requerido, se usarán herramientas de desarrollo y despliegue de la arquitectura de licencias libres o código abierto, haciéndolo factible a nivel tecnológico, ya que no representa inversión extra para su uso, entre ellas: herramientas de modelado, lenguaje de programación, entorno de desarrollo, componentes de arquitectura, bases de datos, servidores de servicios, frameworks de desarrollo, entre otros.

En cuanto a los recursos de hardware necesarios, se cuenta con un computador para el desarrollo y despliegue de la arquitectura. Por otro lado, el laboratorio clínico Mascia, cuenta con los equipos médicos objeto de estudio en la presente investigación como los son: el Immulite 2000 de Siemens y el Coulter LH 500 de Bekman Coulter. El Immulite 2000, por sus características está destinado a laboratorios que atiendan alto volumen de pacientes en el área de hormonas, según el Gerente General en el área de Bioanálisis del laboratorio clínico Mascia, dicho laboratorio puede contar con el equipo, ya que maneja altos volúmenes de pacientes, que provienen de las ocho sucursales ubicadas en el Edo. Lara, haciéndolo asequible desde hace varios años para dicha institución.

Las relaciones comerciales del laboratorio Mascia con los proveedores de dichos equipos, agilizo los procesos para ubicar las especificaciones de los cables de conexión y los drivers necesarios para extraer la información de los resultados de las pruebas diagnosticas realizadas por ambos equipos, además ambos equipos permiten la lectura de códigos de barras provenientes de su SIL.

Factibilidad Institucional

Laboratorio clínico Mascia contando con 70 años de experiencia en el sector y siendo uno de los primeros laboratorios del estado Lara, reconoce que al análisis clínico es el pilar fundamental para el diagnóstico y tratamiento de las enfermedades medicas, por tanto, el desarrollo de una arquitectura para la integración de equipos médicos con el Sistema de Información de Laboratorio, fortalecerá los procesos permitiendo mejorar los tiempos de respuestas, obtener mayor confiabilidad en los informes de resultados clínicos, optimizar la gestión de los recursos humanos, entre otros, enfocándose entonces en mayor grado a la calidad del servicio al paciente.

Factibilidad Operativa

Laboratorio clínico Mascia, cuenta con una unidad de Plataforma Tecnológica con personal altamente calificado en el área de tecnología de información. Dicho personal, está en la capacidad de asumir el proyecto, y a futuro, guiados por el marco referencial desarrollado en la presente investigación, puedan implementar la arquitectura orientada a servicios en la infraestructura instalada del laboratorio, logrando así una factibilidad operativa.

Factibilidad Económica

La inversión requerida no representa un costo adicional en los aspectos relacionados con el licenciamiento del software a utilizar para el diseño y desarrollo del modelo arquitectural, la tendencia es la selección total de licencias libres o de

código abierto. De este modo, se evidencia la factibilidad económica para desarrollar el modelo arquitectural para la interoperabilidad con equipos médicos.

Procedimiento de la Investigación

A fin de abordar los requisitos involucrados en un proyecto factible, se describen las tres fases a ser ejecutadas en la siguiente investigación. En una fase inicial, se identifican las necesidades de la arquitectura orientada a servicios, a fin de determinar desde los aspectos funcionales el comportamiento interno, detalles técnicos, manipulación de datos y otras funcionalidades específicas que mostrarán como los casos de usos serán llevados a la práctica, los cuales, serán complementados por los requisitos no funcionales, que se enfocan en cambio en el diseño o la implementación.

A partir de los resultados de la fase anterior, en una segunda fase, se diseñará el modelo arquitectural orientado a servicios que permita la interoperabilidad de mensajes entre los equipos médicos con el Sistema de Información de Laboratorio, definiendo para ellos los componentes y sus interrelaciones. Finalmente, en la tercera fase, se desarrollará la arquitectura propuesta, a fin de dar solución a la problemática planteada en esta investigación. Para las distintas fases, se hará el uso de las utilidades ofrecidas por la Ingeniería de Software, y siguiendo la metodología de Programación Extrema XP.

Técnicas e Instrumentos de Recolección de Datos

Se utilizará la técnica de revisión bibliográfica, que según Sabino (2004), es aquella técnica empleada “cuando los datos proceden de investigaciones anteriores, son llamados secundarios porque han sido previamente obtenidos por otros investigadores, elaborados y procesados de acuerdo con quienes inicialmente los manejaron”, a fin de soportar los conocimientos necesarios para el diseño y desarrollo

de la arquitectura a proponer, mediante los conceptos, características, tipos y modelado de una Arquitectura Orientada a Servicios (SOA) desplegada sobre un Bus de Servicio Empresarial (ESB) y lograr el enrutamiento de los mensajes entre los equipos médicos y el Sistema de Información de Laboratorio (SIL), a través de un repositorio de datos intermedio.

Por otro lado, se deberá consultar los manuales y especificaciones de los equipos médicos Immulite 2000 de Siemens y Coulter LH 500 de Bekman Coulter, para determinar los tipos, protocolos y estándares de comunicación utilizados que permitirán obtener los informes de resultados de los análisis clínicos.

CAPÍTULO IV

PROPUESTA DEL ESTUDIO

Introducción

La Arquitectura Orientada a Servicios (SOA), está definida por la W3C, por el "conjunto de componentes que pueden ser invocados, cuyas descripciones de interfaces se pueden publicar y descubrir". Al respecto, Microsoft Corporation (2006), menciona que SOA, "supone una estrategia general de organización de los elementos de TI, de forma que una colección abigarrada de sistemas distribuidos y aplicaciones complejas, se pueda transformar en una red de recursos integrados, simplificada y sumamente flexible". Por ello, la presente investigación tendrá como resultado final un modelo arquitectural basado en SOA, como alternativa para la integración de equipos médicos con los Sistemas de Información de Laboratorio SIL. La construcción de dicha arquitectura está guiada en base a los requerimientos funcionales y no funcionales, obtenidos como conclusión de los resultados en la fase de exploración.

La estructura que surge al identificar los elementos, las propiedades visibles y las relaciones del sistema propuesto en la presente investigación, hace referencia a una arquitectura de software, que para su diseño y desarrollo es necesario establecer un conjunto de pasos organizados que guíen el cómo y bajo qué condiciones, principios, técnicas y configuraciones evolucionan dichos elementos y su comunicación, para ello se requiere de una metodología de desarrollo como marco de referencia para planificar una serie de tareas y controlar el orden de su ejecución desde la fase inicial hasta la fase final del ciclo de vida del software.

En este sentido, se ha planteado utilizar la metodología de desarrollo ágil Programación Extrema (XP), el desarrollo ágil se acopla a la presente investigación para garantizar el crecimiento iterativo incremental que conlleva un desarrollo donde los procesos son enfocados en servicios que puede ir creciendo con el tiempo al integrar nuevos equipos médicos, nuevos estándares de mensajes, protocolos de comunicación, etc. Se aplicaron las fases de XP Exploración, Planificación de la Entrega, Iteraciones, dejando fuera Producción, Mantenimiento y Muerte del Proyecto.

La fase de exploración, implicó la revisión del contexto del sistema y los intereses de los diferentes actores involucrados para la especificación de requerimientos funcionales y no funcionales que delimitan el alcance del mismo, y justifican el uso de SOA como modelo arquitectural de referencia para el desarrollo del proyecto. En esta fase se exploran estilos y patrones arquitecturales, arquitecturas de referencia SOA que aplica un modelado en capas, y las diferentes tecnologías que soportan dicho modelado, como resultado se obtiene el conjunto de tareas que serán considerados en la siguientes fases de XP.

La fase de planificación de entregas parte del conjunto de tareas identificadas en la fase de exploración, las cuales atendiendo a un conjunto de historias de usuarios, son útiles para definir un cronograma de trabajo, establecer orden y prioridades para de las distintas entregas que se obtienen en la presente investigación. La planificación consiste en cinco iteraciones, abarcando para esta investigación las cuatro primeras, una primera iteración donde se establece una aproximación a la arquitectura del sistema, la selección de patrones, herramientas y tecnologías, que facilitaran el desarrollo de la misma en las siguientes iteraciones.

Para esta investigación, el enfoque iterativo incremental propuesto por XP, se observa al establecer un ciclo de vida por cada iteración, que inicia con una planificación basada en casos de usos que delimitan objetivos, continua con el diseño

y desarrollo de los subcomponentes incluidos mediante actividades planificadas y cubiertas, y concluye con la prueba de los mismos, donde una entrega funcional puede formar parte de la siguiente iteración o hasta que una entrega iterativa cumpla con las exigencias pautadas por el cliente para su aceptación en un ambiente de producción.

Exploración

Esta fase consiste en estudiar el contexto del sistema mediante el análisis de las historias de usuario, identificar requerimientos funcionales y no funcionales, y definir los casos de usos, con el objeto de establecer el alcance general de la propuesta arquitectural en términos de sus funcionalidades. Se revisa documentación sobre técnicas para describir arquitecturas de software, aproximación de arquitecturas orientadas a servicios, y se establecen el conjunto de tareas y actividades a ser planificadas con tiempos estimados, considerando que podrían variar cuando se analicen más en detalle en cada iteración.

Contexto del Sistema

La Arquitectura Orientada a Servicios para la interoperabilidad con equipos médicos llamado "SOA-Médico", es un middleware que permite una integración flexible, transparente y segura entre los diferentes equipos médicos con el Sistema de Información de Laboratorio a través de la red.

Para entender el entorno en el que funciona el sistema y las entidades que interactúan, se presenta en la Figura 13 el diagrama de contexto en notación UML, propuesto por Rozanski y Woods (2008), quienes mencionan que el diagrama de contexto se suele presentar como una imagen simple y de alto nivel que muestra los límites del sistema y de sus entidades externas adyacentes.

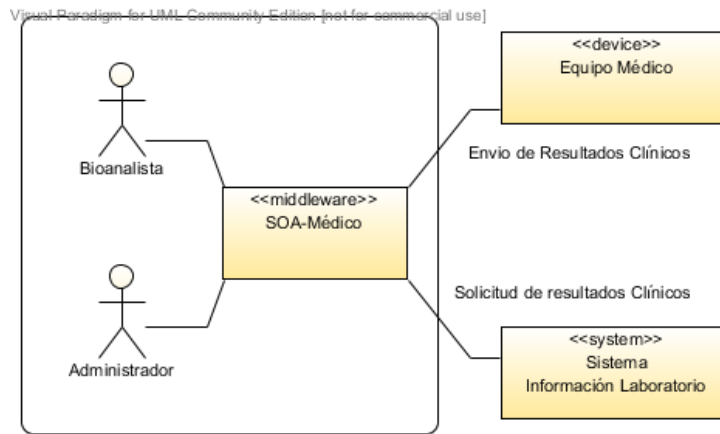


Figura 13. Diagrama de Contexto. Fuente: Autor de la Investigación.

A partir del diagrama de contexto, se observa los límites del sistema a nivel de entidades internas (ubicadas dentro el rectángulo) y externas. Según Rozanski y Woods (2008), una entidad externa, es un ente que se representa como una caja negra para el contexto del middleware, en contraposición, con la entidad interna que tiene interacciones bien definidas con dicho contexto.

Entender la interacción de las distintas entidades, es un paso importante para definir los requerimientos que guían el desarrollo de la arquitectura. A continuación, se explican cada una de ellas:

El equipo médico, es un dispositivo físico que a partir de una muestra médica y determinados reactivos, genera resultados clínicos que son requeridos por el middleware. La forma de comunicación en los equipos en estudio es en serie interfaz RS232-DB9, de tipo síncrona y los protocolo estándar de comunicación de mensajes: ASTM E1394 y HL7 v3. A fines de esta investigación, los equipos se despliegan en la red mediante un cable serial DB9, al cual se le adapta un convertidor DB9-RJ45 a un puerto de red cercano. Además, se ejecuta una software tipo interfaz, que gestiona una comunicación sincrónica para la obtención de resultados y publica la información obtenida en un archivo estandarizado en un directorio o en una cola de un sistema de mensajería.

Sistema de Información de Laboratorio SIL, sistema informático utilizado para llevar la gestión del paciente dentro del laboratorio clínico. Registra la información de las órdenes de trabajo donde se indica, el paciente y los exámenes clínicos a realizar, información que es transmitida a los equipos médicos mediante etiquetas de códigos de barras que son colocadas en las diferentes muestras. Los resultados desplegados por los equipos deben ser expuestos al SIL por el middleware de manera segura y estandarizada.

Bioanalista, profesional capacitado científicamente para efectuar los análisis y exámenes de laboratorio. Se encarga de procesar las muestras médicas en los diferentes equipos médicos y está en el deber, en base a su conocimiento y experiencia, de validar que los resultados obtenidos desde los equipos cubren los parámetros deseables de calidad antes de que sean entregados al paciente. Este profesional, exige que todos los resultados que sean expuestos mediante el middleware para el SIL sean adecuadamente validados por él de manera sencilla y segura.

Administrador, personal capacitado del área de informática del Laboratorio Clínico, encargado de gestionar las tecnologías de información y comunicación. Requiere que el middleware, no afecte el desempeño de las aplicaciones que están ejecutándose actualmente, de manera que ajustándose a la arquitectura tecnológica empresarial actual del laboratorio, sea escalable en el tiempo para poder integrar nuevos equipos.

Revisión de los Requerimientos

Evaluando los intereses de las diferentes entidades que delimitan el contexto del sistema, las historias de usuario y las diversas documentaciones, se describe los requerimientos que debe cubrir el middleware SOA-Médico, en base a las funcionalidades que está obligado a realizar y los atributos de calidad que definen como debe comportarse el sistema en tiempo de ejecución.

Requerimientos Funcionales y no Funcionales

El modelo arquitectural debe desplegarse sobre una arquitectura de servicios que permita una integración de los sistemas dispares, a fin de mantener las aplicaciones que forman parte de la solución sincronizadas, con la restricción que deben seguir desacopladas las una de las otras como antes de la integración, un cambio en una de esta aplicaciones podrá afectar los procesos de negocios que soportan esas aplicaciones.

La información obtenida de los equipos médicos que son considerados para la integración, es ubicada en un directorio específico o una cola de mensajes. El middleware debe estar en la capacidad de lograr una integración con los equipos mediante estilos de integración transferencia de archivos y mensajería, debe establecer un formato estándar para dichos archivos de integración.

Las aplicaciones a integrar tienen modelos de datos muy distintos en concepto y en dominio, los equipos utilizan estándares ASTM E1394 y HL7 v3 para estructurar su información y los SIL su propio modelo de datos basado en bases de datos relacionales, el middleware debe ofrecer un enrutamiento de información y una transformación en dichos paquetes de mensajes que logren la interoperabilidad funcional y semántica de comunicación.

Usar tecnologías de desarrollo multiplataforma que permita su portabilidad en distintos sistemas operativos, y de preferencia de licencias de software libre para asegurar minimizar los costos.

Su diseño debe ser abierto, flexible y escalable a fin de que cualquier especialista en el área de informática, guiado por los lineamientos expuestos en la presente investigación, puedan dar mantenimiento o poder incluir nuevos equipos médicos a la solución. De igual forma, a fin de aprovechar los recursos existentes,

debe estar en la capacidad de separar los componentes que integran el middleware a fin de garantizar que puedan ser desplegados en distintos servidores físicos.

Como es requerida la validación de los resultados por parte del bioanalista antes de ser entregados a los SIL, se requiere que la información obtenida de los equipos médicos, sea guardada en un repositorio de datos, para que sea validada y luego se presente a los SIL mediante servicios web, asegurando independencia de los sistemas. Tener un repositorio intermedio entre los equipos y la SIL da garantía de entrega fiable de los datos.

La información que es controlada por el middleware por estar relacionada con el sector salud es de carácter muy sensible, por ello, se requieren políticas de seguridad al momento de exponer los servicios web a los SIL, y cuentas de usuarios con privilegios a todos los usuarios que requieran información del middleware, a fin de garantizar la confiabilidad de los datos ofrecidos.

Por tanto, SOA-Médico debe ofrecer una interfaz que permita interactuar con su repositorio de datos, mediante la cual, un administrador pueda registrar usuarios y configurar nuevos equipos médicos, y el bioanalista pueda obtener la información de los resultados clínicos para su validación.

En un contexto global, el middleware SOA-Médico debe estar en la capacidad de comunicarse con los equipos mediante una integración de transferencia de archivos y mensajería, recibir paquetes de mensajes enviados que estando en estándar XML, bajo las especificaciones de los estándares ASTM E1394 y HL7, sean transformados en un formato que pueda ser entendido por los componentes de SOA-Médico para su persistencia en el repositorio intermedio. Además, ofrecer al bioanalista, mediante una interfaz para la gestión de dicho repositorio de datos, la capacidad de validar los resultados provenientes de los diferentes equipos, y luego publicarlos como servicios para que sean consumidos por el SIL.

A continuación, en la Figura 14, se muestra el esquema de la propuesta de solución en función del flujo de los procesos requeridos para integrar SOA-Médico como parte de la arquitectura tecnológica empresarial del Laboratorio Clínico Mascia.



Figura 14. Esquema de la Propuesta de Solución. Fuente: Autor de la Investigación.

El flujo de los procesos: (1) inicia con la obtención de los exámenes a realizar por parte del paciente y son transcritos al SIL, el cual genera las etiquetas con código de barra por examen que serán incluidas en los diferentes equipos médicos, (2) el bioanalista o auxiliar toma las muestras necesarias al paciente por cada examen a analizar, (3) se incluyen las muestras con sus respectivos códigos de barras a los diferentes equipos médicos, (4) el middleware SOA- Médico obtiene los resultados desde los equipos médicos y los presenta mediante una interfaz al bioanalista para su validación, (5) el SOA- Médico publica los resultados clínicos validados mediante servicios web para que sean consumidos por el SIL, desde el cual se (6) imprimen los resultados y (7) son entregados al paciente.

Diagrama de Casos de Uso

Las necesidades planteadas en los requerimientos y atributos de calidad expuestos en la sección anterior, permiten identificar como primer bosquejo los siguientes casos de usos, que usando notación UML se presentan de manera gráfica en la Figura 15.

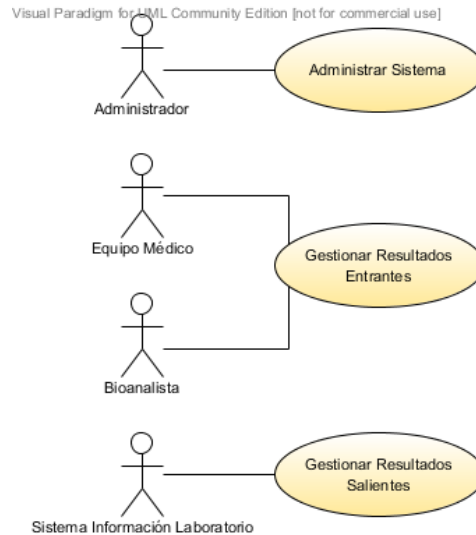


Figura 15. Diagrama de Casos de Uso de SOA-Médico. Fuente: Autor de la Investigación.

Dentro de los actores involucrados, se identifica el actor Usuario en la Figura 16, del cual heredan características el resto de los actores identificados en los casos de uso de la Figura anterior.

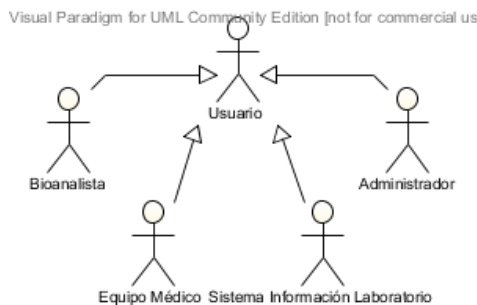


Figura 16. Diagrama de Casos de Uso de SOA-Médico. Fuente: Autor de la Investigación.

Diccionario de Casos de Uso

A continuación se explican los casos de uso identificados:

- **Administrar Sistema:** Este caso de uso es iniciado por el Administrador mediante la interfaz para la gestión del middleware, luego que se ha verificado que es un usuario del sistema, podrá registrar y configurar equipos que serán interconectados al SOA-Médico, registrar los exámenes médicos que son realizados por dichos equipos. Así como también, registrar y configurar los usuarios, que puedan interactuar con el middleware.
- **Gestionar Resultados Entrantes:** Este caso de uso en caso de ser iniciado por el Equipo Médico mediante un evento que surge del adaptador del mismo, al notificar que un archivo o mensaje requiere ser procesado por el SOA-Médico, este verifica el usuario y de ser válido debe recibir los resultados, guardándolos en el repositorio intermedio.
Por otro lado, cuando este caso de uso es iniciado por el Bioanalista mediante la interfaz para la gestión del middleware, luego que se ha verificado que es un usuario del sistema, podrá validar los resultados de los análisis clínicos alojados en el repositorio intermedio.
- **Gestionar Resultados Salientes:** Este caso de uso es iniciado por el Sistema de Información de Laboratorio (SIL) mediante una solicitud de un servicio al middleware, luego que se ha verificado que es un usuario del sistema, se envían los resultados solicitados.

Diccionario de Actores

Dentro de la fase de exploración se identifican 5 actores fundamentales que se ven afectados antes, durante y después de una implementación tecnológica:

- **Equipo Médico:** Dispositivo físico que mediante un adaptador de software envía información asíncrona con los resultados clínicos a un directorio o a una cola de mensajes.
- **Bioanalista:** Es un usuario especializado en el área del sector salud, que se le asigna la responsabilidad de aceptar o rechazar, en base a su conocimiento y experiencia, si los resultados transferidos al middleware cubren los parámetros deseables de calidad.
- **Administrador:** Es un usuario especializado en actualizar los datos generales para el funcionamiento del middleware. Se encarga de introducir los usuarios, configurar equipos y velar por el mantenimiento.
- **Sistema de Información de Laboratorio:** Software que llevan la gestión del paciente dentro del laboratorio clínico, realizan solicitudes al middleware de los resultados validados contenidos en su repositorio.
- **Usuario:** es un actor que interactúa con el middleware del cual todos los actores anteriores heredan.

Todos los casos de uso identificados en la Figura 15, deben ser implementados a lo largo de desarrollo del middleware SOA-Médico. Su función es establecer un punto de partida para definir la siguiente fase de XP, la planificación de entregas. Además, de ser usados como herramienta útil en la definición de la arquitectura, sirven para verificar, que en las siguientes iteraciones la solución técnica está guiada en base las necesidades requeridas.

Planificación de Entregas

Esta fase consiste en definir un cronograma con las diversas iteraciones de XP que se ejecutan en la presente investigación. Partiendo del hecho que deben implementarse todos los casos de usos para que SOA-Médico atienda mediante sus funcionalidades a todos los requerimientos, se decide tomar los casos de uso como

tareas a ser ejecutadas en las iteraciones. Por ello, se propone en la Figura 17 los casos de usos en un nivel más profundo.

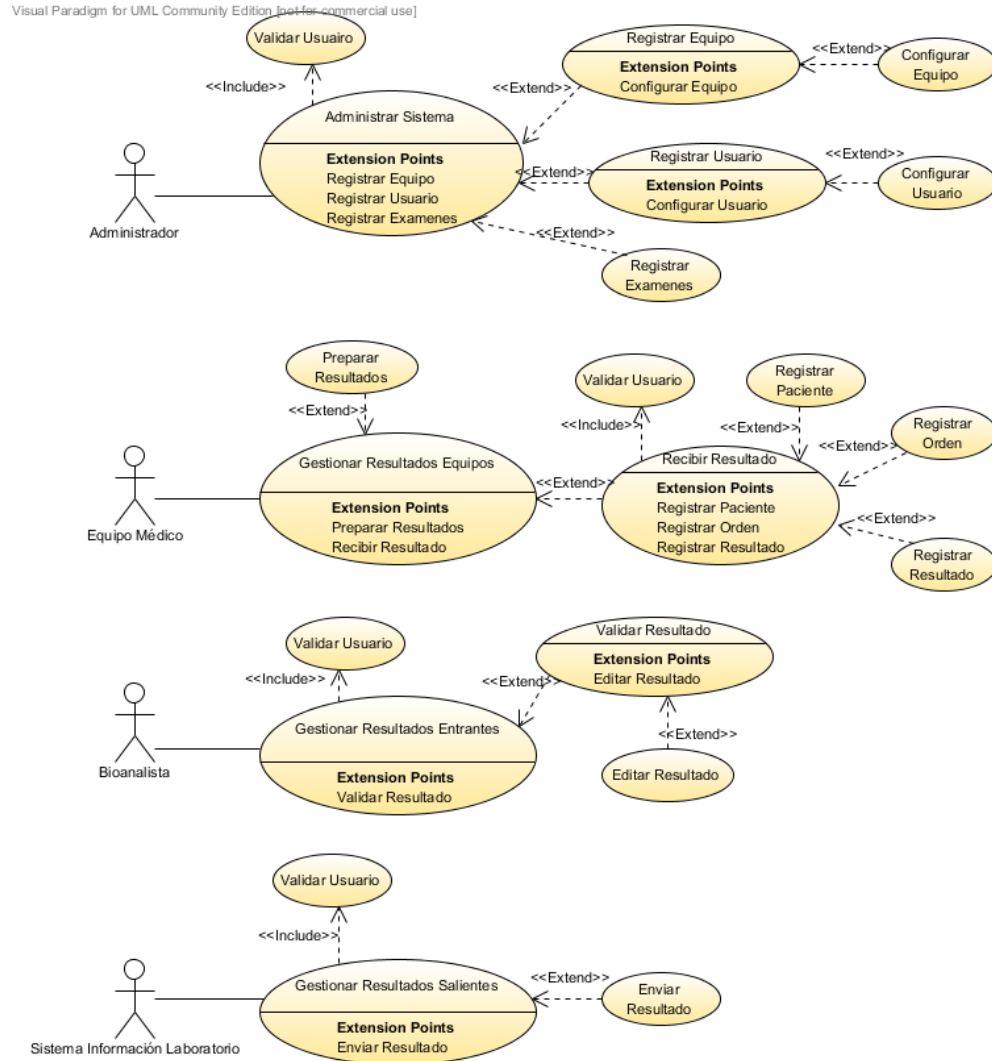


Figura 17. Diagrama de Casos de Uso de SOA-Médico. Fuente: Autor de la Investigación.

De la figura anterior se extraen casos de usos como tareas, se establecen prioridades y se realizan estimaciones de tiempos de entrega, la Tabla 3 muestra la planificación de las iteraciones, los casos de usos se detallan en profundidad en su iteración correspondiente.

Tabla 3: Planificación de Entregas

Iteración	Funcionalidades	Semanas
1	Arquitectura de Software	3
2	Gestión de Información básica SOA-Médico Casos de Usos extendidos de Administrar Sistema: Registrar Equipo, Registrar Examen, Registrar Usuario Casos de Usos extendidos de Recibir Resultados: Registrar Paciente, Registrar Orden, Registrar Resultado.	3
3	Recepción de Resultados SOA-Médico Caso de Uso Gestionar Resultados Entrantes y Recibir Resultado.	5
4	Envío de Resultados SOA-Médico Caso de Uso extendido Gestionar Resultados Salientes y Enviar Resultado.	3
5	<u>Configuración de Sistema</u> Casos de Uso extendido de Registrar Equipo y Usuario: Configurar Equipo, Configurar Usuario.	1
6	<u>Validación de Resultados</u> Casos de Uso extendido de Gestionar Resultados Entrantes: Validar Resultados, Editar Resultado.	2
7	<u>Integración del Sistema</u> Casos de Uso: Administrar Sistema, Gestionar Resultados Entrantes, Gestionar Resultados Salientes, con el caso de uso incluido Validar Usuario.	4

Fuente: El autor de la investigación.

En la planificación de entregas se identifican: (1) número de la iteración, que supone un orden lógico para abordar las tareas, (2) funcionalidades, donde se ubican los casos de uso a ser abordados y (3) tiempo aplicado en semanas, este tiempo se

obtiene de las historias de usuarios que apoyan dichas funcionalidades, según la metodología XP, este tiempo ayuda a estimar la velocidad del proyecto, al usar la velocidad del proyecto contralamos que todas las tareas se puedan desarrollar en el tiempo del que dispone la iteración. Para la presente investigación se atienden las 4 primeras iteraciones.

Iteración 1

La primera iteración se centra en definir la arquitectura, para ello, se propone la presentación de distintas vistas, logrando a grandes rasgos determinar los componentes principales, la conducta de estos y la forma en que los componentes interactúan o se coordinan, a fin de satisfacer las funcionalidades establecidas en la fase de exploración. Se establece el marco de referencia necesario para guiar el desarrollo del middleware, los patrones de diseño, las tecnologías y las herramientas más apropiadas para la construcción de los componentes y sus interacciones.

Arquitectura de Software

A partir de las funcionalidades y los atributos de calidad deseables, basadas en las necesidades inherentes de comunicación entre los equipos médicos y el sistema de Información de Laboratorio, sistemas heterogéneos entre sí, SOA-Médico, aprovecha las ventajas ofrecidas por las Arquitecturas Orientadas a Servicios (SOA) y utiliza este estilo arquitectural como base para definir su arquitectura.

La idea detrás de SOA, es lograr que los recursos de software de una organización estén disponibles y descubiertos en la red como servicios bien definidos, cada servicio puede alcanzar un objetivo predefinido del negocio y ejecutarse en unidades discretas de trabajo. Los servicios son independientes y no obedecen del contexto o estado de otros servicios, ellos trabajan dentro de una arquitectura distribuidas de sistemas.

Un bus de servicio empresarial (Enterprise Service Bus ESB), es una infraestructura que facilita la implementación de la SOA, el ofrece un conjunto de utilidades los cuales pueden ser utilizadas para desarrollar servicios y hace que los servicios interactúen entre sí de forma fiable. Técnicamente un ESB es la columna vertebral que ofrece la conversión de protocolos, la transformación del formato de los mensajes, enrutamiento, permitiendo el flujo de mensajes entre las aplicaciones que estén vinculadas.

Con ambas propuestas, se muestra en la Figura 18, un escenario de ejecución mediante una aproximación arquitectural para el software middleware SOA-Médico.



Figura 18. Aproximación Arquitectural. Fuente: Autor de la Investigación.

Al observar la figura, se puede identificar que existe una capa que es ofrecida a usuarios finales para el consumo de servicios, una capa final donde se encuentra alojados los proveedores de los servicios (equipos médicos, componentes, datos y lógica) y una capa intermedia, el ESB que se encarga gestionar ambas capas mediante transporte, transformación y enrutamiento.

En base a lo antes expuesto y reconociendo el alcance que ha tenido la arquitectura de tres capas (presentación, lógica de negocio y datos) propuesta por Marroquin (2009) para el desarrollo de SIL, se toma como punto de inicio esta arquitectura y adaptándose al presente trabajo de investigación, fue enriquecida, a fin de introducir el concepto de SOA, con las capas de servicio, la capa de modelo de negocio, la capa de integración, y una capa operacional para introducir el concepto de sistemas legados a integrar, representados en este trabajo por los equipo médicos. Se propone la arquitectura en capas para SOA-Médico en la Figura 19.

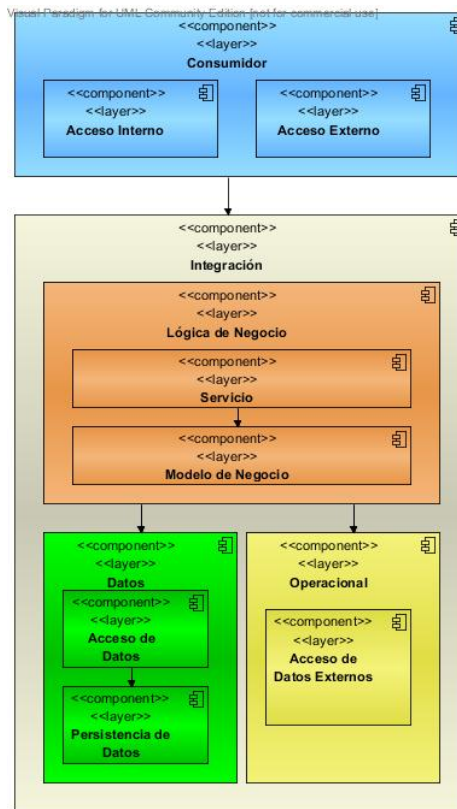


Figura 19. Arquitectura en Capas de SOA-Médico. Fuente: Autor de la Investigación.

La capa Consumidor constituye el exterior del sistema, permite a los actores del sistema enviar y recibir información, está clasificada en dos capas: Acceso Interno, que es una interfaz web para la configuración y gestión del SOA-Médico, que es

accesible por los administradores y bioanalistas a fin de administrar el sistema y validar los resultados clínicos, y Accesos Externos por las solicitudes requeridas por los Sistemas de Información de Laboratorios.

La capa de Lógica de Negocio permite la gestión de las funcionalidades ofrecidas por el sistema a través de una serie de componentes, que gestionan la información de los equipos médicos, acceden al repositorio de datos para la persistencia y validación de los mismos. Se encuentra dividida en dos capas: los servicios que siendo funciones significativas para el negocio serán desplegados en la capa de Servicio, la cual se comporta como una fachada y delega la ejecución de dichas funciones a la capa encargada de implementar la lógica real, el Modelo de Negocio.

La capa de Datos, es donde residen los datos que son recibidos desde los equipos, junto con aquellos necesarios para su administración. Encargada de acceder a los datos, requiere de dos capas internas, la capa de Acceso a Datos donde se gestionan las solicitudes de almacenamientos y la capa de Persistencia de Datos que despliega dichas funciones en un Manejador de Base de Datos.

La capa Operacional delimita el conjunto de equipos médicos con los cuales se integra el sistema, requiere de la capa de Acceso de Datos Externos donde se encuentran los distintos adaptadores a las diferentes interfaces de los equipos médicos y así exponer los datos a la lógica de negocio para que sean persistidos y validados.

La capa de Integración, representada por el bus empresarial de servicio, sirve de mediador entre un consumidor (Sistemas de información de Laboratorio) y un proveedor de servicio (conformado por todos los servicios definidos en la capa de Servicio). Mediante la introducción de un conjunto fiable de capacidades de enrutamiento, transformación y transporte asegura que las normas comunes de

negocio se apliquen en las diferentes capas incluidas Lógica de Negocio, Datos y Operacional.

De conformidad con el principio de arquitectura en capas, debe destacar que cada puede ser accesible desde otra capa, esto permite la independendencia de implementación de cada capa, a fin de lograr reusabilidad, adaptación al cambio y otros. En este sentido, se muestra como es el flujo de información entre las distintas capas en la Figura 20.

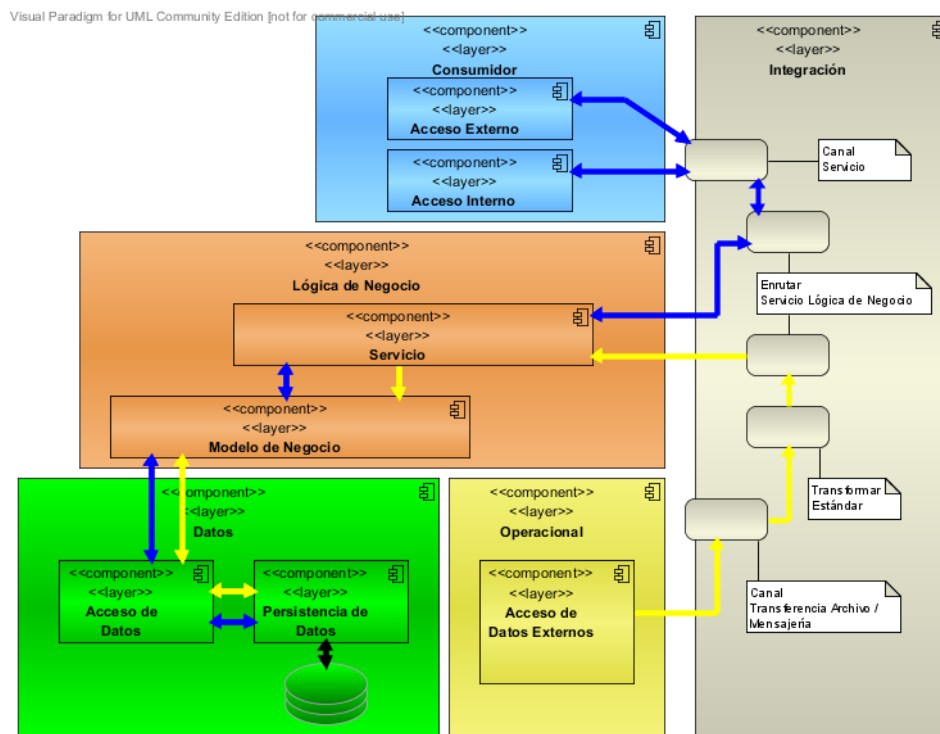


Figura 20. Escenario de Interacción Arquitectura. Fuente: Autor de la Investigación.

En el escenario de interacción entre las capas de la Figura 20, se observa en amarillo, el flujo de los mensajes que nacen con los resultados que son enviados desde los equipos médicos y muere en el repositorio de datos, este flujo comunicación que es de tipo asíncrona, inicia con un acceso de tipo escritura de la capa Operacional sobre el canal (Transferencia Archivo/ Mensajería) de la capa de

Integración, quien a tomar el mando, se encarga de transformar la información basado en un estándar para luego enrutar, invocando el servicio correspondiente dentro de la Lógica de Negocio, quien interactúa con la capa de Datos para la persistencia de los resultados.

Por otro lado, en la Figura 20, en color azul se representa el flujo de los mensajes que de tipo asíncronos, inician con la petición de un servicio desde la capa Consumidor a través del canal (Servicio) provisto por la capa de Integración, esta capa, enruta la petición para el servicio correspondiente en la capa Lógica de Negocio, de ser necesario obtener o actualizar datos se comunica con la capa de Datos y luego se retorna la información por el mismo canal por donde fue solicitada. Se observa que existe flujo de información bidireccional, que a diferencia del flujo en amarillo se permiten accesos de lectura y escrituras, desde la capa Consumidor.

Se mencionan algunas capacidades de la Arquitectura en capas de SOA-Médico:

- La capa de Consumidor no requiere tener en cuenta consideraciones de carácter técnico de la capa Lógica de Negocio, así mismo, acoplar y desacoplar consumidores es totalmente transparente para el sistema.
- Se pueden realizar modificaciones en el Modelo de Negocio sin afectar la capa de Servicio.
- Realizar actualizaciones en la capa de Datos (cambio de manejador de Base de Datos o políticas de persistencia), o incluir/eliminar equipos médicos en la capa Operacional y no afectar la Lógica de Negocio.

En la arquitectura SOA-Médico, cada capa representa un componente, de esta manera al momento de despliegue, pueden separarse los distintos componentes en medios físicos dispersos, la Figura 21 muestra el Diagrama de Despliegue. Se aclara, que los componentes Integración, Lógica de Negocio y Acceso a Datos deben residir juntos.

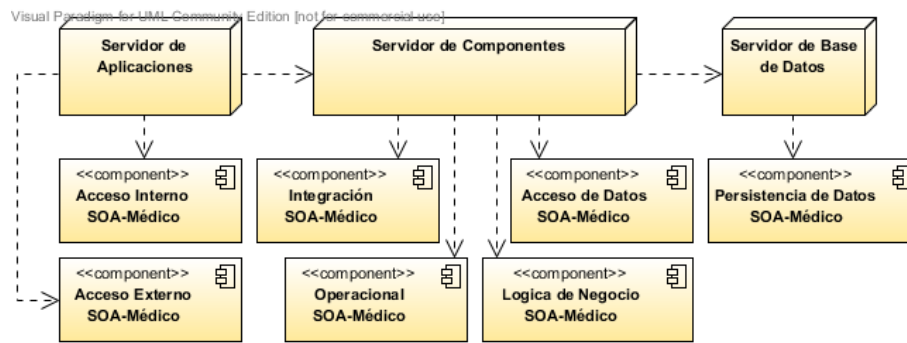


Figura 21. Diagrama de Despliegue SOA-Médico. Fuente: Autor de la Investigación.

Tecnologías, Patrones y Técnicas aplicadas al Desarrollo.

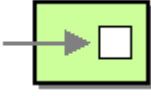


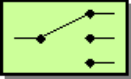
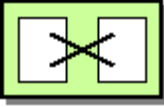
En esta sección, se definen los lineamientos de desarrollo de software que son requeridos por la arquitectura. Se realizan decisiones de gran importancia arquitectónica, como frameworks, patrones de diseño, herramientas de desarrollo, y se concreta la organización de los paquetes para el código fuente.

El primer componente examinado por su alto impacto para la arquitectura de SOA-Médico, fue el de Integración. Seleccionar el bus de servicio empresarial en primera instancia, permite tomar decisiones más concretas y valederas para los sucesivos componentes.

Existen en el mercado, gran cantidad de productos comerciales y de licencia libre, luego de examinar las más reconocidas, se observa dichos productos no que cubren con todas las de características requeridas para un ESB. Por tanto, la selección del bus depende de cuales características son requeridas por la arquitectura y luego seleccionar el software que instanciara dicho componente.

En la Tabla 4 se describen algunos de los patrones que debe cubrir el ESB en la arquitectura de SOA-Médico.

Tabla 4: Patrones requeridos ESB para Arquitectura SOA.

Patrón	Descripción
 <p>Punto Final de Mensaje</p>	Un Punto Final de Mensaje (message endpoint), define una conexión desde un aplicación hasta un canal de mensaje.
 <p>Canal de Mensaje</p>	Un Canal de Mensaje (message channel) permite a las aplicaciones comunicarse unas con otras.
 <p>Adaptador de Canal</p>	Un Adaptador de Canal (channel adapter) define como los sistemas integrados se pueden conectar al ESB para poder recibir o enviar mensajes.
 <p>Ruteo Basado en Contenido</p>	Ruteo Basado en Contenido (Content based router) como su nombre lo indica, los mensajes son enrutados en base al contenido del mensaje.
 <p>Transformador</p>	Transformador (Transformer), es usado cuando el formato de un mensaje necesita ser cambiado antes de ser enviado a un punto o cuando un formato estándar es definido.

Fuente: Hohpe y Woolf (2003).

Dada la necesidad de cubrir dichos patrones, se selecciona Mule ESB Community Edition v2 como software para desplegar el componente de integración, además de tener distribuciones de licencia libre, existe gran documentación en la red. Para MuleSoft Inc., Mule ESB, "es bus de servicio empresarial (ESB) ligero basado en java y una plataforma de integración que permite a los desarrolladores conectar aplicaciones de manera rápida y fácil, permitiendo intercambiar datos entre ellos".

Se menciona algunas de las características del framework que determinaron su elección:

- Sigue el patrón Inversión de control (Inversion of Control IoC) es un método de programación en el que el flujo de ejecución de un programa se invierte respecto a los métodos de programación tradicionales,
- Ofrece una capa de mensajería, sincronía y asíncrona, así como adaptadores a las principales tecnologías de comunicaciones de datos (JMS, JDBC, TCP, UDP, multicast, http, servlet, SMTP, POP3, file, XMPP).
- Ofrece la posibilidad de modelado de servicios así como el registro y descubrimiento de los mismos.
- En cuanto a gestión de servicios, tiene adaptadores a contenedores JBI y a otras herramientas de gestión de servicios, como Xfire/CXF, Axis y Glue que facilitan el descubrimiento o registro de servicios.
- En cuanto a seguridad, soporta transacciones multiprotocolo y ofrece adaptadores para el control de acceso y la encriptación de información.
- Integración con Spring.
- Fácilmente embebible en aplicaciones Java.

Por ser Mule ESB una herramienta de desarrollo ligera basado en java, y para aprovechar sus ventajas al máximo, se selecciona java como lenguaje de desarrollo utilizado para la programación de los diferentes componentes, con sus principales característica de código abierto, de ser multiplataforma y un lenguaje robusto, lo convierten en la mejor elección.

Aprovechando el potencial de java, y su integración con Mule ESB, se selecciona el framework Spring. Al respecto, Spring Source, menciona que es un marco de trabajo de código abierto y para el desarrollo de aplicaciones para la plataforma Java, posee características significantes como:

- El módulo BeanFactory, que implementa el patrón de diseño Factory (Factoría) eliminando la necesidad de crear singletons programáticamente, permitiendo desligar la configuración y especificación de las dependencias de la lógica de programación.
- El módulo Context (Contexto), el cual te provee de herramientas para acceder a los beans de una manera elegante.
- El módulo de Acceso a Datos facilita el trabajo de usar el paquete Mapeo Objeto-Relación, (Object-Relational Mapping ORM) que provee capas de integración para mapeo de objetos con Hibernate, entre otros.
- El módulo Gestión de Transacciones, permite la administración declarativa de las transacciones de los objetos persistentes.

En este sentido, para el componente de datos se utiliza Hibernate, JBoss Inc., menciona que este, es un framework ORM para el mapeo de atributos entre una base de datos relacional, de código abierto, implementando el patrón Domain Store, permite separar la aplicación de la Base de Datos, lo que mejora el Mantenimiento y la escalabilidad de la aplicación. Para el repositorio de datos intermedios, se utiliza PostgreSQL, manejador de base de datos robusto y de licencia libre.

Los diferentes componentes se comunican mediante el uso del patrón Data Acces Object, que es un patrón de diseño que permite utilizar clases simples en java para transferir información.

El componente de Acceso Interno que consta de una interfaz que permite la gestión del middleware, se utiliza Echo2. Al respecto, NextApp, Inc., indica que es un framework para el desarrollo de aplicaciones web ricas, donde las interfaces a nivel de diseño se comporta como una interfaz de Java tipo escritorio librería Swing.

Para las distintas iteraciones, el final de su ejecución está determinado por las pruebas del código generado, para ello se selecciona el framework JUnit, conjunto de

librerías que son utilizadas en programación para hacer pruebas unitarias de aplicaciones Java.

Para el entorno de desarrollo integrado (Integrated Development Environment IDE) se selecciono Eclipse, Eclipse Foundation, sus creadores, mencionan que es un entorno de programación que permite edición, compilación y depuración de código. Además, todos los frameworks utilizados ofrecen soporte de integración para este poderoso IDE de licencia libre. Se exige el uso de herramientas para el control de versiones del código.

El conglomerado de tecnologías y patrones se presentan en la Figura 22, mediante el diagrama de despliegue instanciado, clasificado por componentes.

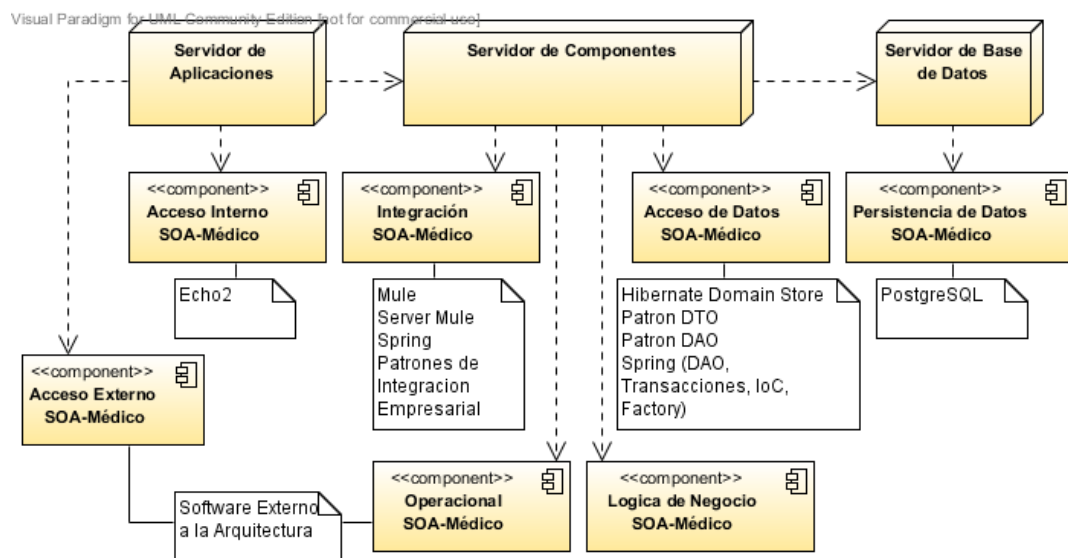


Figura 22. Diagrama de Despliegue con Tecnologías SOA-Médico. Fuente: Autor de la Investigación.

Luego de detallarse las diferentes tecnologías a utilizar, se describe la distribución que debe tenerse para la organización del código generado para la construcción del software.

A continuación se describen la clasificación de los paquetes:

Tabla 5. Descripción de Paquetes Componente de Datos

Paquete	Descripción
com.soamedico.dato.dao.*	Incluye el conjunto de clases para el manejo de la persistencia de los datos.
com.soamedico.dato.dto.*	Incluye el conjunto de clases que representan las entidades del sistema, usadas entre los distintos componentes para transferir información.
com.soamedico.dato.utilidades.*	Se reserva este paquete para las clases que ofrecen utilidades en el contexto de datos.

Fuente: El autor de la investigación.

Tabla 6. Descripción de Paquetes Componente de Integración.

Paquete	Descripción
com.soamedico.esb.recursos.*	Incluye las clases o archivos que pueden servir de recurso útil para el contexto del ESB.
com.soamedico.esb.transformadores.*	Incluye todas aquellas clases necesarias por el ESB para la transformación del contenido de los mensajes que fluyen en él.

Fuente: El autor de la investigación.

Tabla 7. Descripción de Paquetes Componente de Lógica de Negocio

Paquete	Descripción
com.soamedico..negocio.implementacion.*	Contiene el conjunto de clases necesarias para implementar la lógica de negocio.
com.soamedico.negocio.modelo.interfaces.*	Incluye todas aquellas clases que contienen para las interfaces de la lógica de negocio.
com.soamedico.servicios.*	Incluye las clases que se requieren para desplegar los servicios.

Fuente: El autor de la investigación.

Además, se incluyen los paquetes, com.soamedico.recursos.* y com.soamedico.utilidades.*, para clasificar aquellas clases de uso general para el software. También, se ubica una carpeta denominada conf para ubicar todos aquellos archivos de configuración requeridos para el desarrollo, como por ejemplo, la configuración del contexto de spring y de mule.

Iteración 2

Gestión de Información básica SOA-Médico

El conjunto de actividades a ejecutar en esta iteración, están relacionadas con el componente de datos. El objetivo primordial, como se muestra en la Figura 23, es cubrir a nivel de desarrollo las funcionalidades extendidas del caso de uso Administrar Sistema: Registrar Equipo, Registrar Examen, Registrar Usuario y del caso de Recibir Resultados: Registrar Paciente, Registrar Orden, Registrar Resultado.

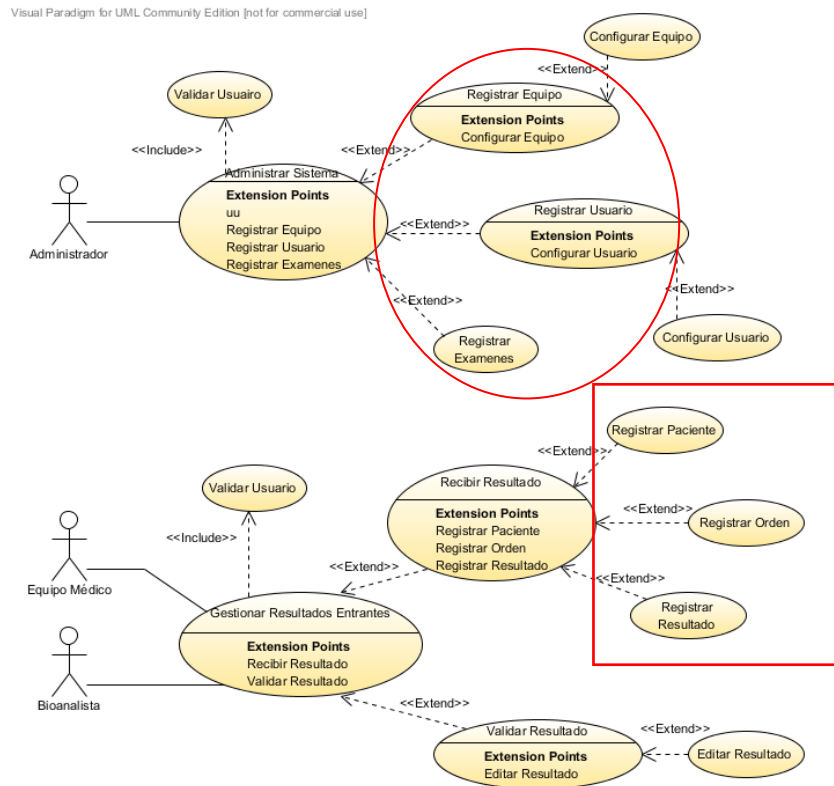


Figura 23. Casos de Uso Gestión para Información básica SOA-Médico. Fuente:
El autor de la investigación.

Cada una de los casos de uso extendidos, Registrar Equipo, Registrar Paciente, Registrar Orden, Registrar Resultado, Registrar Usuario, están enfocados al despliegue de las funcionalidades de incluir, modificar, eliminar y consultar dicha entidad, así como de identificar las relaciones que existen entre ellos y que restricciones deben tenerse en cuenta al momento de ejecutar algunas de esas funciones.

Actividades planificadas en Gestión de Información básica SOA-Médico.

- Determinar las clases, atributos y métodos de las entidades.
- Codificar las distintas clases y configuraciones necesarias para la persistencia bajo los patrones definidos.
- Realizar la pruebas unitarias.

El producto entregable de esta iteración, consiste en los diagramas de clases, el código fuente de la implementación de cada una de las funcionalidades y las pruebas unitarias que dan certeza de su correcto funcionamiento.

Actividades cubiertas en Gestión de Información básica SOA-Médico.

Para la definición de las clases se analizó el estándar ASTM E1394, documento utilizado por los equipo médicos para estandarizar la información a transmitir con los sistemas de información, la idea es considerar un modelo de clases que desde los primeros pasos este acoplado con la información que debe desplegar el middleware en su repositorio de datos. El diagrama de clases se observa en la Figura 24.

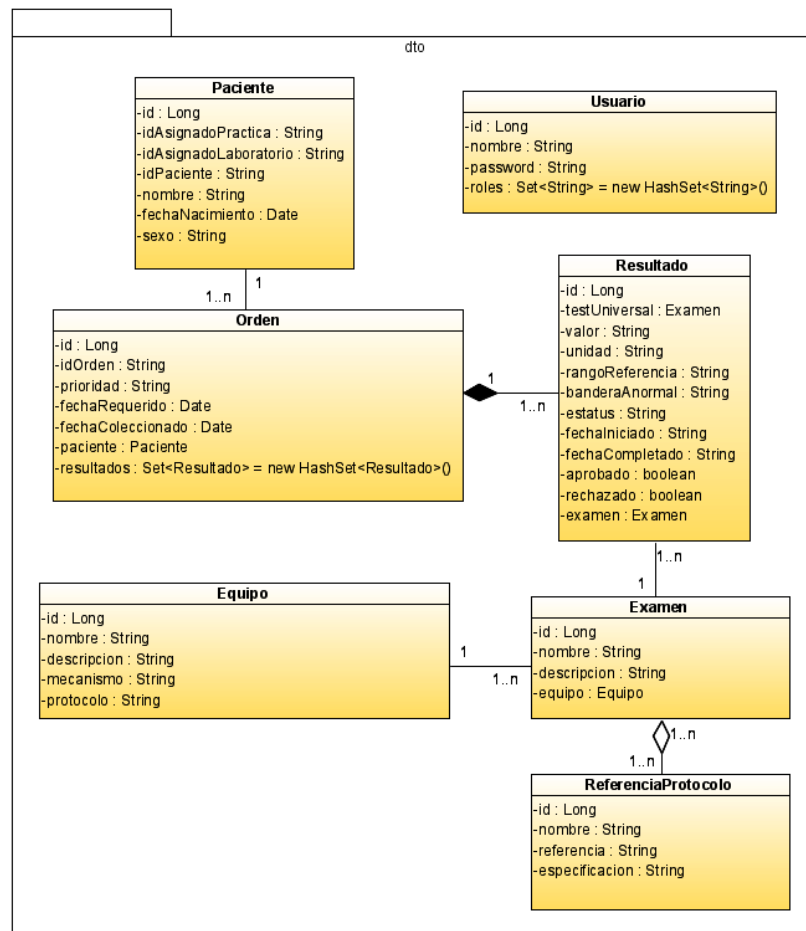


Figura 24. Diagrama de Clases paquete com.soamedico.dato.dto
Fuente: Autor de la Investigación.

En el diagrama de clases se muestra las entidades básicas principales dentro de un Laboratorio Clínico a ser consideradas en SOA-Médico. Las relaciones entre las distintas clases indican cómo se comunican los objetos de esas clases entre sí, lo cual además de permitir ver como navegan los mensajes por las relaciones existentes, permiten determinar la cardinalidad de los mensajes y establecer el comportamiento de los mismos al momento de realizar las operaciones de inserción, modificación y eliminación.

Mediante la definición de las clases anteriores se cubre la implementación del patrón Date Transfer Object (DTO), todas estas clases serán usadas para comunicar los componentes Lógica de Negocio y Datos. El siguiente patrón desarrollado es el Data Access Object (DAO), que provee interfaces para los mecanismos de persistencia, para encapsular los accesos y manipulación de datos.

Realizar el análisis de las relaciones, nos permite identificar restricciones a nivel de los métodos a implementar en los DAOs. Por ejemplo, siguiendo la teoría de las relaciones de composición, que es un tipo de agregación de posesión fuerte, donde el tiempo de vida de la clase que parte del todo (agregada) viene determinado por el tiempo de vida de la clase que representa el todo, por tanto, una eliminación del todo debe generar eliminaciones en cascadas de sus agregadas.

Seguidamente se identifican las clases e implementaciones necesarias para los DAOs, la Figura 25 muestra el diagrama asociado.

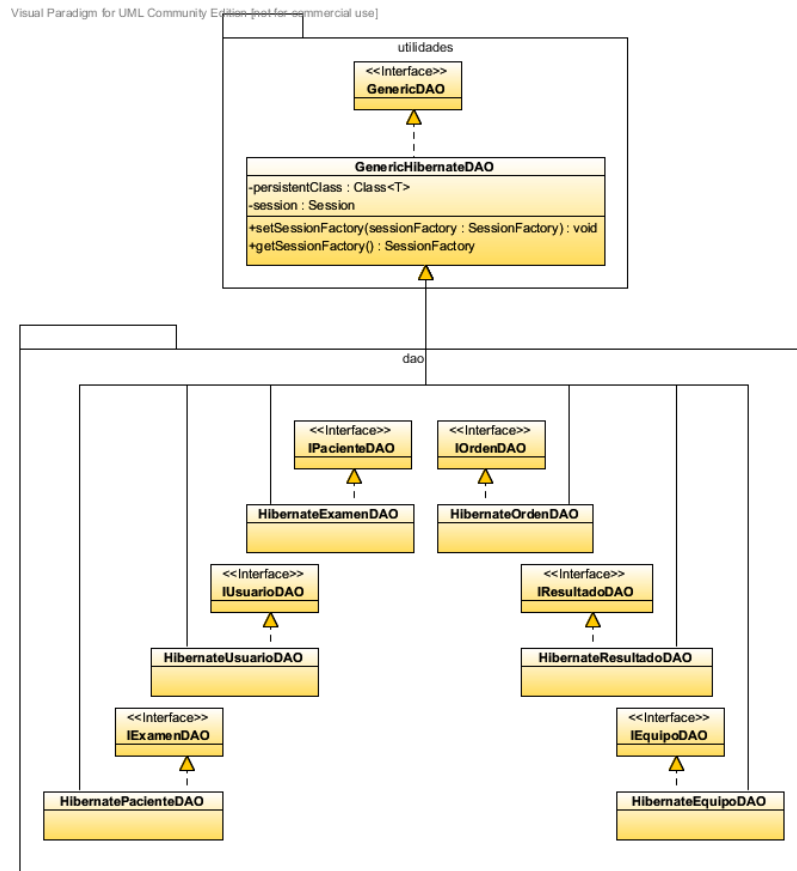


Figura 25. Diagrama de Clases paquete com.soamedico.dato.dao y com.soamedico.dato.utilidades Fuente: Autor de la Investigación.

Por cada entidad se crean dos archivos para implementar el patrón DAO, una interface IEntidadDAO, donde están las operaciones básicas de gestión de datos que deberían ser implementadas por la clase implementadora HibernateEntidadDAO, esta hereda las operaciones en común desde GenericHibernateDAO, pero de ser necesario implementa las específicas para esa entidad, usualmente los métodos de búsqueda. Además, de un archivo Entidad.hbm.xml, donde se declara el mapeo objeto relacional usado por Hibernate.

La sesión de conexión con hibernate y la instancia de la propiedad sessionFactory heredada por cada DAO implementado, se controlan mediante la configuración del archivo contexto-aplicacion-spring.xml de Spring. La inyección de

la sesión de forma declarativa, permite modificar la clase que implementa las operaciones de las interfaces de forma transparente, útil si se desea cambiar el framework ORM Hibernate.

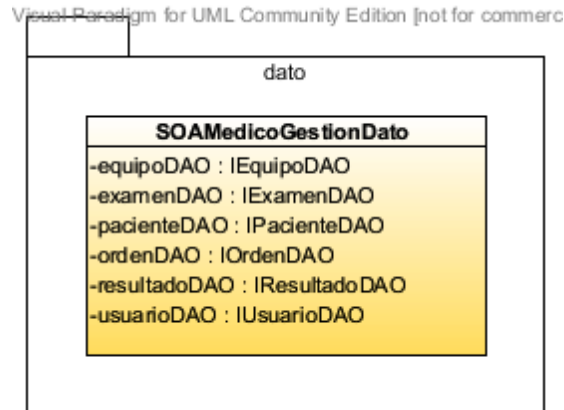


Figura 26. Diagrama de Clases paquete com.soamedico.dato

Fuente: Autor de la Investigación.

El componente Modelo de Negocio requiere acceder a esta capa, para ello se crea la clase SOAMedicoGestionDato, la cual desde ella se podrán ejecutar las operaciones implementadas en los diferentes HibernateEntidadDAO. Esta concentración de operaciones facilita el manejo de las transacciones desde Spring, inyectando el control de las transacciones únicamente a esta clase a través de los atributos de tipo interface.

Iteración 3

Recepción de Resultados SOA-Médico

Una de las funcionalidades a cubrir por el SOA-Médico es facilitar la obtención de los resultados clínicos generados por los equipos, en esta iteración se ejecutan actividades relacionadas con la interacción del componente operacional y el

componente de integración, los objetivos propuestos se observan en función de casos de uso en la Figura 27.

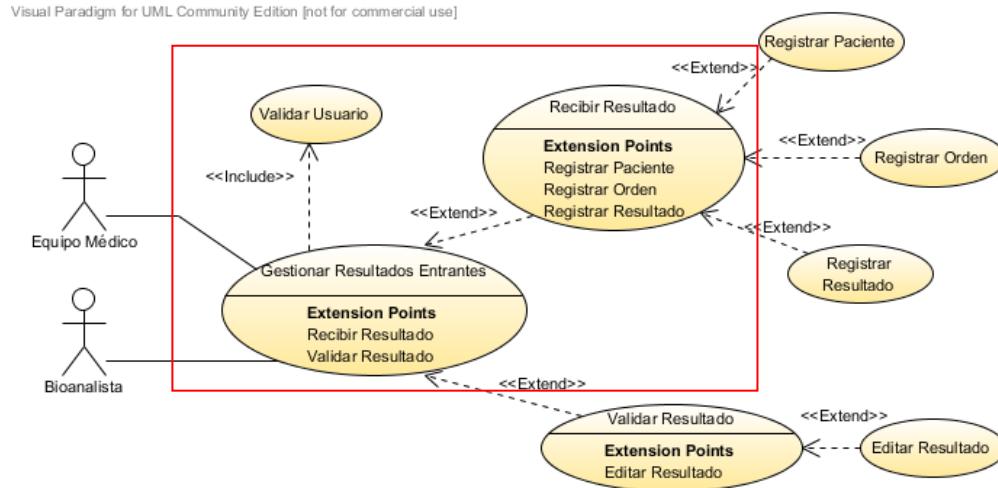


Figura 27. Casos de Uso para Recepción de Resultados SOA-Médico. Fuente: El autor de la investigación.

El caso de uso es iniciado por el equipo médico desde el caso de uso Gestionar Resultados Entrantes, en este caso un adaptador de software que pertenece al componente operacional, gestiona una comunicación directa a la interfaz del equipo, extrae los datos y los envía al componente de integración como un archivo en un directorio o mediante un sistema de mensajería.

Los archivos que son enviados son de tipo XML, pero su descripción varía dependiendo del tipo de mensaje que decida el adaptador del equipo enviar la información. Existen dos tipos:

- ASTM: donde los elementos que definen a el archivo xml son, Tipo (por defecto ASTM), Usuario y Password (para ser validado con el sistema) y el Contenido, información del mensaje en estandar ASTM E1394.

- XML: donde los elementos que definen a el archivo xml son, Tipo (por defecto XML), Usuario y Password (para ser validado con el sistema) y la información de los resultados con una jerarquía de Paciente, Orden y Resultados.
- Se seleccionan únicamente los tipos ASTM y XML, omitiendo el uso de HL7, ya que, por sus características de contemplar el uso de sintaxis XML, de tener una estructura jerárquica y principios de orientación a objetos, su aplicación sería muy similar.

El siguiente paso, consiste en validar la existencia del usuario expresado en el archivo, transformar la información contenida a un formato reconocido por el middleware y persistir los resultados en el repositorio de datos apoyado por el componente Modelo de Negocio.

Actividades planificadas en Recepción de Resultados SOA-Médico.

- Diagramar a nivel de patrones definidos, el bosquejo de como ESB ejecuta la integración.
- Codificar las distintas clases requeridas para transformar los mensajes y los componentes de servicio que soportan la funcionalidad de recibir resultados.
- Implementación de los patrones identificados por el diagrama, para la configuración de Mule.

El producto entregable de esta iteración, consiste en los diagramas de integración, la identificación y codificación de las clases que dan solución a la integración, y las mejores prácticas de Mule para adaptar dicha integración.

Actividades cubiertas en Recepción de Resultados SOA-Médico.

Se muestra en la Figura 28 como los patrones trabajan juntos para lograr una integración mediante transferencia de archivo en un directorio determinado.

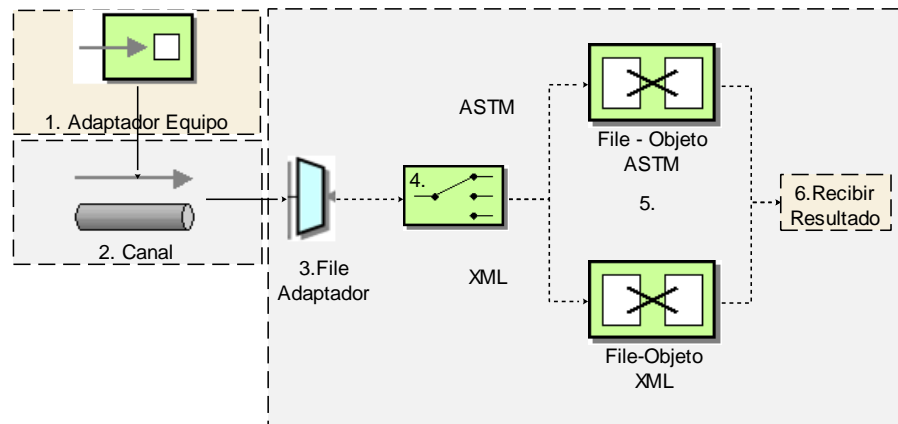


Figura 28. Integración Transferencia de Archivo usando patrones de integración empresarial. Fuente: El autor de la investigación.

A continuación, se describe como el conjunto de patrones trabajan en conjunto para resolver la necesidad de integración mediante transferencia de archivos:

Pasos 1y 2, el adaptador del equipo usa un Mensaje Punto Final(message endpoint) para enviar mensajes a un canal específico, en este caso es un directorio en común que debe ser configurado por el administrador al momento de configurar el equipo.

Paso 3, este directorio es constantemente leído mediante un Adaptador del Canal (channel adapter) de tipo File, usado para gestionar archivos en directorios.

Paso 4, el mensaje es enrutado por el ESB mediante el patron Ruteo Basado en Contenido (content based router), donde examina el tipo de mensaje y en base a su

contenido, ASTM o XML es dirigido al siguiente componente encargado de recibir los resultados.

Paso 5 y 6, el mensaje antes de ser recibido por el componente Recibir Resultado debe ser convertido por el transformador respectivo.

Seleccionar un enrutamiento en base al contenido de los mensajes, lleva consigo las siguientes implicaciones, solo un consumidor recibe cada mensaje, facilita al control central de la data y al mantenimiento. De ser necesario incluir un nuevo formato para la recepción de los mensajes desde los equipos, es necesario hacerle saber al ESB la existencia de ese nuevo formato mediante el archivo de configuración.

Dentro de la dinámica observada bajo los patrones de integración, se requieren los siguientes archivos:

- El esquema para los archivos XML que vienen desde los equipos médicos.
- Conjunto de librerías para el manejo de XML en java.
- Las clases necesarias para convertir los dos tipos de formatos de resultados XML y ASTM a los correspondientes DTO.
- La implementación del componente Recibir Resultado dentro del componente Lógica de Negocio, este se encarga de validar al usuario contenido en el archivo, de ser válido, se reciben los DTO transformados y se transfieren al componente de datos para que maneje la persistencia en el manejador de base de datos.

Para el esquema de los archivos XML se definen los archivos xmlSchema_Astm.xsd y xmlSchema_Xml.xsd, se utiliza XML Schema para la definición del contenido, en las figuras 29 y 30 se muestran ambos archivos.

```

▼<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://xml.soamedico.org/schema/schema"
elementFormDefault="qualified">
  <xsd:element name="Resultado" type="tns:TipoResultado"/>
  ▼<xsd:complexType name="TipoResultado">
    ▼<xsd:sequence>
      <xsd:element name="Formato" type="tns:TipoFormato"/>
      <xsd:element name="Usuario" type="tns:TipoUsuario"/>
      <xsd:element name="Contenido" type="tns:TipoContenido"/>
    </xsd:sequence>
  </xsd:complexType>
  ▼<xsd:complexType name="TipoFormato">
    <xsd:attribute name="tipo" type="xsd:string" use="required"/>
  </xsd:complexType>
  ▼<xsd:complexType name="TipoUsuario">
    ▼<xsd:sequence>
      <xsd:element name="nombre" type="xsd:string"/>
      <xsd:element name="password" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
  ▼<xsd:complexType name="TipoContenido">
    ▼<xsd:sequence>
      <xsd:element name="mensaje" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>

```

Figura 29. Esquema del contenido para xmlSchema_Astm.xsd. Fuente: El autor de la investigación.

```

▼<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://xml.soamedico.org/schema/schema"
elementFormDefault="qualified">
  <xsd:element name="Resultado" type="tns:TipoResultado"/>
  ▼<xsd:complexType name="TipoResultado">
    ▼<xsd:sequence>
      <xsd:element name="Formato" type="tns:TipoFormato"/>
      <xsd:element name="Usuario" type="tns:TipoUsuario"/>
      <xsd:element name="Contenido" type="tns:TipoContenido"/>
    </xsd:sequence>
  </xsd:complexType>
  ▼<xsd:complexType name="TipoFormato">
    <xsd:attribute name="tipo" type="xsd:string" use="required"/>
  </xsd:complexType>
  ▼<xsd:complexType name="TipoUsuario">
    ▼<xsd:sequence>
      <xsd:element name="nombre" type="xsd:string"/>
      <xsd:element name="password" type="xsd:string"/>
    </xsd:sequence>
  </xsd:complexType>
  ▼<xsd:complexType name="TipoContenido">
    ▼<xsd:sequence>
      <xsd:element name="orden" type="tns:TipoOrden"/>
    </xsd:sequence>
  </xsd:complexType>
  ▼<xsd:complexType name="tns:TipoOrden">
    ▼<xsd:sequence>
      <xsd:element name="prioridad" type="xsd:string"/>
      <xsd:element name="fecharequerido" type="xsd:date"/>
      <xsd:element name="fechacoleccionado" type="xsd:date"/>
      <xsd:element name="Resultados" type="xsd:TipoResultados"/>
      <xsd:element name="Paciente" type="xsd:TipoPaciente"/>
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>

```

Figura 30. Esquema del contenido para xmlSchema_Xml.xsd. Fuente: El autor de la investigación.

En la Figura 31, se muestra la distribución en los paquetes del código generado que permitirá la recepción de los resultados desde los equipos médicos y su transformación para su posterior persistencia en la base de datos intermedia.

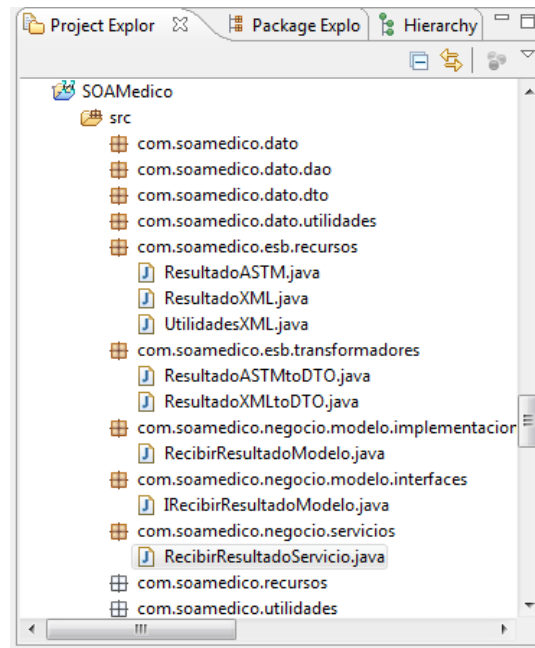


Figura 31. Código obtenido Iteración 3. Fuente: Autor de la Investigación.

Realizar una aproximación del diagrama de integración aplicado a Mule, debe revisarse las distintas documentaciones que ofrece la comunidad de desarrolladores en esta herramienta, las siguientes imágenes Figura 32 y Figura 33 se incluyen para demostrar que puede ser atendido cada uno de los patrones mencionados, bajo distintas sintaxis ofrecidas por dicha herramienta. Los números ubicados en cada una de ellas, atienden a cierto punto del diagrama de la Figura 28.

```

<service name="FileReader">
  <inbound>
    <mule-file:inbound-endpoint address="file:///C:/equipo/resultado/file/in"
      moveToDirectory="C:/equipo/resultado/file/out" moveToPattern="#[DATE]-#[ORIGINALNAME]">
    3 <mule-file:filename-wildcard-filter
      pattern="*.xml"/>
    </mule-file:inbound-endpoint>
  </inbound>
  <outbound>
    <filtering-router>
      <vm:outbound-endpoint path="recibirResultadoAstm"/>
    4 <mule-xml:xpath-filter pattern="/Formato/@Tipo"
      expectedValue="ASTM"/>
    </filtering-router>
    <filtering-router>
      <vm:outbound-endpoint path="recibirResultadoXml"/>
    4 <mule-xml:xpath-filter pattern="/Formato/@Tipo"
      expectedValue="XML"/>
    </filtering-router>
    <forwarding-catch-all-strategy>
      <vm:outbound-endpoint path="userErrorHandler"/>
    </forwarding-catch-all-strategy>
  </outbound>
</service>

```

Figura 32. Configuración Mule Integración Transferencia de Archivo (FileReader). Fuente: El autor de la investigación.

```

<service name="RecibirResultadoAstmUMO">
  <inbound>
    <vm:inbound-endpoint path="recibirResultadoAstm"/>
    5 <jms:jmsmessage-to-object-transformer/>
    <transformer ref="xmlToObjAstm"/>
  </inbound>

  <component>
    6 <spring-object bean="recibirResultadoEquipo"/>
  </component>
</service>

<service name="RecibirResultadoXmlUMO">
  <inbound>
    <vm:inbound-endpoint path="recibirResultadoXml"/>
    5 <jms:jmsmessage-to-object-transformer/>
    <transformer ref="xmlToObjXml"/>
  </inbound>

  <component>
    6 <spring-object bean="recibirResultadoEquipo"/>
  </component>
</service>

```

Figura 33. Configuración Mule Integración (RecibirResultadoEquipo). Fuente: El autor de la investigación.

Para la siguiente integración, basados en sistema de mensajería, se propone el uso de Java Message Service (JMS) integrado con un servidor de mensajería ActiveMQ como canal para recibir mensajes desde los equipos. Para la recepción de

mensajes de los equipos por mensajería se establece el patrón punto a punto (point-to-point), el cual demuestra un buen uso cuando es justamente un solo consumidor interesado en los mensajes, en este caso Mule es el único consumidor.

En la siguiente Figura 34 se presenta la integración del equipo médico al middleware con uso de mensajería, se puede notar como Mule puede cambiar protocolos de comunicación de una manera muy sencilla y reutilizar los servicios que ya estén implementados.

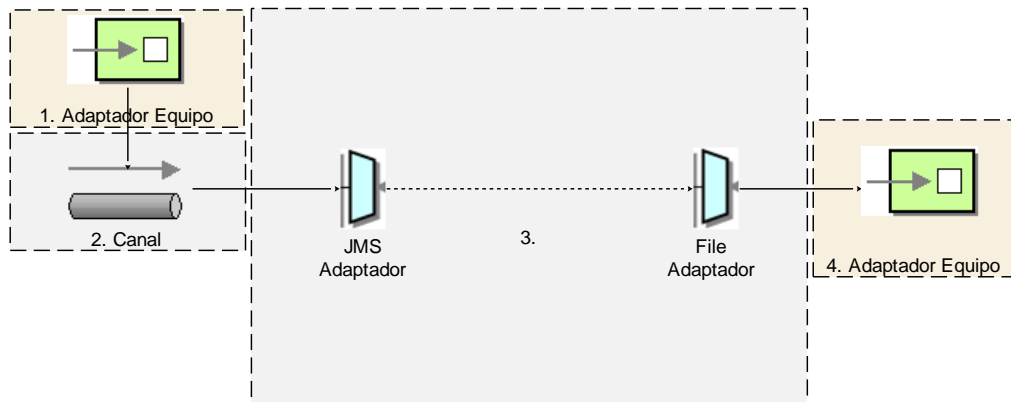


Figura 34. Integración Sistema Mensajería usando patrones de integración empresarial. Fuente: El autor de la investigación.

A continuación, se describe como el conjunto de patrones que trabajan en conjunto para resolver la necesidad de integración mediante sistema de mensajería:

Pasos 1y 2, el adaptador del equipo usa un Mensaje Punto Final(message endpoint) para enviar mensajes a un canal específico, en este caso es una cola de mensajes (queue) con un Adaptador del Canal (channel adapter) JMS.

Paso 3, cuando un mensaje es recibido en la cola, se transfiere al Adaptador File que lo envía al Mensaje de Punto Final (message endpoint) al directorio configurado en la integración basada en transferencia de archivo, activandose dicho mecanismo de integración.

La Figura 35 muestra una aproximación a la solución de la integración mediante sistema de mensajería aplicado.

```
<service name="enrutarResultadoJmsAFile">
  <inbound>
    <mule-jms:outbound-endpoint queue="resultados.in"/>
  </inbound>

  <outbound>
    <pass-through-router>
      <file:outbound-endpoint path="file:///C:/equipo/resultado/file/in">
    </pass-through-router>
  </outbound>
</service>
```

Figura 35. Configuración Mule Integración (EnrutarResultadoJmsAFile).

Fuente: El autor de la investigación.

En la Figura 36 se muestra el flujo de información desde que se obtiene el resultado desde los equipos médicos hasta el repositorio de datos. Mule ESB, es el encargado de recibir mensajes mediante los canales FILE/JMS y transformar la información mediante JAXB y los XML Schema definidos a objetos JAVA y enrutar al componente encargado para la Recepción de Resultados.

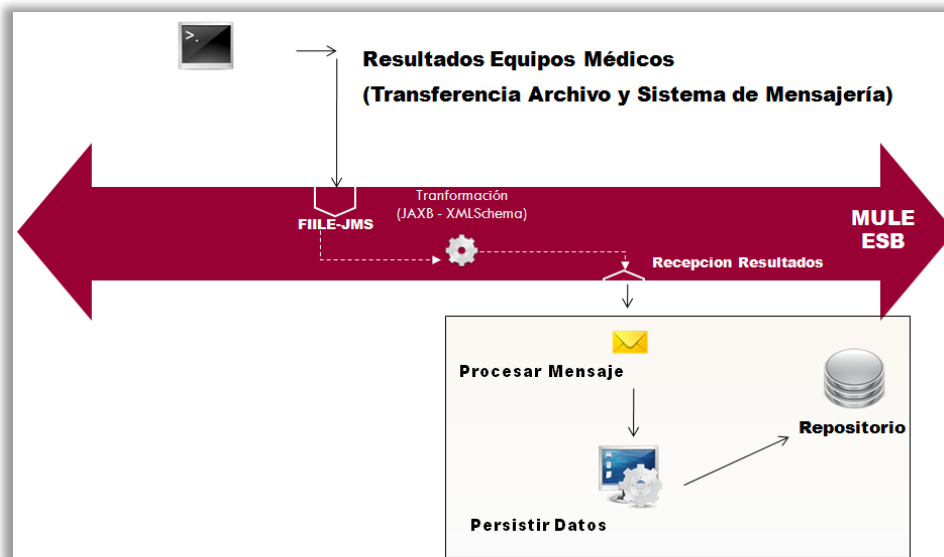


Figura 36. Esquema de Funcionamiento SOA-Médico Recepción de Resultados.

Fuente: El autor de la investigación.

Iteración 4

Envío de Resultados SOA-Médico

El Sistema de Información del Laboratorio (SIL), requiere del conjunto de resultados que están alojados en el middleware, esta iteración se encarga de cubrir las actividades necesarias para el despliegue de dicha información a los SIL.

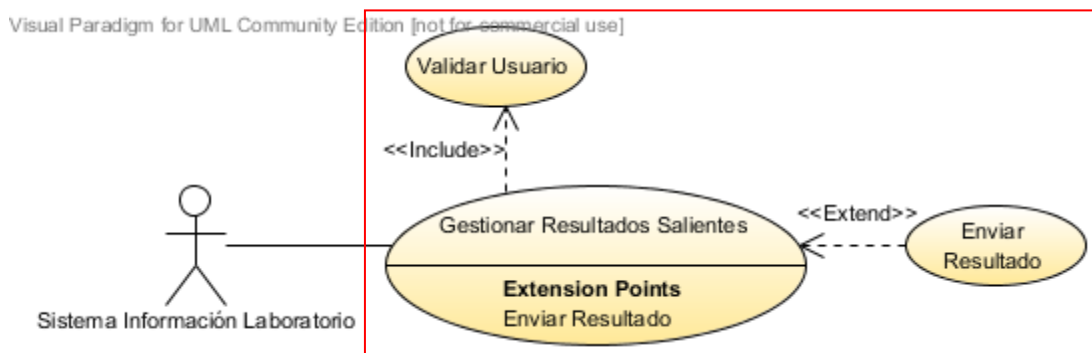


Figura 37. Casos de Uso Envío de Resultados SOA-Médico. Fuente: El autor de la investigación.

El caso de uso es iniciado por el SIL, este realiza una solicitud de un servicio donde se incluyen los datos del solicitante para su validación y el método del servicio deseado, el middleware responde con la información solicitada en caso de ser un usuario valido del sistema.

Actividades planificadas en Envío de Resultados SOA-Médico.

- Diagramar a nivel de patrones definidos, el bosquejo de como ESB ejecuta la integración.
- Identificar las distintas clases o archivos requeridos para transformar los mensajes y los componentes de servicio que soportan la funcionalidad de recibir resultados.

- Estudiar algunas implementaciones de los patrones identificados por el diagrama, para la configuración de Mule.

El producto entregable de esta iteración, consiste en los diagramas de integración, la identificación y codificación de las clases que dan solución a la integración, y las mejores prácticas de Mule para adaptar dicha integración.

Actividades cubiertas en Envío de Resultados SOA-Médico.

Se muestra en la Figura 38 como los patrones de integración empresarial trabajan en conjunto para lograr una integración mediante servicios web.

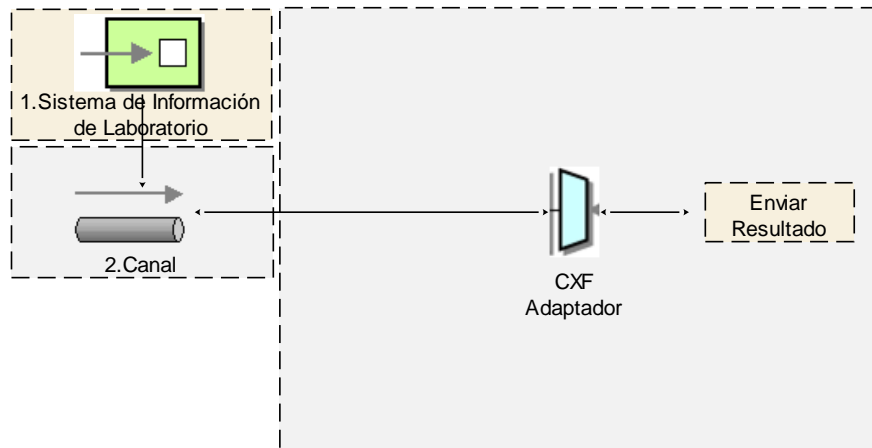


Figura 38. Integración Servicio Web usando patrones de integración empresarial. Fuente: El autor de la investigación.

A continuación, se describe como el conjunto de patrones que trabajan en conjunto para resolver la necesidad de integración mediante sistema de mensajería:

Pasos 1y 2, el adaptador del equipo usa un Mensaje Punto Final(message endpoint) para enviar mensajes a un canal específico, en este caso es una cola de mensajes (queue) con un Adaptador del Canal (channel adapter) JMS.

El Adaptador del Canal CXF es un framework de servicios de software sobre de Apache, ayuda a construir y desarrollar servicios utilizando JAX-WS que es el proyecto de referencia para establecer estandar java para servicios web, como API de programación. Estos servicios pueden hablar una gran variedad de protocolos como SOAP, XML/HTTP, HTTP RESTful, o CORBA, y pueden trabajar sobre transportes como HTTP, JMS o JBI.

Dentro de la dinámica observada bajo los patrones de integración, se requieren los siguientes archivos:

- El esquema para los archivos XML utilizados para intercambiar entre los SIL y el middleware.
- Conjunto de librerías para el manejo de XML en java.
- La implementación del componente Enviar Resultado dentro del componente Lógica de Negocio, este se encarga de validar al usuario contenido en la solicitud, de ser válido, se envían los datos solicitados.

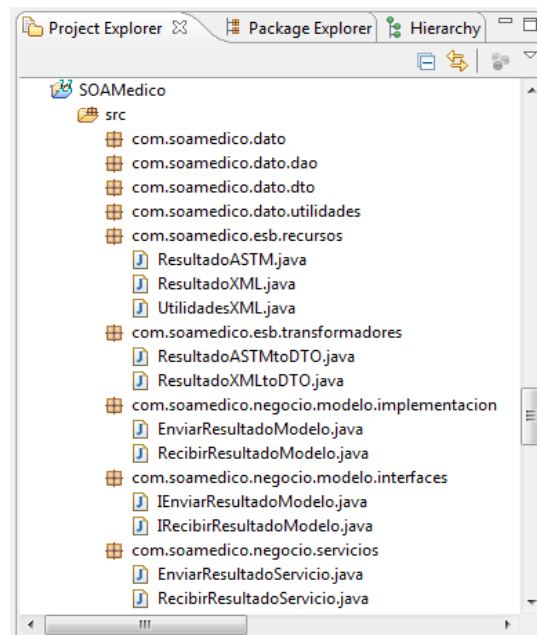


Figura 39. Código obtenido Iteración 3 Fuente: Autor de la Investigación.

En la Figura 40, se muestra como un servicio web es desplegado facilmente por Mule, a partir clases en java.

```

<service name="EnviarResultadoService">
  <inbound>
    <cxfl:inbound-endpoint address="http://localhost:65082/services/EnviarResultadoService"/>
  </inbound>
  <component>
    <spring-object bean="enviarResultadoService"/>
  </component>
</service>

```

Figura 40. Configuración Mule Integración (Enviar Resultado Servicio). Fuente:
El autor de la investigación.

Tabla 8. Definición de Servicio

Nombre	ObtenerResultadosClinicos
Grupo de Servicio	EnviarResultadoServicio
Uso	Obtener resultados clínicos del repositorio de datos de SOA-Médico.
Entrada	resultadoOrdenProcesada
Salida	Coleccion de resultadoOrdenProcesada
Precondición	Los resultadoOrdenProcesada deben tener usuario valido, los resultados deben estar aprobados.
Postcondición	N/A
Acciones	<ol style="list-style-type: none"> 1. Verificar usuario solicitante. 2. De ser válido, determinar los resultados de la orden solicitada 3. Obtener los resultados del repositorio de datos del middleware 4. Retornar la información.
Clases de Negocio	EnviarResultadoServicio

Fuente: El autor de la investigación.

El conjunto de actividades que no se cubren en esta iteración se incluyen en las iteraciones siguientes, haciendo una replanificación en las entregas, característica de la metodología aplicada XP.

Resultados Obtenidos de la Propuesta del Estudio

Luego de culminadas las distintas fases de la metodología XP propuesta en la presente investigación, entre ellas, la exploración donde se identifican requerimientos funcionales y no funcionales, la planificación de entregas determinando el orden de las cinco iteraciones, donde en la iteración inicial se realiza una aproximación a la arquitectura mediante su diseño y las cuatro iteraciones restantes atienden al desarrollo del modelo arquitectural.

Se logra ver, que el diseño del modelo arquitectural permitió hacer una abstracción de alto nivel para la selección de componentes y conectores de SOA-Medico, útil para la selección de las tecnologías, frameworks y patrones requeridos, los cuales, al ser probados en la fase de desarrollo de la arquitectura en los distintos componentes por separado, aseguran que la combinación total en el desarrollo integrado del middleware SOA-Medico, cubrirá con la totalidad de los requerimientos determinados en la fase de exploración, y por tanto se convertirá en una herramienta que aplicada en los entornos de los sistemas de información de laboratorios clínicos y los equipos médicos, logra la interoperabilidad de manera que se obtiene la reducción de tiempos de transcripción y mejoras notables en los errores ocasionados por la transferencia manual de resultados.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

En atención a los objetivos planteados en la presente investigación y a los resultados obtenidos de reconocer los requerimientos funcionales y no funcionales y desplegarlos en un arquitectura, objetos de estudios, se presenta un cuerpo de conclusiones que resume los hallazgos más relevantes del estudio. Igualmente, se establecen algunas sugerencias como alternativas posibles para contribuir con el Desarrollo de una Arquitectura Orientada a Servicios que permita la interoperabilidad de equipos médicos y el Sistema de Información de Laboratorio como parte de la arquitectura tecnológica empresarial del laboratorio clínico Mascia, sede Central, Barquisimeto, Estado Lara.

Con respecto a identificar los requerimientos funcionales y no funcionales que permiten una integración de equipos médicos y el Sistema de Información de Laboratorio (SIL), se puede inferir que su determinación permitió establecer las primeras bases para establecer la arquitectura orientada a servicios, las cuales fueron reforzadas en las diversas iteraciones mediante su diseño y desarrollo.

En relación a describir los componentes y mecanismos de integración incluidos en el diseño del modelo arquitectural para la interoperabilidad de equipos médicos y el Sistema de Información de Laboratorio (SIL), se señala que ésta no es solamente una tecnología, sino una arquitectura que trata de estructurar todos sus componentes de software en la infraestructura tecnológica de una organización para responder de forma ágil y flexible a las demandas del mercado actual en el sector salud, específicamente en los laboratorios clínicos.

Por ello, con el desarrollo de la arquitectura orientada a servicios se logra determinar que la aplicación combinada de tecnologías, metodológica de desarrollo, patrones de diseños, estilos arquitectónicos, buenas prácticas de desarrollo de software, entre otros, permiten resolver los problemas de conectividad entre los equipos médicos y los Sistema de Información de Laboratorio, obteniendo:

- La selección del lenguaje de programación Java para el desarrollo de los componentes incluidos en las diferentes capas de la arquitectura, caracterizado por ser de código abierto, orientado a objeto, robusto y de alta demanda en el sector de desarrollo de software, permitió contar con un conjunto de librerías de licencias libres para el procesamiento de archivos XML, formato que por ser un estándar, fue utilizado para recibir mensajes desde los equipos médicos.
- El uso de Mule ESB como herramienta para desarrollar el bus de servicio empresarial de la capa de integración, demostró ser un componente de software capaz de adaptar canales para entrada de datos, transformación y enrutamiento de mensajes, permitiendo controlar el flujo de los datos provenientes desde los equipos, decodificar el contenido de los mismos en objetos para su persistencia en un repositorio de datos. Caracterizado por un estilo de programación declarativa, asegura que la inclusión de nuevos equipos a la arquitectura se realiza de manera fácil, permitiendo reutilizar los componentes existentes.
- La combinación de Spring y Hibernate facilitó la instancia de los objetos de manera declarativa mediante la inversión de control y el mapeo de los datos objeto a relacional para la persistencia de los mismos.

En este sentido, los hallazgos logrados luego del desarrollo de la arquitectura orientada a servicios, se resumen en: 1) interoperabilidad, porque hizo posible

exponer los resultados clínicos obtenidos desde los equipos como servicios, implementar flujos de trabajo para el intercambio de información entre los mismos, 2) flexibilidad, al dividir la arquitectura en varias capas permite encapsular los componentes permitiendo alto mantenimiento y dinamismo, 3) reusabilidad porque está basada en servicios que por definición son reutilizables y 4) rentabilidad porque ofrece a la organización nuevas funcionalidades que permiten mejorar los procesos, a costos incomparables.

La implementación de la presente propuesta de estudio, constituye la base que garantizaría la agilidad de los procesos de un laboratorio clínico, un prerequisite fundamental para alcanzar el éxito en el actual mercado competitivo, pues esta agilidad tiene la capacidad de añadir, modificar y optimizar fácilmente los procesos de negocio mediante el aprovechamiento de las sinergias de servicios y propiciar un ambiente colaborativo que reditúa en beneficios organizacionales y en los implicados del sector salud como información fiable con mejores tiempos de respuestas para la transcripción y entrega de resultados clínicos.

Finalmente, se consigue materializar el papel de las tecnologías de la información como ingrediente fundamental de los nuevos procesos de negocio que dan forma a una empresa ágil y con capacidad de adaptación. En este sentido, la arquitectura posee el potencial para mejorar la eficiencia, la adaptabilidad y la agilidad operativas tanto a nivel de procesos negocio como a nivel del departamento de tecnología.

Recomendaciones

Ante los planteamientos expuestos en esta investigación se hace imperioso considerar, por parte de la organización las siguientes recomendaciones: pensando que del debido análisis se pueda optimizar las relaciones laborales, lograr la identificación plena de los empleados con su organización y mejorar los procesos que

llevan a lograr los objetivos planteados.

- Abordar la implantación de una Arquitectura Orientada a Servicios por fases y de forma iterativa para ajustar la nueva tecnología, organización y procedimientos de trabajo.
- Aplicar el planteamiento de la arquitectura Orientada a Servicios al propio Laboratorio Clínico, pues a medida que aumenta el alineamiento del negocio con la tecnología, es necesario que ésta desarrolle capacidades para aplicar el modelo de arquitectura Orientada a Servicios. De esta manera el área de negocio, además de definir las aplicaciones orientadas a procesos, deberá ser consciente de la necesidad de reutilizar otros procesos y servicios existentes, así como que los procesos y servicios que definan en un futuro puedan ser también piezas reutilizables.
- Llevar a cabo el proyecto de automatización en todos los Laboratorios Clínicos de la Zona, respetando la lista de prioridades en el proceso de automatización.

Como parte de la implementación de mecanismos y herramientas de transmisión del conocimiento dentro de una Arquitectura Orientada a Servicios acorde a estos medios de comunicación, es necesario incorporar investigaciones multidisciplinarias para medir el aprendizaje por los diferentes actores y participantes de la organización, estableciendo parámetros de medición por grupos de trabajo, tipos de receptor, características propias organizacionales y por el volumen de información y participantes dentro del repositorio de información y usuarios de dicha arquitectura.

REFERENCIAS BIBLIOGRÁFICAS

- Aguirre, A. y Otros (2002). **Capítulo I: Conceptos y objetivos del LTR. Metodología. En: el Laboratorio a Tiempo Real (L.T.R.)**. Publicación de Fundación Signo y Roche Diagnostics S. L. Recuperado el 13 de marzo de 2010 de URL: <http://www.fundacionsigno.com/archivos/20101008140320.pdf>
- Alves, P., Foti, P. y Scalone, M. (2006). **Proyecto Batuta - Generador de Aplicaciones Orquestadoras. s.l.** : Facultad de Ingeniería - Universidad de la República, 2006. Versión 1.2.1.
- Arias, F. (2006). **El Proyecto de Investigación, Introducción a la Metodología Científica**. (5ª Edición). Caracas, Venezuela: Epísteme.
- Arias, J. (2005). SOA: **¿Sólo un estilo de arquitectura más o una burbuja en evolución?** Recuperado el 01 de abril de 2010 de URL: http://www.acis.org.co/fileadmin/Revista_111/columnistainvitado.pdf
- ASTM international (2002). **ASTM E1394-97 Standard Specification for Transferring Information Between Clinical Instruments and Computer Systems**. <http://www.astm.org>
- Barreiro, J. (2002). **Introducción. En: el Laboratorio a Tiempo Real (L.T.R.)**. Publicación de Fundación Signo y Roche Diagnostics S. L. Recuperado el 13 de marzo de 2010 de URL: <http://www.fundacionsigno.com/archivos/20101008140320.pdf>
- Barry, J.; Barry, D. 2003. **La Salud en la Sociedad de la Información**. Recuperado el 02 de febrero de 2010 de URL: http://www.frcu.utn.edu.ar/deptos/depto_3/32JAIIO/sis/SIS_10.pdf
- Bernstein, P. (1996). **Middleware: A Model for Distributed System Services**. Washington State University
- Bieberstein, N.; Bose, S.; Fiammante, M.; Jones, K.; Shah, R.. (2005) **Service-Oriented Architecture Compass: Business Value, Planning, and Enterprise Roadmap**. IBM Press.
- Bologna, J. y Walsh, A. M. (1997). **The Accountant's Handbook of Information Technology**. Editorial John Wiley and Sons.

- Buschmann, F., y otros. 1996. *Pattern Oriented Software Architecture: A System of Patterns*. Inglaterra : John Wiley & Sons, 1996. 0471958697.
- Campoli, M. (2003). **Sistemas de Información en el Sector Salud: “Utopía o realidad”**. Tesis Publicada. Fundación Universitaria Dr. René Favaloro.
- Carrascoso, Y. y Chaviano, V. (2008). **Propuesta de Arquitectura Orientada a Servicios para el Módulo de Inventario del ERP Cubano**. Investigación Concluida. Universidad de las Ciencias Informáticas. Ciudad de la Habana, Cuba.
- Cedeño, T. (2008). **Análisis de los Tiempos de Respuesta del Laboratorio Clínico para el Servicio de Urgencias del Hospital San Rafael de Alajuela durante los meses de mayo a septiembre del 2007**. Recuperado el 27 de enero de 2010 de URL:http://biblioteca.icap.ac.cr/BLIVI/TESIS/2008/Cedeño_Cascante_Tatiana_%202008_SA.pdf
- Clements, P. (1996). **A Survey of Architecture Description Languages**.
- Constitución de la República Bolivariana de Venezuela. (1999). **Gaceta oficial de la República Bolivariana de Venezuela, Nro. 36.860**, (Extraordinario) de fecha 30 de diciembre de 1999, Caracas, Venezuela.
- Fabregas, J. y Bauza, J. (1991). **Administración de Proyectos**. Venezuela: Editorial Miro C.A.
- Fowler, Martin, y otros. 2002. *Patterns of Enterprise Application Architecture*. s.l. : Addison Wesley, 2002. 0-321-12742-0.
- Frantz, R. (2008). **Integración de Aplicaciones. Un lenguaje específico de dominio para el diseño de solución de integraciones**. Editorial: The Distributed Group.
- Fuenmayor, C. y Morales, A. (2008). **Arquitectura de Servicios de e-salud para Salud Chacao sobre Redes Inalámbricas Metropolitana**. Trabajo de Ascenso. Caracas, Venezuela. Universidad Central de Venezuela.
- García, A. (2007). **Arquitecturas Empresariales. Orientación a Servicios (SOA) y Gestión de Procesos de Negocio (BPM)**. Recuperado el 28 de abril de 2010 de URL: <http://www.scribd.com/doc/12732986/Arquitecturas-Empresariales-Orientacion-a-Servicios-SOA-y-Gestion-de-Procesos-de-Negocio-BPM>
- Gaya, J (1999). **Integración de la información en el Laboratorio Clínico como modelo de informática departamental**. Recuperado el 16 de junio de 2011 de URL: <http://sediglac.org/documentos/colaboraciones/informaticalab.shtml>
- Gómez, A. y otros. (2003). **Implementación de mensajería HL7 en un sistema de**

- consulta de resultados de laboratorio.** Recuperado el 15 de febrero de 2010 de URL: http://www.frcu.utn.edu.ar/deptos/depto_3/32JAIIO/sis/SIS_06.pdf
- Hernández, K. (2008). **Desarrollo tecnológico y su vinculación con la educación.** México: Instituto de Ciencias, Humanidades y Tecnología de Guanajuato.
- Hernández, R., Fernández, C. y Baptista, P. (2006). **Metodología de la Investigación.** (6°. Ed.). México: McGraw Hill. Interamericana, S.A.
- Hohpe G. y Woolf B. (2003). **Enterprise Integration Patterns.** Addison-Wesley. ISBN 0321200683.
- IEEE Std 1471 (2004). **IEEE Recommended Practice for Architectural Description of Software Intensive Systems.** IEEE Software, Institute of Electrical and Electronics Engineers.
- Kruchten, P. (1995). **Architectural Blueprints. The 4 + 1 View of Software Architecture.** IEEE Software, Institute of Electrical and Electronics Engineers.
- Laudon, K. y Laudon, J. (1996). **Administración de los Sistemas de Información.** Tercera Edición. Prentice Hall Hispanoamericana, S.A. Naucalpan de Juárez.
- Ley Orgánica de Ciencia, Tecnología e Innovación (2001). **Gaceta Oficial No.37.291,** miércoles 26 de septiembre de 2001.
- Marroquin, N. (2009). **Sistemas de bajo costo para laboratorios clínicos en Guatemala.** Tesis Publicada. Guatemala: Universidad de San Carlos de Guatemala.
- Méndez, C. (2008). **Metodología. Diseño y desarrollo del proceso de investigación con énfasis en ciencias empresariales.** 4ta. edición. México: Editorial Limusa.
- Microsoft Corporation (2006). **La Arquitectura Orientada a Servicios (SOA) de Microsoft aplicada al mundo real.** Recuperado el 10 de junio de 2010 de URL: <http://www.microsoft.com/latam/servidores/applications.msp>
- O'brien, J. (2001). **Capítulo I. En: Sistemas de Información Gerencial.** 4ta. Edición. Colombia: Ed. Irwin. McGraw-Hill.
- Páez, J. (2006). **Diseño de una arquitectura de software para educación a distancia.** Trabajo de Grado. Barquisimeto, Venezuela. Universidad Centroccidental Lisandro Alvarado.
- Park, W., Yi, S., Kim, S., Song, J, y Kwak, J. (2005). **Association between the**

- implementation of a laboratory information system and the revenue of a general hospital**, Archives of Pathology and Laboratory Medicine.
- Pascual, C., Raventos, J., Pelegri, D., Bedini, J. Y Ballesta, A. (1994). **Sistemas de Información de Laboratorios (SIL)**. Recuperado el 03 de marzo de 2010 de URL: <http://www.uv.es/~docmed/documed/documed/1483.html>
- Perry, D. y Wolf, A. (1992). **Foundations for the study of software architecture**. ACM SIGSOFT Software Engineering Notes.
- Pressman, R. (2002). **Ingeniería de Software. Un enfoque practico**. McGraw.Hill/Interamericana de España.,
- Ramos, R. y Simón, A. 2007. **Prolab Sistema Automatizado para Equipos de Laboratorio Clínico**. Recuperado el 07 de marzo de 2010 de URL: <http://cencomed.sld.cu/socbio2007/trabajos/pdf/t124.pdf>
- Reynoso, C. (2006). **MSDN Introducción a la Arquitectura de Software**. Recuperado el 16 de junio de 2010 de URL: http://www.microsoft.com/spanish/msdn/arquitectura/roadmap_arq/intro.mspx.
- Reynoso, C. y Kicillof, N. (2004). **MSDN Estilos y Patrones en la Estrategia de Arquitectura de Microsoft**. Recuperado el 16 de junio de 2010 de URL: http://www.microsoft.co.ke/spanish/msdn/arquitectura/roadmap_arq/style.aspx.
- Rosales, R. (2009). **Los cuatro pilares del laboratorio clínico-asistencial. Bioquímica**. Recuperado el 02 de febrero de 2010 de URL: <http://redalyc.uaemex.mx/src/inicio/ArtPdfRed.jsp?iCve=57612011001>.
- Rozanski, N. y Woods E. (2005). **Software Systems Architecture: Working With Stakeholders Using Viewpoints and Perspectives**. Addison-Wesley Professional. ISBN:0321112296
- Rozanski, N. y Woods E. (2008). **An architectural description document template that provides a suggested structure for ADs when viewpoints and perspectives are used to drive the architectural definition process**. Recuperada el 17 de junio de 2011 de URL: <http://www.viewpoints-and-perspectives.info/>
- Sabino, C. (2004). **Como Hacer una Tesis**. Quinta Edición. Venezuela: Editorial Panapo.
- Siemens Medical Solutions Diagnostics (2006), **LIS Basics**.
- Siemens Medical Solutions Diagnostics (2007), **IMMULITE® LIS Manual**.
- TIBCO Software Inc. (2008). **The Power of Now, El papel de un bus de servicios**

empresariales (ESB) en una SOA. Recuperado el 16 de abril de 2010 de URL: [http://www.willydev.net/InsiteCreation/v1.0/WillyCrawler/2008.06.08.Articulo.El%20papel%20de%20un%20bus%20de%20servicios%20empresariales%20\(ESB\)%20en%20una%20SOA.pdf](http://www.willydev.net/InsiteCreation/v1.0/WillyCrawler/2008.06.08.Articulo.El%20papel%20de%20un%20bus%20de%20servicios%20empresariales%20(ESB)%20en%20una%20SOA.pdf)

Universidad Centroccidental Lisandro Alvarado (2002). Manual para la Presentación de Trabajo de Grado. Barquisimeto

Valera, R. (2005). **Desarrollo de una Herramienta para la Gestión de Documentos y Contenidos en el proceso enseñanza aprendizaje dentro del Decanato de Ciencias y Tecnología de la UCLA.** Trabajo de Grado. Barquisimeto, Venezuela. Universidad Centroccidental Lisandro Alvarado.

Vázquez, J. (2010). **¿Por qué un Enterprise Service Bus (ESB)?** Recuperado el 14 de abril de 2010 de URL: <http://blogs.tecsisa.com/articulos-tecnicos/por-que-un-enterprise-service-bus/>

Villegas N. y Tamura G. (2008). Aspectos claves en la definición y adopción de estándares de interoperabilidad electrónica de datos: El caso de HL7 en el área de la salud. Vol. 6 No. 12. Universidad ICESI. Recuperado el 16 de junio de 2011 de URL: http://bibliotecadigital.icesi.edu.co/biblioteca_digital/bitstream/item/1840/1/3aspectos_claves.pdf

Zubillaga, B. y Acevedo, A. (2001). **Experiencia en la realización de interfaces con equipos de laboratorios clínicos.** Recuperado el 03 de marzo de 2010 de URL: <http://www.hab2001.sld.cu/arrepdf/00176.pdf>