

**SISTEMA DE INFORMACIÓN WEB BASADO EN LA ARQUITECTURA
ORIENTADA A SERVICIOS (SOA) PARA LA GESTIÓN DE USUARIOS,
PACIENTES, PROVEEDORES, HISTORIAS CLÍNICAS Y REPORTES DE LA
FUNDACIÓN IPS PARA LA SALUD FÍSICA Y MENTAL FIME.**

JOSÉ MANUEL CONDE ESCOBAR

Código: 1356431-3743

jose.conde@correounivalle.edu.co

**Universidad del Valle Sede Tuluá
Facultad de Ingeniería Sistemas
Escuela de Ingeniería de Sistemas y Computación
Ingeniería de Sistemas
Tuluá
2017**

**SISTEMA DE INFORMACIÓN WEB BASADO EN LA ARQUITECTURA
ORIENTADA A SERVICIOS (SOA) PARA LA GESTIÓN DE USUARIOS,
PACIENTES, PROVEEDORES, HISTORIAS CLÍNICAS Y REPORTE DE LA
FUNDACIÓN IPS PARA LA SALUD FÍSICA Y MENTAL FIME.**

JOSÉ MANUEL CONDE ESCOBAR

Código: 1356431-3743

jose.conde@correounivalle.edu.co

Trabajo de grado para optar al título de Ingeniero de Sistemas

Directora: DIANA LORENA VELANDIA VANEGAS. MSc.

**Universidad del Valle Sede Tuluá
Facultad de Ingeniería Sistemas
Escuela de Ingeniería de Sistemas y Computación
Ingeniería de Sistemas
Tuluá
2017**

AGRADECIMIENTOS

Agradezco principalmente a mi familia, a mis padres por el apoyo, los consejos y los regaños que me mantuvieron firme a lo largo de estos años, también a Diana, por el regalo más grande que puede recibir un hombre, un hijo. Juan Manuel tú me has dado impulso para continuar luchando por mis metas. Los amos.

Agradezco a todos y cada uno de mis profesores, compañeros, amigos que día a día contribuyeron de una u otra forma en mi crecimiento personal y profesional. Especialmente, mi agradecimiento y admiración quiero dirigirlos a la profesora Diana Lorena Velandia por su inmensa colaboración durante este proceso.

Mil gracias.

CONTENIDO

<u>AGRADECIMIENTOS</u>	<u>III</u>
<u>GLOSARIO</u>	<u>X</u>
<u>1 PLANTEAMIENTO DEL PROBLEMA</u>	<u>17</u>
<u>2 OBJETIVOS.....</u>	<u>21</u>
2.1 OBJETIVO GENERAL.....	21
2.2 OBJETIVOS ESPECÍFICOS	21
<u>3 ORGANIZACIÓN DEL DOCUMENTO</u>	<u>22</u>
<u>CAPÍTULO 1 – MARCO DE REFERENCIA</u>	<u>23</u>
<u>4 ENTIDADES PRESTADORAS DE SALUD (IPS).....</u>	<u>23</u>
<u>5 FUNDACIÓN IPS FIME.....</u>	<u>25</u>
<u>6 CONCEPTOS Y HERRAMIENTAS DE INGENIERÍA DE SOFTWARE</u>	<u>26</u>
6.1 SISTEMAS DE INFORMACIÓN WEB.....	26
6.2 ARQUITECTURAS PARA EL DESARROLLO DE SISTEMAS DE INFORMACIÓN WEB29	
6.3 ARQUITECTURA ORIENTADA A SERVICIOS SOA.....	35
6.3.1 ARQUITECTURAS DE SERVICIOS WEB DEL W3C	35
6.3.2 CONSIDERACIONES DE SEGURIDAD EN LA ARQUITECTURA SOA	37
6.4 IMPLEMENTACIONES PARA APLICACIONES SOA.....	38
6.4.1 SOAP. COMO SE MENCIONÓ ANTERIORMENTE SOAP ES UNA TECNOLOGÍA RECOMENDADA POR W3C QUE ESTABLECE LA COMUNICACIÓN ENTRE CLIENTE Y SERVIDOR POR	

MEDIO DE MENSAJES ESTRUCTURADOS EN XML SOBRE PROTOCOLOS DE LA CAPA DE APLICACIÓN COMO LO ES HTTP.	38
6.4.2 REST.	41
6.5 METODOLOGÍAS PARA EL DESARROLLO DE SISTEMAS DE INFORMACIÓN	
WEB SOA	42
6.6 PLATAFORMA DE DESARROLLO JAVA	46
APLICACIONES JAVA J2EE	46
6.6.1 ORM (MAPEO OBJETO RELACIONAL)	47
6.6.2 DAO (DATA ACCESS OBJECT)	47
6.6.3 JPA Y JPQL	48
6.6.4 EJB	48
6.6.5 MANAGED BEAN.....	48
6.6.6 JSF (JAVA SERVER FACES)	48
6.6.7 PRIMEFACES.....	49
6.6.8 SERVIDOR DE APLICACIONES GLASSFISH	49
6.6.9 MYSQL	50
6.6.10 SERVICIOS WEB.....	50
<u>7 DESARROLLO DEL PROYECTO</u>	52
7.1 DESARROLLO DE METODOLOGÍA SOMA	54
7.1.1 DISCIPLINA MODELADO DE NEGOCIO.....	54
7.1.2 DISCIPLINA ANÁLISIS Y DISEÑO	57
7.1.3 DISCIPLINA IMPLEMENTACIÓN.....	61
7.2 HERRAMIENTAS USADAS	62
7.3 IMPLEMENTACIÓN DEL PROYECTO	65
<u>CAPÍTULO 3 – RESULTADOS Y ANÁLISIS</u>	68
<u>8 RESULTADOS</u>	68
8.1 RESULTADOS IMPLEMENTACIÓN PROYECTO EN LA FUNDACIÓN IPS FIME	69

8.2	RESULTADOS USO DE ARQUITECTURA ORIENTADA A SERVICIOS SOA.....	70
<u>CAPÍTULO 4 – CONCLUSIONES Y TRABAJOS FUTUROS</u>		<u>71</u>
<u>13</u>	<u>CONCLUSIONES.....</u>	<u>71</u>
<u>14</u>	<u>TRABAJOS FUTUROS.....</u>	<u>72</u>
<u>15</u>	<u>REFERENCIAS BIBLIOGRÁFICAS</u>	<u>73</u>

LISTA DE FIGURAS

<i>Ilustración 1 Organigrama Fundación IPS FIME</i>	25
<i>Ilustración 2 Modelo de 3 Capas Físicas</i>	30
<i>Ilustración 3 Modelo de 3 Capas Lógicas</i>	30
<i>Ilustración 4 Los servicios Web (SOAP) en Funcionamiento</i>	34
<i>Ilustración 5 Modelos de arquitectura de servicios del W3C</i>	37
<i>Ilustración 6 Mensaje SOAP Aplicacion agencia de viajes</i>	40
<i>Ilustración 7 Fases clave de la metodología SOMA</i>	45
<i>Ilustración 8 Arquitectura de las Aplicaciones Web</i>	47
<i>Ilustración 9 Servicios de aplicación web para FIME</i>	53
<i>Ilustración 10 Caso de uso MS1-Usuarios</i>	56
<i>Ilustración 11 Diagrama BMP MS1 Pacientes</i>	59
<i>Ilustración 12 Base de Datos MS1-Usuarios</i>	60
<i>Ilustración 13 Conjunto de Tecnologías Java EE</i>	62
<i>Ilustración 14 Relación de módulos con respecto al consumo de servicios</i>	65
<i>Ilustración 15 Captura de pantalla Gestion de Usuarios</i>	66
<i>Ilustración 16 Captura de pantalla Gestión de proveedores, registro de empresas</i>	66
<i>Ilustración 17 Captura de patanlla Gestion de pacientes</i>	67

LISTA DE TABLAS

<i>Tabla 1 Procesos de la Fundación FIME</i>	20
<i>Tabla 2 Aplicaciones de escritorio y web</i>	28
<i>Tabla 3 Recursos REST Agenda Telefónica</i>	41
<i>Tabla 4 Disciplinas metodología SOMA</i>	44
<i>Tabla 5 Requerimientos MS1-Usuarios Elaboración Propia</i>	56
<i>Tabla 6 Descripción de Servicios</i>	58
<i>Tabla 7 Prueba funcional RF05-MS4 Historias</i>	62
<i>Tabla 9 Objetivo vs Resultados Obtenidos</i>	68

LISTA DE ANEXOS

<i>Anexo 1 Formato HC-Consulta Externa.....</i>	<i>19</i>
<i>Anexo 2 Formato HC-Audiometría.....</i>	<i>19</i>
<i>Anexo 3 Formato HC- Optometría.....</i>	<i>19</i>
<i>Anexo 4 Formato HC-Salud Ocupacional.....</i>	<i>20</i>
<i>Anexo 5 Requerimientos para MS2-Pacientes.....</i>	<i>54</i>
<i>Anexo 6 Requerimientos MS3-Proveedores.....</i>	<i>54</i>
<i>Anexo 7 Requerimientos MS4-Historias Clínicas.....</i>	<i>54</i>
<i>Anexo 8 casos de uso MS2-Pacientes.....</i>	<i>57</i>
<i>Anexo 9 casos de uso MS3-Proveedores y.....</i>	<i>57</i>
<i>Anexo 10 casos de uso MS4-Historias Clínicas.....</i>	<i>57</i>
<i>Anexo 11 diagrama BPM MS1-Usuarios.....</i>	<i>58</i>
<i>Anexo 12 diagrama BPM MS3-Proveedores.....</i>	<i>58</i>
<i>Anexo 13 diagrama BPM MS4-Historias.....</i>	<i>58</i>
<i>Anexo 14 modelo de base de datos MS1-Usuarios,.....</i>	<i>61</i>
<i>Anexo 15 modelo de base de datos MS2-Pacientes.....</i>	<i>61</i>
<i>Anexo 16 modelo de base de datos MS3-Proveedores.....</i>	<i>61</i>
<i>Anexo 17 Pruebas.....</i>	<i>61</i>

GLOSARIO

FRAMEWORK: se refiere a una estructura de software compuesta de componentes personalizables e intercambiables para facilitar y agilizar el desarrollo de aplicaciones web. [1]

JAVA SERVER FACES: es un framework basado en el modelo vista controlador para aplicaciones java basadas en web, permitiendo que cada miembro del equipo de desarrollo pueda enfocarse en su parte del proceso, y proporciona un sencillo modelo de programación para enlazar las piezas. [2]

SERVIDOR DE APLICACIONES GLASSFISH: como su nombre lo indica es un servidor de aplicaciones con una licencia de código abierto (open source) que ofrece un dominio de ejecución donde se valida la llamada y ejecución de métodos definidos en los beans. [3]

ENTERPRISE BEANS: proporciona un estándar para el desarrollo de clases que encapsulan la funcionalidad y las reglas de negocio que luego serán accedidas desde aplicaciones clientes. [3]

USUARIO: Su definición es utilizada en el campo de la informática para referirse a la persona que utiliza un hardware o software con determinados privilegios para interactuar con él sistema. [4]

PACIENTE: persona que recibe tratamiento médico. [5]

HISTORIA CLÍNICA: es un documento privado, obligatorio y sometido a reserva, en el cual se registran cronológicamente las condiciones de salud del paciente, los actos médicos y los demás procedimientos ejecutados por el equipo de salud que intervienen en dicha atención. Dicho documento solo puede ser conocido por tercero previa autorización del paciente o por la ley. [6]

AUDIOMETRÍA: es una prueba para valorar la capacidad auditiva a través de la estimulación del sistema auditivo por medio del uso de sonidos puros de diferentes tonos a una misma intensidad o de igual tono a diferentes intensidades. [7]

OPTOMETRÍA: Evaluación del sistema visual permitiendo realizar diagnósticos de los defectos visuales presentes en el trabajador, así como la necesidad del uso de corrección óptica de forma permanente o preventiva, con la determinación de la fórmula de corrección óptica necesaria para dicho defecto de refracción.[8]

PROVEEDOR: persona o empresa encargada de suministrar servicios o productos a otra empresa.

SERVICIO WEB: Un servicio web es una interfaz de software que describe un conjunto de operaciones a las cuales se puede acceder por la red a través de mensajería XML estandarizada. Usa protocolos basados en el lenguaje XML con el objetivo de describir una operación para ejecutar o datos para intercambiar con otro servicio web. Un grupo de servicios web que interactúa de esa forma define la aplicación de un servicio web específico en una arquitectura orientada a servicios (SOA) [9].

EPS: Entidad prestadora de salud donde la mayoría de ciudadanos se afilian según la ley. [10]

IPS: Es una institución prestadora de servicios de salud, como hospitales, clínicas, consultorios o laboratorios. [10]

FIME: Fundación IPS para la salud física y mental.

SOA: Es un estilo de arquitectura basada en servicios reutilizables, con interfaces publicas bien definidas, donde consumidores y proveedores de servicios interactúan en forma desacoplada para realizar los procesos de negocio. [11]

SOAP: es un protocolo de comunicaciones que encapsula los mensajes y permite la comunicación entre dos servicios. Se basa en XML definiendo como el cliente puede invocar una operación del servicio web y obtener resultados. [12]

REST: es un modelo para construir servicios web empleando las operaciones del protocolo HTTP, básicamente Post, Get, Put, Del [12]

WSDL: es un lenguaje basado en XML para definir y describir un servicio web. [12]

SDA: Documento de Arquitectura de Software.

RESUMEN

En este proyecto se diseñó e implementó una aplicación web basada en la arquitectura (SOA). Una arquitectura que junto a la metodología de desarrollo SOMA, un híbrido entre la metodología RUP[13] y un enfoque basado en modelo de negocios, exigió un análisis tanto de los requerimientos de la aplicación, como del modelo de negocio de la fundación IPS FIME. En el análisis del modelo de negocio se identificaron políticas de la organización claves para establecer el alcance del proyecto y también, se clasificaron actividades en procesos de negocio con el fin de detectar los servicios a implementar. El siguiente paso, fue diseñar los servicios de acuerdo a los objetivos de la empresa, es decir a su modelo de negocio y políticas internas apoyándose en casos de usos y diagramas de Modelos y Notación de Procesos de Negocio (BPMN). Por otra parte, en la implementación se tomó como base la tecnología REST [12], una alternativa a SOAP [14] recomendado por W3C para aplicaciones robustas, la cual establece criterios para la comunicación entre servicios web empleando el protocolo HTTP. Una vez implementados los servicios en módulos de usuarios, pacientes, proveedores e historias clínicas, se integraron cada uno de estos módulos en una aplicación web desarrollada haciendo uso del framework Java Server Faces (JSF), el servidor de aplicaciones web GlassFish V4.0, el gestor de base de datos MySQL y la biblioteca de componentes PrimeFaces.

Palabras claves: SOA, SOAP, Web Service, Servicios, RestFULL,, REST, JSF, SOMA, RUP, BPMN.

ABSTRACT

In this project, a web application based on architecture (SOA) was designed and implemented. An architecture that in conjunction with the SOMA development methodology, a hybrid between the RUP methodology [13] and a business model approach, requires an analysis of both, the application requirements and the business model of the IPS Fime Foundation. In the analysis of the business model, key organizational policies were identified to establish the scope of the project and also, there were classified activities in business processes in order to detect the services to be implemented. The next step was to design the services according to the objectives of the company, that is to say its business model and internal policies supported on cases of use and Business Process Model and Notation (BPMN) diagrams. On the other hand, the implementation was based on REST technology [12], an alternative to SOAP [14] recommended by the W3C for robust applications, which establish criteria for communication between web services by implementing the HTTP protocol. Once the services were implemented in users, patients, providers and clinical histories modules, each of these were integrated into a web application developed using the Java Server Face (JSF) framework, the web application server GlashFish V4.0, the database management system MySQL and the PrimeFaces component library.

Keywords: SOA, SOAP, Web Service, Service, RestFULL,, REST, JSF, SOMA, RUP, BPMN.

INTRODUCCIÓN

En temas de salud el gobierno ha generado una estructura jerárquica para la prestación de servicios de salud, pasando por las entidades promotoras de salud (EPS) hasta su nivel más bajo que son los centros médicos, los hospitales o las instituciones prestadoras de salud más conocidas como IPS.

La fundación IPS para la salud física y mental FIME es una institución privada sin ánimo de lucro la cual se encuentra certificada en el nivel de atención III, este nivel define los niveles de responsabilidad y complejidad en la atención de pacientes. Con una experiencia de más de dos años prestando los servicios de salud en el municipio de El Cerrito, ya cuenta con un flujo diario de más de 15 pacientes generando datos de todo tipo en relación a los servicios prestados, sin embargo, a pesar de la buena calidad del servicio han generado problemas en el manejo de información entorpeciendo algunos procesos.

Debido a esto, se buscó sistematizar las actividades de registro de pacientes, y proveedores, de gestión de historias clínicas y de reportes con el fin de disminuir problemas en la integridad y seguridad de los datos que son generados a diario por la fundación, ya que se ha evidenciado que estos datos en su mayoría son de carácter privado y se encuentra en distintos formatos como hojas de cálculo, correos electrónicos, memorias usb, cd's o papel, siendo accesibles para cualquier empleado.

Teniendo en cuenta el problema que se detectó en la fundación, se determinaron una serie de objetivos con el fin de desarrollar un conjunto de cuatro módulos, un módulo para el registro y validación de usuarios con el fin de establecer criterios de seguridad al acceso de la aplicación, un módulo para el registro de pacientes y

proveedores, un módulo para la gestión de historias clínicas: audiometría, consulta externa, optometría y salud ocupacional y otro para reportes.

Finalmente, se especificaron claramente las funcionalidades de dicho sistema, el cual se desarrolló bajo la metodología SOMA (Service Oriented Modeling and Architecture), que a pesar de ser pesada permitió realizar el análisis y diseño de este proyecto contando con un proceso base sólido soportado en casos de uso y diagramas para un mejor control durante el desarrollo.

1 PLANTEAMIENTO DEL PROBLEMA

La IPS FIME (Fundación para la salud física y mental) brinda atención médica en El Cerrito-Valle y sus alrededores desde hace más de dos años, con más de 10 empleados y una planta física compuesta por tres consultorios médicos y uno odontológico, donde se viene prestando los servicios de medicina general, psicología, psiquiatría, ginecología, urología, medicina laboral, fonoaudiología, fisioterapia y laboratorio clínico.

Cada uno de estos servicios genera información de carácter administrativo y financiero, como el registro de usuarios o pacientes, ordenes médicas, asignación de citas, registro de especialistas, e información de los costos y gastos de funcionamiento. Se atienden aproximadamente un promedio de 15 pacientes diarios, y la información es almacenada en diferentes formatos como hojas de cálculos, papel y correos electrónicos.

Con la información en diferentes formatos, se hacía difícil consultar una determinada historia clínica y cualquier otro tipo de información. Por ejemplo, en la gestión de los pacientes se evidenciaba redundancia en los datos, encontrándose varios registros del mismo paciente, y registros incompletos de proveedores. Por otro lado, en la gestión de las historias clínicas se relacionaban especialistas que no se encuentran registrados, y órdenes médicas de las que no se conocía su estado actual. En los reportes que se generaban en hojas de cálculos se exponían inconsistencias con respecto a los soportes físicos, como la cantidad de citas que no concordaba con los pacientes atendidos. Además, no se podía garantizar la integridad de la información registrada, los resultados que reflejaba un reporte, ni la seguridad de la información, ya que la IPS estaba expuesta a la manipulación de cualquier empleado, porque no se contaba con mecanismos que controlarían el acceso de la información, reflejando un problema de seguridad, como ocurre en el diagnóstico de un paciente que puede ser alterado en una historia clínica al igual que los resultados

de un examen ya que se encuentran en carpetas compartidas en la red donde todos los empleados tienen acceso.

Debido al creciente número de pacientes que se ha evidenciado en los últimos dos meses se estimaba que el volumen de información afectaría la integridad de los datos y el tiempo de ejecución en las tareas de asignación de citas, registro y consulta de pacientes/proveedores, la gestión de historias clínicas, y la elaboración de reportes estadísticos de las dichas tareas.

Finalmente, y teniendo en cuenta los aspectos nombrados, se implementa una solución informática que apoye la gestión de información de los pacientes e historias clínicas, agilizando procesos y llevando un control total sobre la información generada. En la tabla 1 se muestran los principales procesos manejados por la empresa y como se desarrollaban antes de la implementación del sistema propuesto.

Proceso	Desarrollo
Gestión de usuarios	Total ausencia en el control de usuarios. Se evidenciaban archivos en red de carácter privado como ordenes médicas o historias clínicas, donde cualquier persona con acceso a un computador conectado a la misma, podría consultarlos, manipularlos y eliminarlos.
Gestión de información de pacientes	Se generaba una base de datos en un libro de Excel donde se registraban los datos personales del paciente por cada cita. Es decir, si el paciente requería citas de control periódicas, este se registraba una y otra vez en el libro creando una base de datos con información redundante. Por otro lado, esos datos registrados tenían que volverse a diligenciar al momento de generar la historia clínica.
Gestión de información de proveedores	La información de los proveedores de insumos, servicios de laboratorio o radio-x, no se encontraban en una base de

	<p>datos que permitiera gestionarla. Cuando se necesitaba consultar la información de un proveedor o hacer contacto con él, se remitían al historial del correo electrónico para buscar dicha información, o en el peor de los casos en facturas archivadas físicamente.</p>
<p>Gestión de información de Historia Clínica de Consulta externa</p>	<p>Para el registro de la historia clínica de consulta externa se contaba con un formato hecho en un procesador de texto el cual se imprimía y se diligenciaba a mano. Posteriormente se almacenaba en un archivador.</p> <p>Para observar el formato, Ver Anexo 1 Formato HC-Consulta Externa</p>
<p>Gestión de información de Historia Clínica de audiometría</p>	<p>Para el registro de la historia clínica de audiometría se contaba con un formato hecho en un procesador de texto el cual se imprimía y se diligenciaba de forma manual. Este formato además de los datos del paciente y el examen, requería dibujar una gráfica de líneas para reflejar el resultado de la audiometría. Posteriormente se almacenaba en un archivador.</p> <p>Para observar el formato, Ver Anexo 2 Formato HC-Audiometría.</p>
<p>Gestión de información de Historia Clínica de Optometría</p>	<p>Para el registro de la historia clínica de Optometría se contaba con un formato el cual se imprimía y se diligenciaba de forma manual. Posteriormente se almacenaba en un archivador.</p> <p>Para observar el formato, Ver Anexo 3 Formato HC-Optometría.</p>

<p>Gestión de información de Historia Clínica de Salud Ocupacional</p>	<p>Para el registro de la historia clínica de Salud ocupacional se contaba con un formato diseñado y diligenciado en una hoja de cálculo, el cual se imprimía para almacenarse en un archivador. Esta historia requería de otras historias clínicas o exámenes diagnóstico del paciente para consultar antecedentes, resultados, o la misma información del paciente haciendo demorada la consulta.</p> <p>Para observar el formato, Ver Anexo 4 Formato HC-Salud Ocupacional</p>
--	---

Tabla 1 Procesos de la Fundación FIME

Fuente: Elaboración propia.

2 OBJETIVOS

2.1 OBJETIVO GENERAL

Desarrollar un sistema de información web basado en la arquitectura orientada a servicios para la gestión de usuarios, historias clínicas y generación de reportes para la IPS FIME.

2.2 OBJETIVOS ESPECÍFICOS

- Modelar los procesos de gestión de usuarios, pacientes, proveedores, historias clínicas y reportes.
- Especificar los servicios definiendo los contratos funcionales en el desarrollo de la aplicación web orientada a servicios.
- Implementar módulo para la gestión de usuarios y privilegios en el sistema, pacientes y proveedores de servicio de la IPS.
- Implementar módulo de historia clínica que permita la gestión de información de las áreas de salud ocupacional, psicología, audiometría, optometría y medicina general.
- Desarrollar módulo para generar reportes relacionados con los pacientes, proveedores e historias clínicas de la IPS FIME.
- Validar módulos del sistema teniendo en cuenta los servicios, sus interfaces y la interacción con otros servicios.

3 ORGANIZACIÓN DEL DOCUMENTO

A continuación, en el Capítulo I se describen los conceptos teóricos que aportan al trabajo: metodología SOA del plug-in SOMA, Framework JSF Prime Faces, Arquitectura de Aplicaciones Multinivel, Glassfish. En el Capítulo 2 se presenta el desarrollo del proyecto, donde se describe la metodología utilizada incluyendo la fase de pruebas e implementación. En el Capítulo 3 se describen los resultados obtenidos en cada uno de los objetivos propuestos. En el Capítulo 4 se presentan las conclusiones, además de los trabajos futuros que se pueden emprender. Finalmente, se presenta la lista de referencias bibliográficas que soportan el desarrollo del trabajo.

CAPÍTULO 1 – MARCO DE REFERENCIA

En esta sección se presenta el marco teórico del trabajo, la cual aborda 4 temas: en primer lugar, se lleva a cabo la revisión acerca del estado actual de los procesos en la fundación IPS FIME. En segundo lugar, se abordan las bases conceptuales del proceso de usuarios, pacientes, proveedores e historias clínicas. En tercer lugar, se aborda las bases conceptuales de los modelos y herramientas usados para el desarrollo del proyecto como son, los sistemas de información, la ingeniería de requisitos y su importancia en el desarrollo de software, la metodología SOA del plug-in SOMA y finalmente el uso de plataforma Java con el Framework JSF PrimeFaces.

4 ENTIDADES PRESTADORAS DE SALUD (IPS)

En Colombia el sistema general de salud cuenta con una estructura de tres niveles, en el primer y más alto nivel encontramos al gobierno que a partir del Ministerio de Salud actúa como ente de control y regulador mediante leyes y resoluciones frente a los siguientes niveles. En el siguiente o segundo nivel se ubican todas las entidades promotoras de salud, estas entidades son conocidas como EPS y no prestan directamente servicios de salud, su función es promover servicios relacionados con la salud por medio de un mecanismo de aseguramiento vinculante que se define por un régimen contributivo o subsidiado por el gobierno. Ahora como último nivel y parte fundamental de esta estructura de salud se encuentran las instituciones prestadoras de salud (IPS). Dentro de estas instituciones tenemos clínicas, hospitales, centros de salud, laboratorios y/o consultorios, estos deben cumplir con unos exigentes requerimientos del gobierno para certificarse como (IPS) y poder ofrecer sus servicios a las entidades promotoras de salud [15], aunque las IPS pueden contar con un carácter privado y pueden ofrecer sus servicios de salud de manera independiente.

En las IPS interfieren varios actores los cuales podemos agrupar en los siguientes grupos:

- Usuarios o pacientes: Se caracterizan por solicitar un servicio relacionado con la salud como un examen de sangre ante la IPS. Los pacientes son de vital importancia y se deben considerar sus derechos y deberes, entre ellos el manejo de la información que los relaciona con la IPS como las historias clínicas.
- Proveedores de servicios de salud: Aquí podemos encontrar laboratorios clínicos, especialistas en medicina, oftalmología, pediatra, entre otros. También centros de salud con los que se contratan algún tipo de equipamiento médico como radiografías.
- Y como intermediario, toda la infraestructura física y humana de la IPS.

Las IPS al ser una entidad que tiene contacto directo con los usuarios y como prestadora de salud ante el gobierno debe buscar mecanismos para garantizar el acceso, la disponibilidad y la pertinencia de los servicios ofrecidos. Uno de los factores claves para cumplir con estas características se enfoca en el manejo de información, en la gestión de los datos que genera la IPS en relación a los actores mencionados. El gobierno es consciente y fomenta el uso de las TIC [16].

En la actualidad el sistema general de salud busca que todas las IPS públicas y privadas participen de los cambios tecnológicos apoyándose en las TIC en busca de mejorar el rendimiento y la calidad en la atención de los pacientes. Con los diferentes medios tecnológicos y el acceso a diferentes fuentes de información entidades como las IPS deben exigirse la implementación de nuevas formas de comunicación por la cuales puedan divulgar información y ofertas. El uso de las TIC en las instituciones trae beneficios importantes como señalan algunos libros: el mejoramiento de los sistemas de información sobre servicios sociales, el intercambio de información entre instituciones, la prestación de servicios públicos con mayor eficiencia, y el incremento en la transparencia de las políticas sociales de la misma institución [17].

5 FUNDACIÓN IPS FIME

La fundación IPS para la salud física y mental FIME es hoy la infraestructura de una institución prestadora de servicios de salud, privada sin ánimo de lucro, ubicada en la zona urbana del municipio de El Cerrito Valle, con fácil acceso a toda la población; la IPS cuenta con profesionales calificados que garantizan la prestación del servicio con calidad oportuna y eficacia. A continuación, se describe la organización de la fundación con respecto a sus funciones. En la ilustración 1 se muestra el organigrama de la empresa.

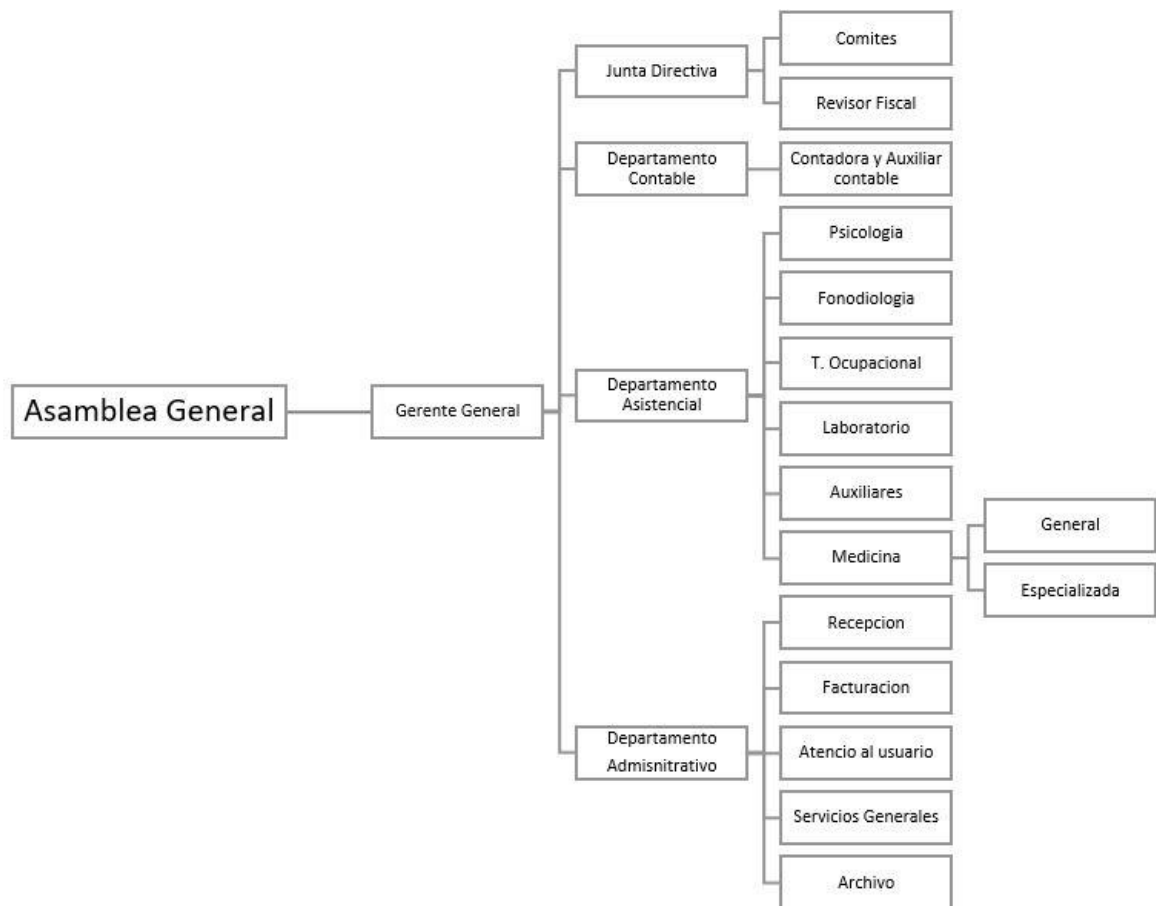


Ilustración 1 Organigrama Fundación IPS FIME

Fuente: Elaboración propia a partir de datos obtenidos en la empresa.

La fundación fue reconocida por la gobernación del valle del cauca con personería jurídica el 14 de agosto de 2014 por medio de la resolución No. 055, registrada con el número de identificación tributario Nit. 900.764.401-3 y habilitada en salud con el código del prestador No 762480978001 por la secretaría de salud de este mismo departamento.

Como fundación hace dos años viene desarrollando programas de apoyo social y orientación psicológica, brigadas de salud entre otras; gracias a su cuerpo médico, personal administrativo y profesionales de la salud, la fundación es reconocida como la institución con la oferta en salud física y mental que trabaja por el bienestar del ser humano, dando respuesta a las necesidades de salud.

6 CONCEPTOS Y HERRAMIENTAS DE INGENIERÍA DE SOFTWARE

6.1 SISTEMAS DE INFORMACIÓN WEB

En el desarrollo de software los sistemas de información web representan la tendencia a la hora de escoger sobre que plataforma diseñar un software. Entre los factores que determinan este fenómeno podemos mencionar la alta compatibilidad entre plataformas, es decir entre diferentes sistemas operativos y dispositivos electrónicos, otro factor es la disponibilidad de acceso inmediato ya que, si se dispone de un navegador web, una conexión a internet y en su defecto de una cuenta de usuario valida se puede acceder a la aplicación desde cualquier ubicación geográfica, por otro lado, el soporte y mantenimiento de estos SIW es más eficaz pues de manera rápida se pueden corregir bugs y aplicar actualizaciones, ya que se trabaja sobre un mismo sistema operativo y unas características de hardware claramente definidas.

Para entender el concepto de sistema de información, debemos tener en cuenta la definición de un sistema. Con base en autores como Alarcón (2006) que concibe el concepto como “un conjunto de componentes que se interconectan entre sí para lograr un objetivo en común” (p. 11), se toma como definición para un sistema de

información el aporte de los autores Laudon y Laudon que establecen: Un sistema de información (SI) como un conjunto de componentes interrelacionados que recolectan (o recuperan), procesan, almacenan y distribuyen información para apoyar los procesos de toma de decisiones y de control en una organización. Además de apoyar la toma de decisiones, la coordinación y el control, los sistemas de información también pueden ayudar a los gerentes y trabajadores del conocimiento a analizar problemas, visualizar temas complejos y crear nuevos productos [18].

Además, podemos agregar que los sistemas de información son los encargados de contener y gestionar la información referente a la organización. Esta información a diferencia de los datos ha sido modelada con el objetivo de que sea fácilmente interpretada por el ser humano, mientras que los datos son elementos que representan acciones dentro del contexto o piezas que conforman el mismo. Por ejemplo, se tiene una lista de usuarios con atributos como la identificación, nombre, teléfono la cual recibe la etiqueta de datos, pues no representan nada más. Una vez que se toman esos datos y se ajustan a un modelo factura de venta, que también se relaciona con una lista de productos, ahora se podría decir que se está generando información, ya que se dependiendo de la interpretación se puede determinar que usuarios han realizado compras, que productos se han vendidos o que no.

Tanto los datos como la información pueden hacer parte de un sistema de información basado en ordenadores o un sistema de información manual a lápiz y papel. En este caso consideremos únicamente los sistemas basados en ordenadores los cuales hacen usos de las tecnologías de la información.

La forma en que se requiere gestionar la información, es decir, los criterios de cómo voy a obtener los datos, como voy a moldearlos, que tan accesibles van a ser, la calidad de los mismo, y la forma de representar los resultados determinan que tipo

de sistema de información es el más viable. De manera general podemos tener dos opciones para enfocar un desarrollo, sistemas de información basados en aplicaciones de escritorio o basados en aplicaciones web.

Ambas opciones tienen sus ventajas y desventajas, teniendo en cuenta que una aplicación es un programa informático diseñado para automatizar una o varias tareas con un solo objetivo. Podemos encontrar aplicaciones de escritorio o web. En la tabla 2, se muestran las características de los dos tipos de aplicación que podemos encontrar.

Sistemas de información	
Aplicaciones de escritorio	Aplicaciones web
Su desarrollo se enfoca a un sistema operativo definido y características específicas de hardware. Se debe ajustar los requerimientos en el desarrollo de la aplicación si se quiere usar en más de un sistema operativo diferente.	Su desarrollo se enfoca a un servidor web que se ajuste a las características definidas por los requerimientos de la aplicación. Al contrario de una aplicación de escritorio estas aplicaciones no necesitan ser programadas para un sistema operativo u otro.
Se debe realizar una instalación por cada computador donde se quiera ejecutar. Limitando su acceso.	Solo es necesario acceso a internet, un navegador web y un usuario para ingresar a la aplicación. Brindando un acceso mucho mayor.
Difícilmente tiene protocolos de actualización para todas las instancias de la aplicación, y dependiendo del tamaño de la aplicación, así mismo se dificulta su actualización.	De la aplicación solo existe una instancia, la que se encuentra en el servidor a la cual pueden acceder muchos computadores. Esto hace que la actualización y el mantenimiento sea más rápido. Se corrige solo una vez.
Tienen un rendimiento muy bueno, sus tiempos de respuestas son más rápidos.	Actualmente con la velocidad de internet y la capacidad de procesamiento de los servidores web en aumento, los tiempos de respuesta frente a las aplicaciones de escritorios son muy similares.

Tabla 2 Aplicaciones de escritorio y web

6.2 ARQUITECTURAS PARA EL DESARROLLO DE SISTEMAS DE INFORMACIÓN WEB

Las aplicaciones web son un tipo especial de aplicaciones cliente/servidor [5], es decir son programas que sirven o consumen recursos de otro dentro de una red. A partir de esta premisa se define la arquitectura cliente/servidor la cual implica que existe una relación entre un proceso llamado cliente que solicita un servicio y otro proceso que responden a esta solicitud con el nombre de servidor. La arquitectura cliente/servidor es la base de muchos paradigmas de programación web que dan origen a otras, como por ejemplo arquitecturas de N-capas.

En arquitectura de n-capas encontramos modelos de dos capas donde una capa hace las veces de servidor de datos y la otra capa llamada cliente contiene la presentación y la lógica de negocio de la aplicación web. Por otro lado, encontramos modelos con características similares que contienen una capa de más, es decir tres capas (modelo, vista, controlador), una capa para el cliente con la presentación, una capa destinada para la lógica de negocio y finalmente otra para el servidor de datos. A partir de esta cantidad de niveles el concepto de capa no se limita a la organización lógica del código, si no que se expande a niveles físicos donde cada capa puede representar un servidor totalmente independiente, como se muestra en la Ilustración 2.

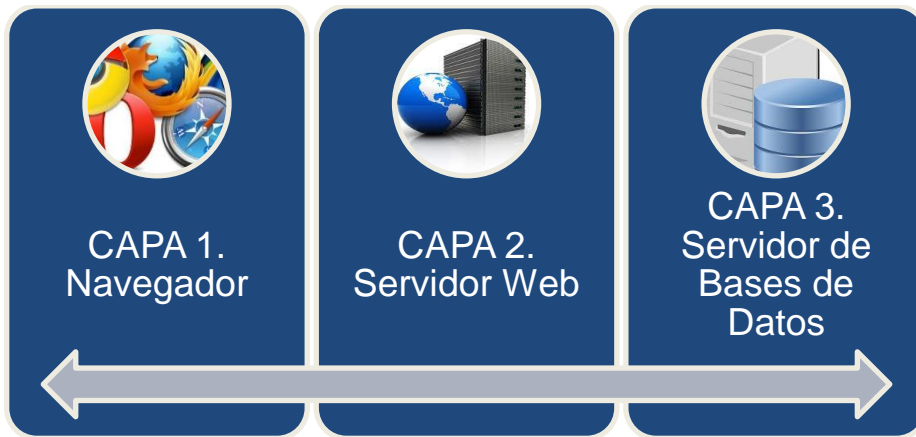


Ilustración 2 Modelo de 3 Capas Físicas

Fuente: Elaboración propia

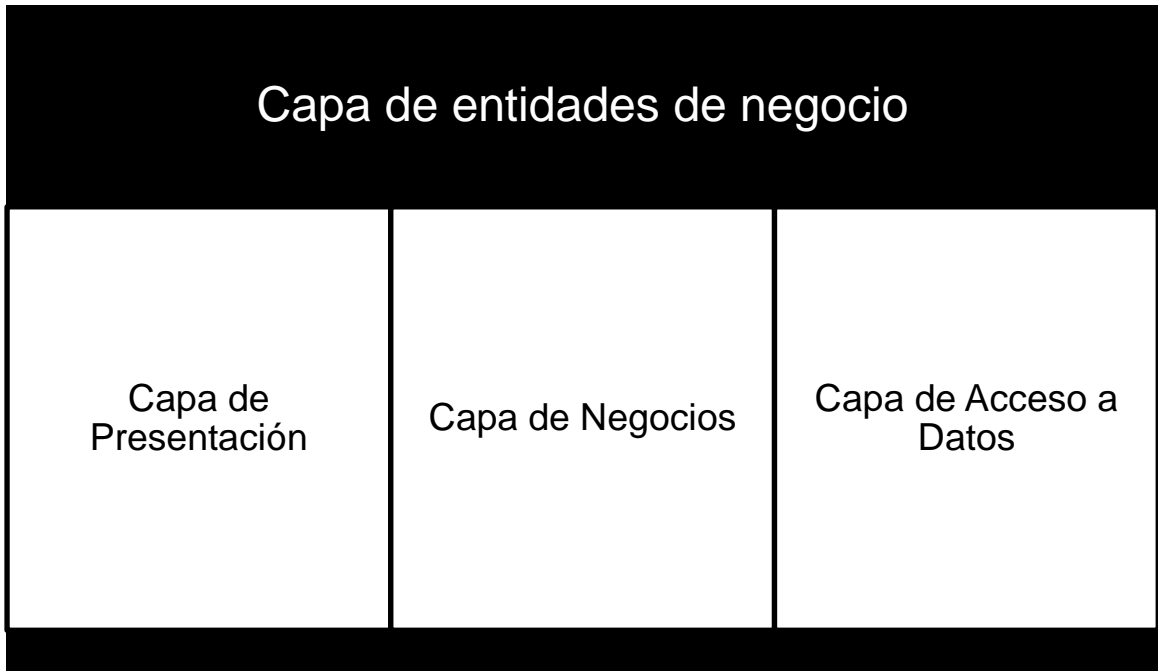


Ilustración 3 Modelo de 3 Capas Lógicas

Fuente: Elaboración propia

Como se mencionó anteriormente, a partir de modelos con 3 capas se toman en cuenta otros paradigmas, ya no solo se habla de una organización lógica del código, sino de una física, dando pie a modelos de sistemas distribuidos, pues la aplicación

web ya cuenta con una arquitectura descentralizada que incluye más de un servidor para realizar las tareas del sistema.

En un sistema distribuido el procesamiento de información se distribuye en varios computadores [6], donde uno de estos computadores puede ser un servidor de datos, un servidor de impresión y otro un servidor de correo. La comunicación entre estos servidores se realiza por medio de un programa llamado Middleware, [6] el cual se identifica como el intermediario de peticiones de objetos que permite gestionar datos sobre diferentes computadores que hacen parte del mismo sistema. A continuación, se mencionan los principales estándares [7] Middleware para la comunicación entre objetos distribuidos; CORBA¹ ligado a la gestión de objetos remotos, DCOM² ligado a sistemas operativos Windows y JAVA RMI³ ligado a java, donde cada uno de estos estándares aportan una API (interfaz de programación de aplicaciones) para el diseño de aplicaciones distribuidas.

Por otra parte, los sistemas distribuidos involucran un concepto Llamadas de Procedimientos Remotos (RPC, Remote Procedure Calls) [7]. RCP es un protocolo que permite ejecutar programas que están alojados en servidores. Este protocolo facilita las tareas en un sistema distribuido donde, por ejemplo, el servidor de datos es el encargado de compartir los datos como un recurso del sistema, y sobre este recurso se pueden realizar operaciones o procedimientos para agregar, modificar, o eliminar datos. Este protocolo facilita que el cliente no solo solicite un recurso, sino un procedimiento o la ejecución de un programa a dicho servidor. Cuando un cliente solicita un procedimiento a un servidor, realmente está solicitando que el servidor ejecute determinada aplicación web y le devuelva unos resultados.

A medida que los sistemas distribuidos fueron creciendo también se incrementaron las llamadas a procedimientos remotos, ya que al existir más recursos compartidos

¹ <http://www.corba.org/>

² <https://msdn.microsoft.com/es-es/library/cc201989.aspx>

³ <https://docs.oracle.com/javase/7/docs/platform/rmi/spec/rmiTOC.html>

las operaciones para gestionarlos, aumentaron. Esto evidenció problemas de compatibilidad entre plataformas, pues si bien el sistema garantizaba la disponibilidad de los recursos compartidos en él, no podría asegurar la comunicación entre servidores ya que cada servidor podría representar una plataforma y lenguaje de programación diferente, y los estándares actuales como DCOM de Microsoft se veían limitados frente a otros sistemas. Por otro lado, las RCP también exponían sus limitaciones frente a componentes físicos instalados para aumentar la seguridad de la red como un firewall, este componente bloquea el acceso a puertos en los servidores evitando establecer una conexión.

Frente a las limitaciones de los estándares y protocolos que se presentaban en los sistemas distribuidos a la hora de gestionar recursos, y la necesidad de crear un estándar para la comunicación entre distintas plataformas y lenguajes de programación, aparece el concepto de Web Service (Servicio Web) que integra el significado de servicio como procedimiento y lo que implica el termino web. Un Servicio Web [8] se puede definir como la referencia a una tecnología que permite que las aplicaciones se comuniquen entre sí a través de internet. Esta comunicación se logra por medio del intercambio de mensajes que no dependen de la plataforma ni del lenguaje de programación ya que se estructuran para usarse con protocolos de internet mediante un estándar XML⁴.

Los servicios web se pueden considerar como la evolución de las RCP, pues estos no se apoyan únicamente en protocolos TCP/IP, sino, también en protocolos estándar de internet como HTTP Y SMTP para enviar y recibir mensajes a través de la internet. Para una red el tráfico que genera el intercambio de mensajes entre servicios web es trafico normal, como si se tratase del tráfico durante el intercambio de correos electrónicos. Finalmente, los servicios web superan las limitaciones de sus antecesores estableciendo un conjunto de estándares para el intercambio de datos entre aplicaciones web.

⁴ <http://www.w3c.es/Divulgacion/GuiasBreves/TecnologiasXML>

Estándares como WSDL (Web Service Description Language) definen una interface de servicio, REST (Representational State Transfer) define un servicio basado en recursos y SOAP (Simple Object Access Protocol) define un servicio basados en mensajes. Estos estándares son los más usados en la actualidad a la hora de implementar servicios [7].

WSDL [9]: basándose en XML describe la forma en que los clientes se van comunicar con los servicios web existentes en un servidor. En general, establece un contrato entre el cliente y el servidor definiendo como, donde y que hacer dependiendo del mensaje que envía y recibe.

REST [7]: utiliza métodos HTTP para comunicarse con un servicio web, por ejemplo, usa POST para crear un recurso en el servidor, GET para obtener un recurso, PUT para cambiar el estado de un recurso o actualizarlo y DELETE para eliminar un recurso.

SOAP [7]: es un estándar recomendado por W3C para la comunicación entre cliente y servidor, donde esta comunicación se caracteriza por el intercambio de mensajes basados en XML sobre algunos protocolos como HTTP, SMTP o FTP. SOAP es el estándar más utilizado a la hora de implementar un mecanismo para comunicar dos servicios o más, ya que este además ofrece un encapsulamiento del mensaje con XML y, brinda seguridad desde el origen al destino aplicando el protocolo de seguridad WS-Security (Web Service Security).

En la siguiente ilustración se puede observar el funcionamiento de distintos servicios web, estos servicios se comunican por medio de un estándar SOAP manteniendo un contrato de servicio definido WSDL.

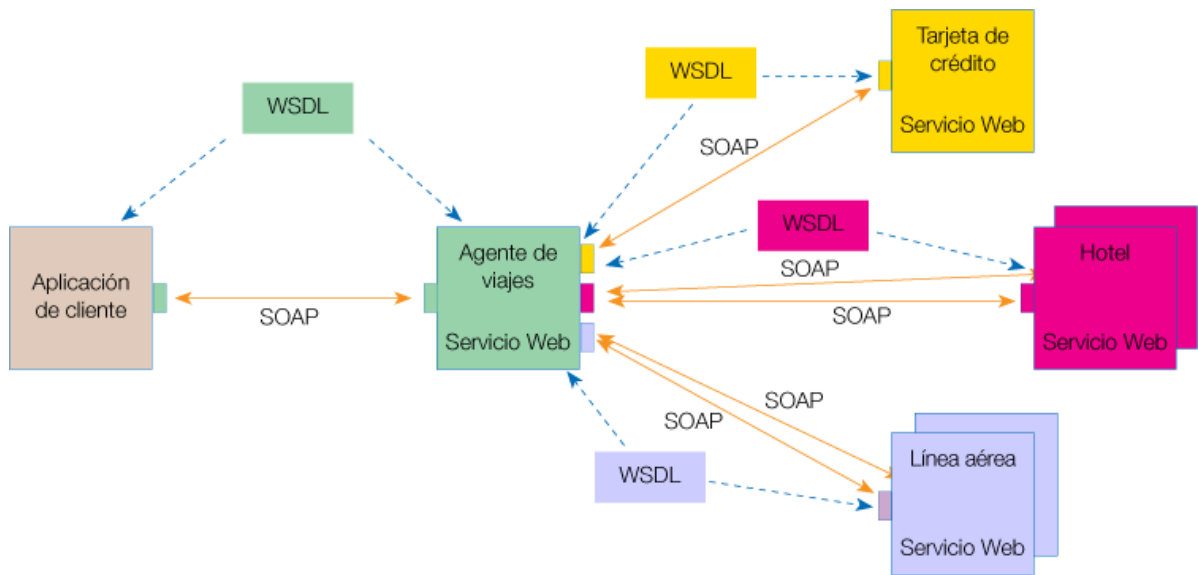


Ilustración 4 Los servicios Web (SOAP) en Funcionamiento

Fuente: <http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>

Por otro lado, en la ilustración anterior podemos observar cómo interactúan varios servicios. Este fenómeno se describe como un proceso de negocio, donde varios servicios totalmente independientes entre sí se orientan en una secuencia definida para alcanzar un objetivo concreto. Este fenómeno se basa en una arquitectura de software orientada a servicios (SOA) [10].

SOA (Service Oriented Architecture) se identifica como una arquitectura orientada a servicios que abstrae el modelo de negocio de una organización en sus servicios más básicos, estos servicios son reutilizables mediante una interfaz que genera un contrato, es decir la condiciones para que los proveedores (servidores) y consumidores (clientes) de dichos servicios interactúen para realizar procesos de negocios. Para lograr una mejor abstracción de un proceso de negocio SOA clasifica los servicios en básicos [10], intermedios, centrados en procesos y públicos dentro de un nivel denominado backend.

Precisando de una vez, SOA considera dos niveles: frontend que se encuentra en la parte del cliente iniciando y controlando la secuencia de un proceso de negocio suministrando datos de entrada, y backend del lado del servidor llevando a cabo los

procesos para obtener un resultado y devolverlo al nivel frontend. El nivel frontend se pueden ubicar estructuras HTML, estilos CSS y el lenguaje JavaScript, ya que este nivel interactúa directamente con el usuario, al contrario de lo que pasa en el nivel backend donde se estructura la lógica de negocio (transparente para el usuario) usando lenguajes como PHP, gestores de base de datos y frameworks para procesar la información suministrada por el usuario.

6.3 ARQUITECTURA ORIENTADA A SERVICIOS SOA

La arquitectura orientada a servicio es un modelo conceptual que permite el desarrollo de aplicación web distribuidas usando servicios de terceros o propios a través de la red.

El paradigma SOA es bastante complejo y está en constante desarrollo. La organización encargada de crear estándares SOA se llama OASIS (Organization for the Advancement of Structured Information Standards).

Ahora este modelo no hace referencia a que tecnología utilizar para implementar un servicio web, si asocia las siguientes: WSDL (Web Services Description Language), SOAP (Simple Object Access Protocol), REST (Representational State Transfer).

6.3.1 Arquitecturas de servicios web del w3c

El W3C propone una arquitectura de servicios web teniendo en cuenta modelos, conceptos y relaciones a modo de guía para los diseñadores de servicios web. Esta propuesta cuenta con cuatro modelos[12]:

6.3.1.1 Modelo orientado a mensaje

En este modelo de arquitectura se centra en los aspectos relacionados con los mensajes y su procesamiento, definiendo una estructura en los mensajes de tal manera que relacione los emisores y receptores. Entre las tecnologías que emplean este modelo podemos encontrar,

- **SOAP.** que estructura los mensajes por medio de XML y el transporte se lleva a cabo por protocolos HTTP, FTP, SMTP, etc. [12]
- **JMS API (Java Message Service).** Componente de Java EE que permite crear aplicaciones distribuidas donde sus elementos se comunican de forma asíncrona por medio de mensajes. [12]
- **AMQP API (Advanced Message Queuing Protocol).** Es un protocolo que permite el paso de mensajes que incluyen una capa de seguridad en aplicaciones distribuidas.[12]
- **MSMQ (Microsoft Message Queuing).** Es un protocolo desarrollado por Microsoft que se encuentra presente en sistemas operativos Windows Nt 4.0 y permite la comunicación de forma asíncrona de aplicaciones distribuidas.[19]

6.3.1.2 Modelo orientado a recursos

Se considera que el servicio web es un recurso, por lo tanto, se apoya en el protocolo HTTP para acceder a los recursos almacenado en los servidores a través de la url por medio de los métodos del protocolo (GET, POST, PUT...). El ejemplo más extendido de este tipo de modelo es la arquitectura REST.

6.3.1.3 Modelo orientado políticas y contratos de servicios

Este concepto hace referencia a la seguridad, el control de acceso, la calidad del servicio que deben considerarse en el desarrollo de la aplicación SOA. Por otro lado, se habla de un contrato que define como funciona el servicio a detalle, además de que acciones se pueden realizar para accederlo y consumirlo. Entre los estándares conocidos se pueden mencionar:

- WSDL (Web Services Description Language)[12]
- WADL (Web Application Description Language) [20]
- RAML (RESTful API Modeling Language) [12]

6.3.1.4 Modelo orientado a micro servicios

Es un concepto que ha generado polémica dentro de la arquitectura SOA, ya que hablar de este término no aporta nada nuevo debido a que la idea de servicio ya se encuentra presente en el modelo SOA. Este modelo consiste en la construcción de software a partir de numerosos micro servicios que representan cada una de las tareas más básicas del sistema. De esta manera busca minimizar las dependencias y maximizar la reutilización de software. [12], [21]

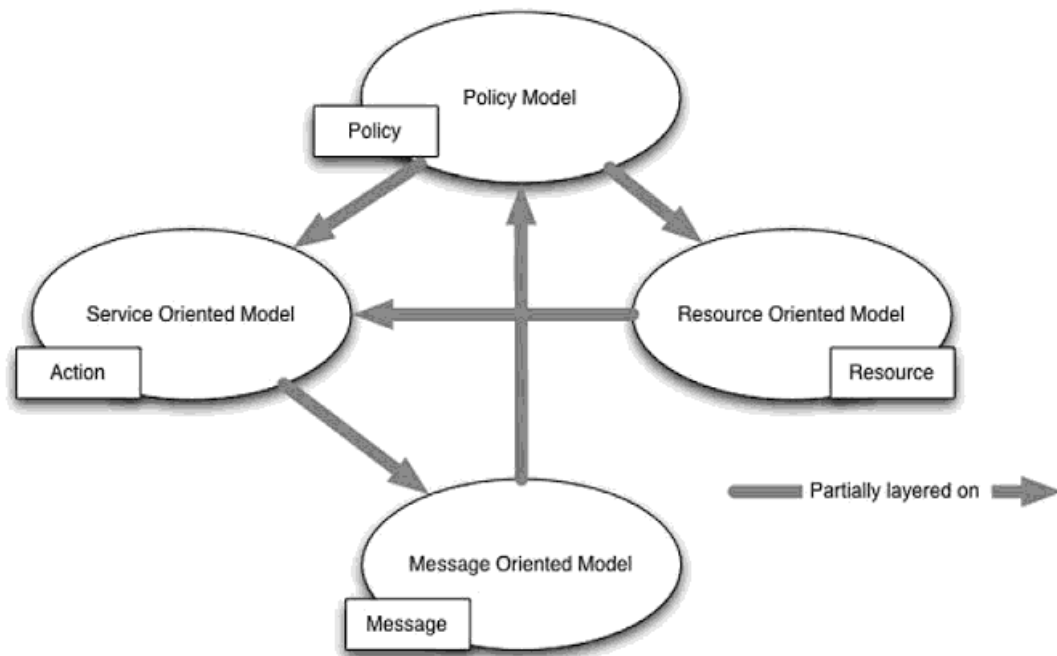


Ilustración 5 Modelos de arquitectura de servicios del W3C. Fuente: www.w3.org

6.3.2 Consideraciones de seguridad en la arquitectura soa

La seguridad es uno de los aspectos más importantes de una aplicación web por lo tanto debe representar el nivel de importancia de la información que será administrada por el sistema.

Los siguientes son criterios de seguridad que debe garantizar la aplicación orientada a servicio:

- **Autenticación;** Que el usuario que acceda al sistema es quien dice ser.
- **Autorización.** Un usuario puede realizar en el sistema solo lo que sus privilegios o credenciales le permitan.
- **Confidencialidad.** Garantizar que los datos y las comunicaciones no son accedida por persona no autorizadas.
- **Integridad.** Conservan la información en su estado original sin modificaciones.
- **No repudio.** Debe garantizar que la comunicación se envíe y reciba correctamente.
- **Disponibilidad.** El servicio debe estar disponible en todo momento.
- **Audibilidad.** Todas las acciones que se realizan en el sistema deben ser registradas y revisadas para determinar posibles o futuras fallas.

6.4 IMPLEMENTACIONES PARA APLICACIONES SOA

Actualmente se consideran dos tecnologías para el desarrollo de aplicaciones orientadas a servicios, SOAP y REST. Estas son muy diferentes entre ellas, mientras que SOAP se centra en el intercambio de mensajes en XML para comunicar cliente y servidor, REST ofrece pautas para implementar servicios web sobre el protocolo HTTP permitiendo recursos accesibles por los clientes. A continuación, se describe con más detalle cada tecnología mencionada.

6.4.1 SOAP. Como se mencionó anteriormente SOAP es una tecnología recomendada por W3C que establece la comunicación entre cliente y

servidor por medio de mensajes estructurados en XML sobre protocolos de la capa de aplicación como lo es HTTP.

6.4.1.1 Elementos de SOAP.

La base para la implementación SOAP se compone de una estructura fuertemente definida para dar origen a un mensaje. Esta estructura se debe centrar en cuatro elementos:

- **Envelope:** Es elemento principal que identifica el mensaje como SOAP
- **Header:** Es un elemento opcional, pero, puede extender la funcionalidad del servicio ya que se pueden contener información de control y procedimientos para el tratamiento del mensaje por los clientes.
- **Body:** Contiene la información que se desea compartir con el cliente. Es de carácter vital para la comunicación entre cliente y servidor.
- **Fault:** Se detallan errores que hayan podido ocurrir.

En la ilustración 6 se puede observar un ejemplo claro de la estructura de un mensaje.

```

<?xml version='1.0' Encoding='UTF-8' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
  <env:Header>
    <m:reservation xmlns:m="http://travelcompany.example.org/reservation"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next">
      <m:reference>uuid:093a2da1-q345-739r-ba5d-pqff98fe8j7d</m:reference>
      <m:dateAndTime>2007-11-29T13:20:00.000-05:00</m:dateAndTime>
    </m:reservation>
    <n:passenger xmlns:n="http://mycompany.example.com/employees"
      env:role="http://www.w3.org/2003/05/soap-envelope/role/next">
      <n:name>Fred Bloggs</n:name>
    </n:passenger>
  </env:Header>
  <env:Body>
    <p:itinerary xmlns:p="http://travelcompany.example.org/reservation/travel">
      <p:departure>
        <p:departing>New York</p:departing>
        <p:arriving>Los Angeles</p:arriving>
        <p:departureDate>2007-12-14</p:departureDate>
        <p:departureTime>late afternoon</p:departureTime>
        <p:seatPreference>aisle</p:seatPreference>
      </p:departure>
      <p:return>
        <p:departing>Los Angeles</p:departing>
        <p:arriving>New York</p:arriving>
        <p:departureDate>2007-12-20</p:departureDate>
        <p:departureTime>mid-morning</p:departureTime>
        <p:seatPreference></p:seatPreference>
      </p:return>
    </p:itinerary>
  </env:Body>
</env:Envelope>

```

Ilustración 6 Mensaje SOAP Aplicacion agencia de viajes

Fuente: https://www.ibm.com/support/knowledgecenter/es/SSMKHH_9.0.0/com.ibm.etools.mft.doc/ac55780_.htm

Los elementos contenidos en el interior de header body están definidos por la aplicación en su propio XML. La cabecera contiene dos bloques, reservation y passenger, con información de control, la fecha y hora de la reserva y su referencia, así como el nombre del pasajero. El cuerpo del mensaje contiene los datos del itinerario en el elemento itinerary, que contiene la información de ida y vuelta del viaje. Este documento XML puede ser generado de manera automática por medio de librerías o frameworks durante la implementación del servicio.

6.4.2 REST.

Representational State Transfer o REST se diferencia de SOAP por su enfoque a recursos. Propone una implementación más simple empleando métodos del protocolo HTTP para comunicarse con un servicio web ya que no define una estructura para representar la información pues esta se puede encontrar en formato XML o notación JSON.

Ejemplo de recursos REST		
Método HTTP	URI	
	<i>/usuarios/</i>	<i>/usuarios/id</i>
GET	Devuelve el listado completo de usuarios de la agenda telefónica. En lugar de devolver los identificadores, un diseño RESTful devolvería las URI de cada uno de estos usuarios.	Devuelve la información del usuario de la agenda telefónica identificado en la URI.
POST	Añade un nuevo usuario. Los datos del mismo irán en el cuerpo de la petición.	A veces se emplea también para modificar los datos (PUT), aunque sería un error de diseño.
PUT	Reemplaza la lista de usuarios con otra completamente nueva enviada en el cuerpo de la petición.	Reemplaza este usuario por uno nuevo enviado en la petición.
DELETE	Elimina todos los usuarios de la agenda.	Elimina este usuario de la agenda.

Tabla 3 Recursos REST Agenda Telefónica

Fuente:

<https://play.google.com/books/reader?id=AZ3gDAAAQBAJ&printsec=frontcover&output=reader&hl=es&pg=GBS.PA33>

En el diseño de servicios REST se tienen en cuenta una serie de restricciones, para considerarlo REST.[12]

- **Cliente-Servidor.** la aplicación debe tener como arquitectura cliente-servidor permitiendo el desarrollo de ambas partes de manera independiente.
- Sin **estado.** El cliente es el encargado de enviar la información necesaria para procesar correctamente una solicitud, el servidor no almacena ninguna información de sesión.

- **Cachable.** la implementación de la aplicación cliente debe permitir almacenar datos en una memoria cache, estos datos se caracterizan por la demanda de solicitudes. La idea no es sobre cargar el servidor con solicitud de información que es constantemente requerida.
- **Interfaz uniforme.** definir estándares que permitan cierta uniformidad en el desarrollo la interfaz de la aplicación, con el fin de identificarlos fácilmente sin necesidad de conocer previamente el servicio.
- **Sistema en capas.** El sistema debe contar con numerosas capas, por ejemplo, lógica, modelo o vista, la idea es que el cliente no identifique con que capa esta interactuando, permitiendo establecer políticas de seguridad y control.

6.5 METODOLOGÍAS PARA EL DESARROLLO DE SISTEMAS DE INFORMACIÓN WEB SOA

SOA trae un nuevo paradigma de programación. Durante el desarrollo debemos identificar un modelo de negocio y cada uno de los procesos (procesos de negocio) que intervienen en él. Seguidamente, como si se tratara de un juego de bloques se descomponen estos procesos en sus piezas más pequeñas, para clasificarlas y detectar patrones. Al final cada pieza se implementa de manera individual como un procedimiento (servicio) independiente con características claras para comunicarse e interactuar con otros procedimientos estructurando nuevamente el proceso. El éxito de este paradigma radica en un buen análisis y diseño del modelo de negocio donde las metodologías de desarrollo tradicionales como XP, SCRUM o RUP no consideran un enfoque de negocio.

Como resultado de estas limitaciones, se han planteado diferentes metodologías de desarrollo orientadas a servicios, algunas son nuevas, otras construidas a partir de metodologías tradicionales orientadas al desarrollo de sistemas centralizados o distribuidos. Como ejemplo, SOMA [11] añade actividades enfocadas a estructurar el modelo de negocio, ajustando la metodología pesada RUP [10] (Rational Unified

Procesos) para SOA. También, Stojanović [12] una propuesta que deriva de una metodología basada en componentes, y finalmente, Sod-M [13] una ligera diseñada específicamente alrededor del concepto de servicio.

Para el desarrollo del proyecto se trabajó con SOMA. La visión SOMA presenta cuatro fases para el desarrollo de servicios, denominadas: Identificación, Especificación, Realización y Despliegue de servicios [11]. Estas fases se realizan dentro de las del RUP, por ej. la identificación y especificación de servicios son principalmente actividades de la fase de Elaboración, y las tareas asociadas se asignan a las Disciplinas del RUP. Requiere la realización de las actividades definidas en las Disciplinas del RUP, agregando Disciplinas y actividades específicas para SOA, así como entregables y roles, también en las Disciplinas de Modelado del Negocio, Diseño e Implementación. La tabla 4 describe brevemente SOMA del RUP.

Disciplina	Actividades	Roles	Entregables
Modelado de Negocio	Identificar objetivos del Negocio/desempeño, para identificar objetivos del negocio y gestionarlo con los indicadores definidos. Refinar Caso de Uso del Negocio para obtener Casos de Uso del Sistema que puedan realizarse como procesos de negocio. Análisis de Área Funcional, para relacionar funciones del negocio con áreas funcionales.	Analista de procesos del negocio	Visión del negocio y objetivos del Negocio. Modelo de Casos de Uso del Negocio. Especificación suplementaria del negocio
Análisis y Diseño	Analizar los procesos del Negocio para identificar servicios, su especificación, dependencias y comunicaciones. Analizar Casos de Uso del Negocio (SOA) identificando servicios candidatos para realizarlos.	Arquitecto de Software Analista de procesos del negocio	Modelo de Análisis del negocio. Modelo de Servicios que contiene toda la

	<p>Analizar activos existentes para identificar posibilidades de reutilización.</p> <p>Aplicar test de filtro sobre los servicios candidatos evitando el “síndrome de proliferación de servicios”.</p> <p>Especificar servicios categorizándolos y definiendo orquestaciones y coreografías entre servicios.</p> <p>Diseñar subsistemas (SOA).</p>	Diseñador e Implementador	información de los servicios.
Implementación	Documentar decisiones de realización de servicios indicando en el Modelo de Servicios, los componentes de software asociados que ejecutaran.	Diseñador, Implementador y Arquitecto de Software	Modelo de Servicios, y otros como el SAD y el Modelo de Diseño

Tabla 4 Disciplinas metodología SOMA

En la tabla anterior se describió cada una de las disciplinas involucradas en esta metodología. Ahora, para identificar la base de dichas disciplinas es importante recalcar de manera general las fases que intervienen e influyen en toda la metodología. A continuación, se exponen las fases la metodología SOMA detallando tareas a seguir:[13]

- **Identificación del servicio.** En esta fase se deben identificar los posibles servicios que serán candidatos a la implementación. Tengamos en cuenta que cada servicio es visto como un proceso de negocio o un activo empresarial que lleva a cabo tareas específicas como reportes, facturación, rutinas de control, entre otros en la organización. Básicamente la identificación de servicio invita a descomponer todos los procesos de la empresa, definir un modelo de servicios y finalmente analizar el estado actual de la misma con el fin de deputar el modelo definido. En esta fase se recomienda como tarea realizar un análisis de proceso empresarial, un modelo de uso empresarial (SOA), análisis del estado actual de la empresa, también señala definir y consolidar los objetivos y reglas empresariales.

- **Especificación de servicios.** Esta fase es de vital importancia en el desarrollo, ya que es aquí donde se deben seleccionar, clasificar y asignar a módulos o subsistemas los servicios candidatos a implementar. En esta fase se recomienda como tareas, aplicar pruebas para la selección de servicios, especificar claramente los servicios y subsistemas, diseñar mensaje y especificar los componentes.
- **Realización del servicio.** Se establece el desarrollo de los servicios a implementar teniendo en cuenta los análisis, los modelos, y los subsistemas estudiados en las fases anteriores. Además, de una clara documentación.

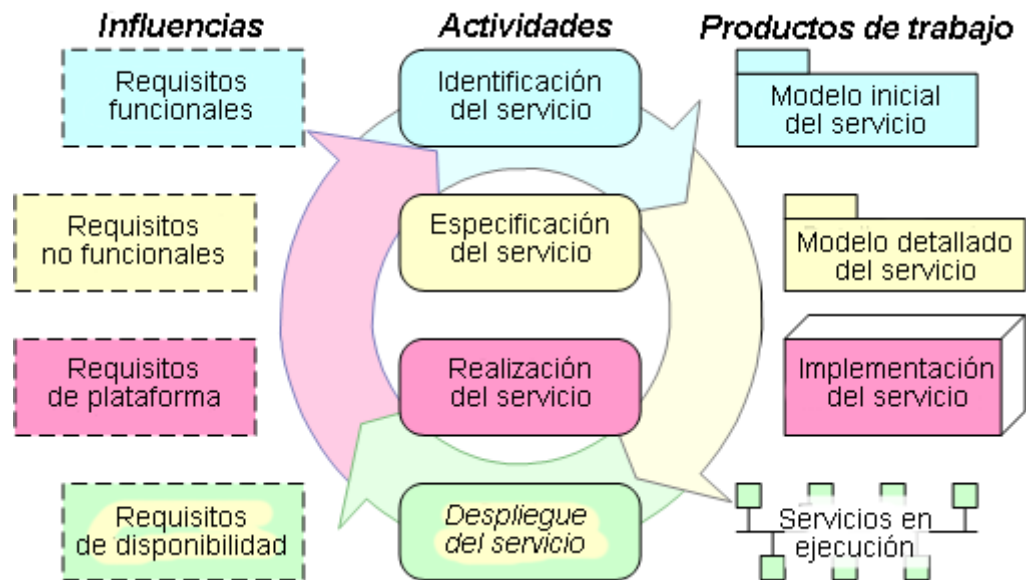


Ilustración 7 Fases clave de la metodología SOMA

Fuente:

http://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/index.htm#soa.rup_soma/guidances/concepts/developing_service-oriented_solutions_38D3FA86.html

6.6 PLATAFORMA DE DESARROLLO JAVA

Aplicaciones Java J2EE

Java Enterprise Edition, define un conjunto de estándares que habilita soluciones para manejo y desarrollo de aplicaciones empresariales multicapa que son centralizadas en el servidor. Su fin principalmente es ser utilizado en desarrollo de aplicaciones empresariales, ya que sus especificaciones y funcionalidades son orientadas a negocios. Sus grandes ventajas son: robustez, fiabilidad, estabilidad y seguridad. El objetivo principal es simplificar las aplicaciones empresariales, basándolas en componentes modulares y estandarizados, para proveer un completo conjunto de servicios[14].

En la ilustración 8 se muestra la arquitectura de las aplicaciones web, el primer nivel que corresponde al cliente y la vista, es donde está la interfaz de usuario y se realiza una iteración con el mismo (en aplicaciones o en páginas web). El segundo nivel que es la lógica, está referido habitualmente a algún tipo de programa o script y es el encargado de realizar todas las validaciones desde los procesos manejados en el sistema, según reglas de negocio establecidas por los stakeholder. El tercer nivel, denominado de persistencia o de objetos de acceso a datos, son los encargados de permitir la interacción con la base de datos a través del mapeo de clases generadas en la capa de modelo.

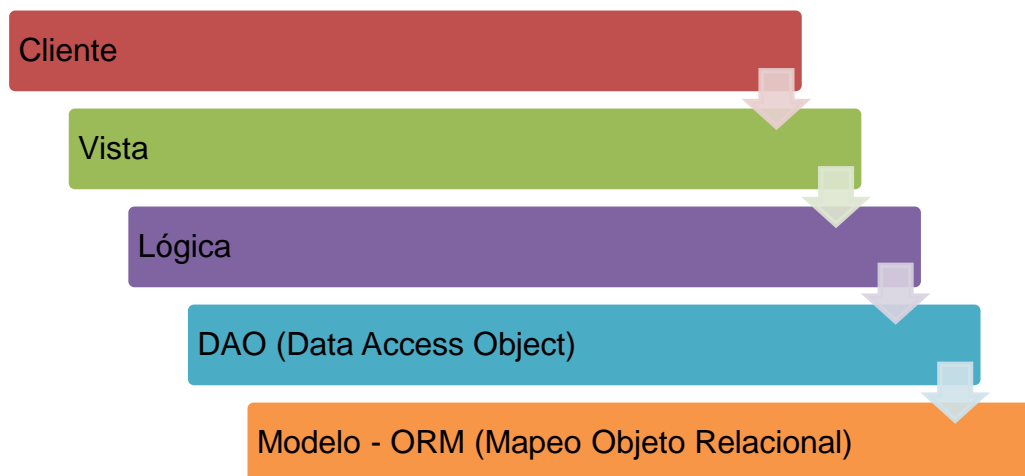


Ilustración 8 Arquitectura de las Aplicaciones Web

Fuente: Elaboración Propia.

6.6.1 ORM (Mapeo Objeto Relacional)

El mapeo objeto-relacional es una técnica de programación para convertir datos del sistema de tipos utilizado en un lenguaje de programación orientado a objetos al utilizado en una base de datos relacional. En la práctica esto crea una base de datos virtual orientada a objetos sobre la base de datos relacional. Esto posibilita el uso de las características propias de la orientación a objetos (esencialmente la herencia y el polimorfismo)[22].

6.6.2 DAO (Data Access Object)

El DAO implementa el mecanismo de acceso requerido para trabajar con la base de datos. Se utiliza para abstraer y la encapsular todos los accesos a la base de datos. El DAO gestiona la conexión con la fuente de datos para obtener y almacenar datos. Esencialmente, la DAO actúa como un adaptador entre el componente y la base de datos[23].

6.6.3 JPA y JPQL

La API de persistencia de Java (JPA) es una solución basada en estándares Java para la persistencia. La persistencia utiliza un enfoque de mapeo objeto / relacional para cerrar la brecha entre un modelo orientado a objetos y una base de datos relacional. La API de persistencia de Java también se puede utilizar en Java SE aplicaciones, fuera del entorno Java EE. Java Persistence consiste en lo siguiente áreas[24]:

- La API Java Persistence
- El lenguaje de consulta (JPQL)
- Objeto / metadatos de mapeo relacional.

6.6.4 EJB

Los Enterprise Java Beans son componentes del lado del servidor que encapsulan la lógica del negocio de una aplicación[25].

6.6.5 Managed Bean

Son objetos gestionados por contenedor de peso ligero (POJO) con requisitos mínimos, para apoyar a un pequeño conjunto de servicios básicos, como la inyección de recursos, las devoluciones de llamada de ciclo de vida, y los interceptores. Estos representan una generalización de los beans gestionados especificados por la tecnología JavaServer Faces y se puede utilizar en cualquier lugar de una aplicación Java EE, no sólo en la Módulos Web [26].

6.6.6 JSF (Java Server Faces)

La tecnología Java Server Faces establece el estándar para la construcción de interfaces de usuario del lado del servidor. Con los aportes de un grupo de expertos, la API Java Server Faces fue diseñados para que puedan ser aprovechados por las herramientas que harán que el desarrollo de aplicaciones web sea incluso más fácil[27].

6.6.7 PrimeFaces

Es una librería de componentes visuales de código abierto para el conjunto Java Server Faces 2.0 desarrollada y mantenida por Prime Technology. Su objetivo principal es ofrecer un conjunto de componentes para facilitar la creación y diseño de aplicaciones web[2].

6.6.8 Servidor de Aplicaciones Glassfish

GlassFish es un servidor de aplicaciones desarrollado por Sun Microsystems destacado por su fiabilidad y estabilidad. Sus principales características son:

- Compatibilidad con las tecnologías de interoperabilidad de Web Services (WSIT) – Sun trabaja estrechamente con Microsoft para garantizar la interoperabilidad de las tecnologías de empresa de los servicios web, como la optimización de mensajes, la mensajería fiable y la seguridad.[28]
- Compatibilidad con JBI: Java Business Integration (JBI) amplía Java EE con Service Provider Interfaces (SPI) de integración empresarial. Estas interfaces (SPI) permiten a los desarrolladores crear o implementar el entorno de integración empresarial Java para especificaciones como WSCI, BPEL4WS y W3C Choreography Working Group.[29]
- Admite la replicación "en memoria" – La replicación "en memoria" en otros servidores proporciona un almacenamiento ligero de los datos del estado de la sesión sin necesidad de obtener una base de datos aparte. Este tipo de duplicación utiliza memoria de otros servidores para el almacenamiento de alta disponibilidad de datos de sesión HTTP y de beans de estado de la sesión.[29]
- Mejoras en el equilibrio de la carga – Se han añadido algunas mejoras en el plugin del equilibrio de la carga.[29]
- Comunidad Open Source y GlassFish : En junio de 2005, Sun estrenó la comunidad GlassFish cuyo propósito es desarrollar un servidor de aplicaciones libre, de código abierto y de nivel comercial que implemente las

funciones más recientes de la plataforma Java EE 5 y de las tecnologías de empresa relacionadas.[29]

6.6.9 MYSQL

El servidor de base de datos MYSQL es muy rápido, fiable y fácil de usar. Este ofrece una serie de características entre ellas, la capacidad de tratar grandes bases de datos mientras registra un alto rendimiento.

Por otro lado, MYSQL un gestor de base de datos de código abierto basado en lenguaje de consulta estructurado (sql) puede ejecutarse en la mayoría de sistemas operativos conocidos como Windows y Linux, también presenta interfaces de programación para acceder a sus bases de datos desde lenguaje de programación c,c+, java, perl, php, python, entre otros. [30]

6.6.10 Servicios Web

Se considera servicio web a la evolución de la tecnología RPC (llamada de procedimiento remotos) que hace uso de la web para diseñar software distribuido e interoperable. RPC es una técnica de programación donde un programa que se ejecuta en una maquina puede realizar llamadas a procedimientos en otras máquinas diferentes como si estos procedimientos estuvieran disponibles en la misma máquina. [12]

Un servicio web debe cumplir con las siguientes características:

- **Visibilidad.** Es la capacidad del servicio de ser conocido por aquellos que desean usarlo.[12]
- **Reusabilidad.** Puede ser usado una y otra vez en diferentes escenarios.
- **Interactividad.** Mediante paso de mensajes el servicio debe interactuar con otros servicios o multitud de clientes. [12]
- **Interoperabilidad.** Su comunicación es estándar. Además, debe mostrar independencia al lenguaje de programación o sistema operativo del cliente y el servidor.[12]

- **Seguridad.** Un servicio debe atender solo las peticiones de aquellos usuarios con las credenciales o privilegios adecuados.[12]

CAPÍTULO 2. DESARROLLO DEL PROYECTO - METODOLOGÍA

En esta sección se presenta el desarrollo del trabajo, a través de la descripción de cada una de las fases de la metodología SOMA. Se presenta también las herramientas usadas para el desarrollo y como se realizó la fase de pruebas e implementación del sistema.

7 DESARROLLO DEL PROYECTO

Una metodología para desarrollo con servicios no implica necesariamente un nuevo proceso, es posible utilizar el de la Organización, agregando elementos para servicios. Teniendo en cuenta la naturaleza cambiante de los requerimientos y el hecho de trabajar con SOA en este trabajo, se utilizó un proceso base adaptación del RUP [31], que prevea entregables intermedios para seguimiento y control del proyecto.

El proceso utilizado, tiene como el RUP dos dimensiones, el tiempo y las disciplinas, manteniendo en la del tiempo sus cuatro fases: Inicial, Elaboración, Construcción y Transición, en las disciplinas: requerimientos, diseño, implementación y verificación; más gestión del proyecto, calidad, configuración e implantación. Además, según [11] también se definieron actividades y entregables de entrada y salida, así como iteraciones y una agenda semanal de actividades y entregables. Los roles que se proponen en [11] no se tuvieron en cuenta, como se expuso también en la tabla 3, dado que el proyecto se desarrolló por una sola persona.

La Extensión SOA en la metodología SOMA, requirió cumplir con el proceso base, agregando elementos específicos para desarrollo SOA. Siendo así, se hizo énfasis en el Modelado del Negocio, Diseño e Implementación para modelar procesos de negocio e identificar servicios para realizarlos, diseñarlos y construirlos. A continuación se describe la metodología usada para los 4 módulos o servicios principales del proyecto como se muestra en la ilustración 6, en donde de aquí en

adelante se referenciarán como MS1-Usuarios para el caso de la gestión de usuarios, MS2-Pacientes para la gestión y consumo de información de pacientes, MS3-Proveedores para los servicios propios de información de proveedores y MS4-Historias para el módulo principal de la aplicación que consume los servicios para el control de información en las historias de consulta externa, audiometría, optometría y salud ocupacional.

En la identificación de requerimientos se dividieron los módulos como servicios independientes y bajo estos, se inicia con la metodología aplicando para cada uno cada una las disciplinas de la metodología como iteraciones que iban generando cada aplicación de manera independiente.

Según las características del sistema, se inició por el módulo de usuarios, posteriormente pacientes y proveedores y finalmente el de historias.

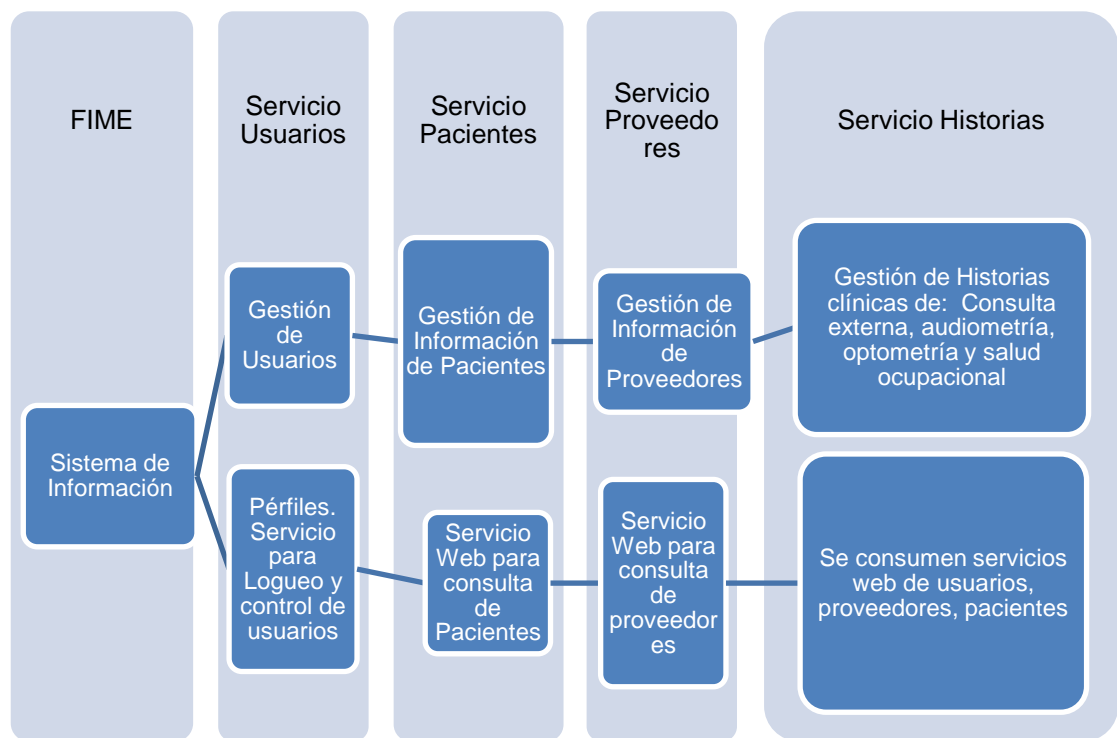


Ilustración 9 Servicios de aplicación web para FIME

7.1 DESARROLLO DE METODOLOGÍA SOMA

7.1.1 Disciplina Modelado de Negocio

En la primera disciplina o fase de la metodología trabajada, se evaluó la organización identificando los requerimientos generales del sistema para luego definir aspectos como detalles de funcionalidades y roles de usuario a través de casos de uso del negocio. Para modelar los procesos de negocio se usó los diagramas de Casos de Uso del Negocio incluyendo notación grafica UML.

La tabla 5 muestra los requerimientos para el MS1-Usuarios.

Ver Anexo 5 Requerimientos para MS2-Pacientes.

Ver Anexo 6 para Requerimientos MS3-Proveedores.

Ver Anexo 7 para Requerimientos MS4-Historias Clínicas.

Requerimientos módulo de usuarios	
Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Autenticación de Usuario.
Características:	Los usuarios deberán identificarse para acceder a cualquier parte del sistema.
Descripción del requerimiento:	El sistema deberá pedir un usuario y contraseña con el fin de validar el usuario y sus permisos. A su vez deberá direccionar al usuario a una ventana acorde a sus privilegios.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF05 • RNF08
Prioridad del requerimiento: Alta	
Identificación del requerimiento:	RF02
Nombre del Requerimiento:	Registrar Usuarios.
Características:	Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del sistema.
Descripción del requerimiento:	El sistema permitirá al Gerente registrar uno o más usuarios. Cada usuario debe suministrar datos como: C.C, Nombre, Apellido, Dirección, Fecha de nacimiento, E-mail, Teléfono, Cargo o Profesión, Usuario y Password.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF05 • RNF08
Prioridad del requerimiento: Alta	
Identificación del requerimiento:	RF03

Nombre del Requerimiento:	Modificar y actualizar usuarios del sistema.
Características:	El sistema permitirá al usuario gerente modificar y actualizar los datos de los usuarios
Descripción del requerimiento:	El sistema ofrecerá al usuario Gerente modificar los usuarios registrados en el sistema, actualizar sus datos personales, contraseñas y privilegios de usuario.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02
Prioridad del requerimiento:	Alta
Identificación del requerimiento:	RF04
Nombre del Requerimiento:	Eliminar usuarios del sistema.
Descripción del requerimiento:	El sistema permitirá al gerente eliminar un perfil de usuario.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02
Prioridad del requerimiento:	Alta
Identificación del requerimiento:	RF05
Nombre del Requerimiento:	Consultar Usuarios.
Características:	El sistema permitirá al administrador consultar la información del registro de cada usuario.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02
Prioridad del requerimiento:	Alta
Identificación del requerimiento:	RF06
Nombre del Requerimiento:	Registro de acciones en el sistema.
Características:	El sistema deberá garantizar una bitácora de acciones de cada usuario.
Descripción del requerimiento:	El sistema proporcionara el registro automático de acciones u operaciones realizadas por acá usuario del sistema, relacionando la fecha y la hora del evento.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF05 • RNF06
Prioridad del requerimiento:	Alta
Identificación del requerimiento:	RF07
Nombre del Requerimiento:	Consultar registro de acciones en el sistema
Características:	El sistema deberá permitir al usuario gerente consultar la bitácora de acciones o eventos.
Descripción del requerimiento:	Se podrá consultar el registro de eventos de todos y cada uno de los usuarios de manera general o individual. También se podrán consultar los eventos por fecha.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF05 • RNF06

Prioridad del requerimiento: Alta	
Identificación del requerimiento:	RF08
Nombre del Requerimiento:	Mantenimiento de usuarios del sistema.
Características:	El módulo de usuario del sistema deberá ofrecer un mecanismo para el mantenimiento de usuarios.
Descripción del requerimiento:	El sistema deberá notificar al gerente sobre los usuarios que llevan más de un mes inactivos (sin ingresar al sistema), y periódicamente solicitar la actualización de datos.
Requerimiento NO funcional:	<ul style="list-style-type: none"> • RNF01 • RNF02 • RNF05 • RNF06
Prioridad del requerimiento: Alta	

Tabla 5 Requerimientos MS1-Usuarios
Elaboración Propia

La ilustración 10 muestra el diagrama de casos de uso para el MS1-Usuarios en el cual se muestra los actores involucrados y las funcionalidades que ofrecerá el módulo.

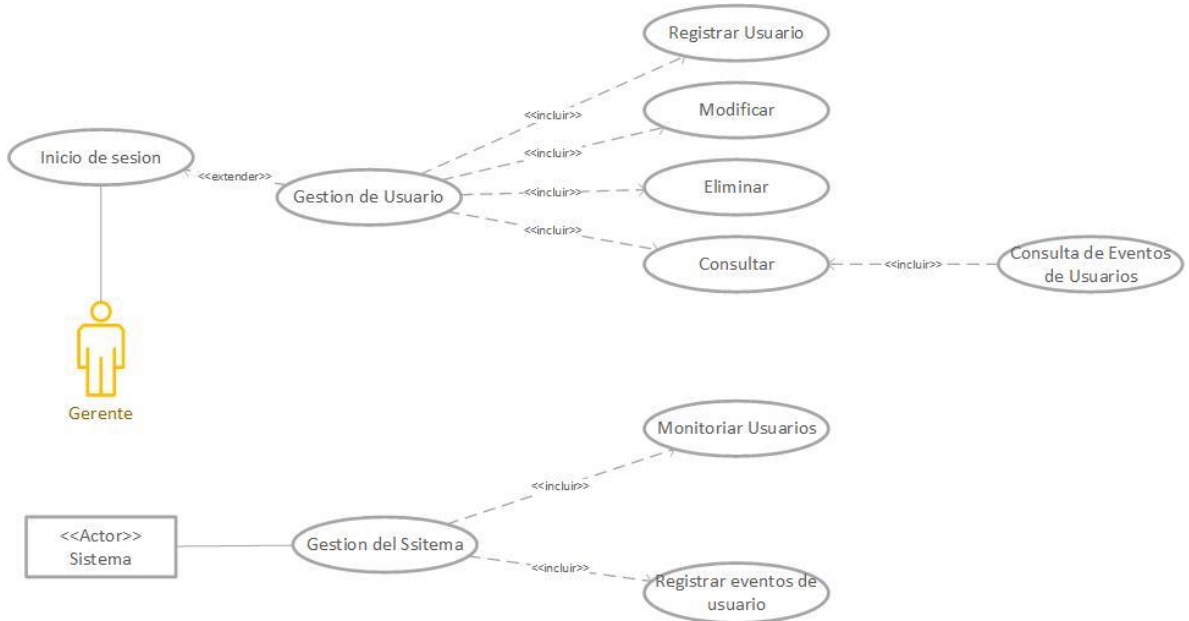


Ilustración 10 Caso de uso MS1-Usuarios

Ver Anexo 8 para casos de uso MS2-Pacientes.

Anexo 9 para casos de uso MS3-Proveedores.

Anexo 10 para MS4-Historias Clínicas.

7.1.2 Disciplina Análisis y Diseño

En la segunda disciplina trabajada, se validaron y categorizaron los servicios para realizar los procesos del negocio y especificar las funcionalidades que iba a permitir cada servicio definiendo los contratos funcionales incluyendo interfaces y métodos. Además, se establecieron las funcionalidades a reutilizar o aquellas que iban a ser consumidas desde los demás módulos.

Los servicios de cada uno de los módulos se muestran en la tabla 6, en donde se indica las funcionalidades que cada uno va a permitir y los módulos que harán uso de los mismos.

Módulo	Servicio Ofrecido	Métodos	Módulo que consume el servicio
MS1-Usuarios	<ul style="list-style-type: none"> Validación de usuario. Registro de usuarios Consulta de usuario/s Actualización de usuario Eliminación de usuarios 	<ul style="list-style-type: none"> find() create() edit() findRange() remove() 	<ul style="list-style-type: none"> MS2-Pacientes MS3-Proveedores MS4-Historias
MS2-Pacientes	<ul style="list-style-type: none"> Registro de pacientes Consulta de paciente/s Actualización de paciente Eliminación de paciente 	<ul style="list-style-type: none"> find() findAll() create() edit() findRange() remove() 	<ul style="list-style-type: none"> MS2-Historias
MS3-Proveedores	<ul style="list-style-type: none"> Registro de proveedores Consulta de proveedor/es Actualización de proveedor 	<ul style="list-style-type: none"> find() findAll() create() edit() findRange() remove() 	<ul style="list-style-type: none"> MS2-Historias

	<ul style="list-style-type: none"> • Eliminación de proveedor 		
MS4-Historias	<ul style="list-style-type: none"> • Registro de Historias 	Créate()	Este servicio dentro de la aplicación solo es consumido por sí mismo.

Tabla 6 Descripción de Servicios

Para la definición de cada una de las funcionalidades de los módulos y características de los servicios, se utilizó BPM, a través del BPMS Bonita Soft. La ilustración 11 muestra el diagrama BPM para el caso del módulo de Pacientes MS2-Pacientes. Ver: Anexo 11 para diagramas BPM MS1-Usuarios. Anexo 12 para diagrama BPM MS3-Proveedores. Anexo 13 para diagrama BPM MS4-Historias.

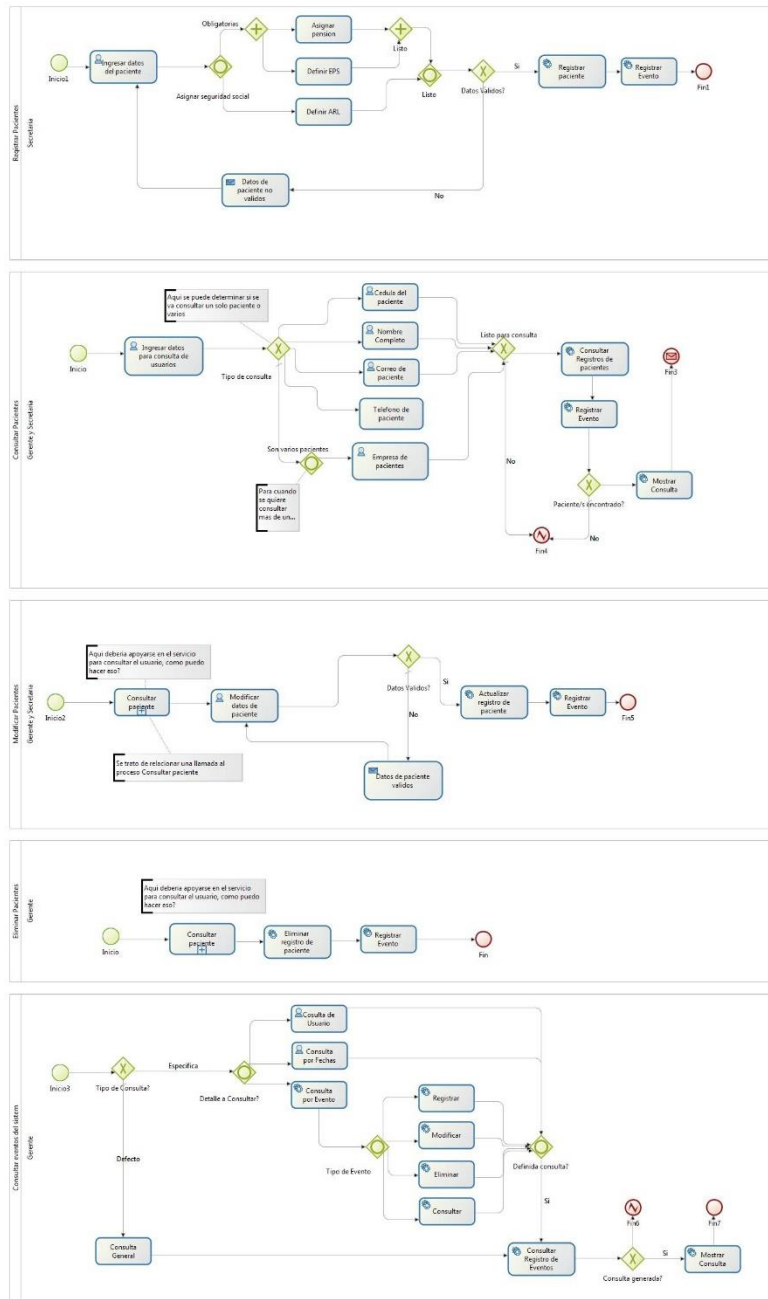


Ilustración 11 Diagrama BMP MS1 Pacientes

Finalmente, en esta disciplina se define el modelo de base de datos de cada componente de la aplicación. La ilustración 12 muestra el modelo de base de datos

para MS4-Historias, el cual consumirá los servicios que ofrecen MS1-Usuarios, MS2-Pacientes y MS3-Proveedores.

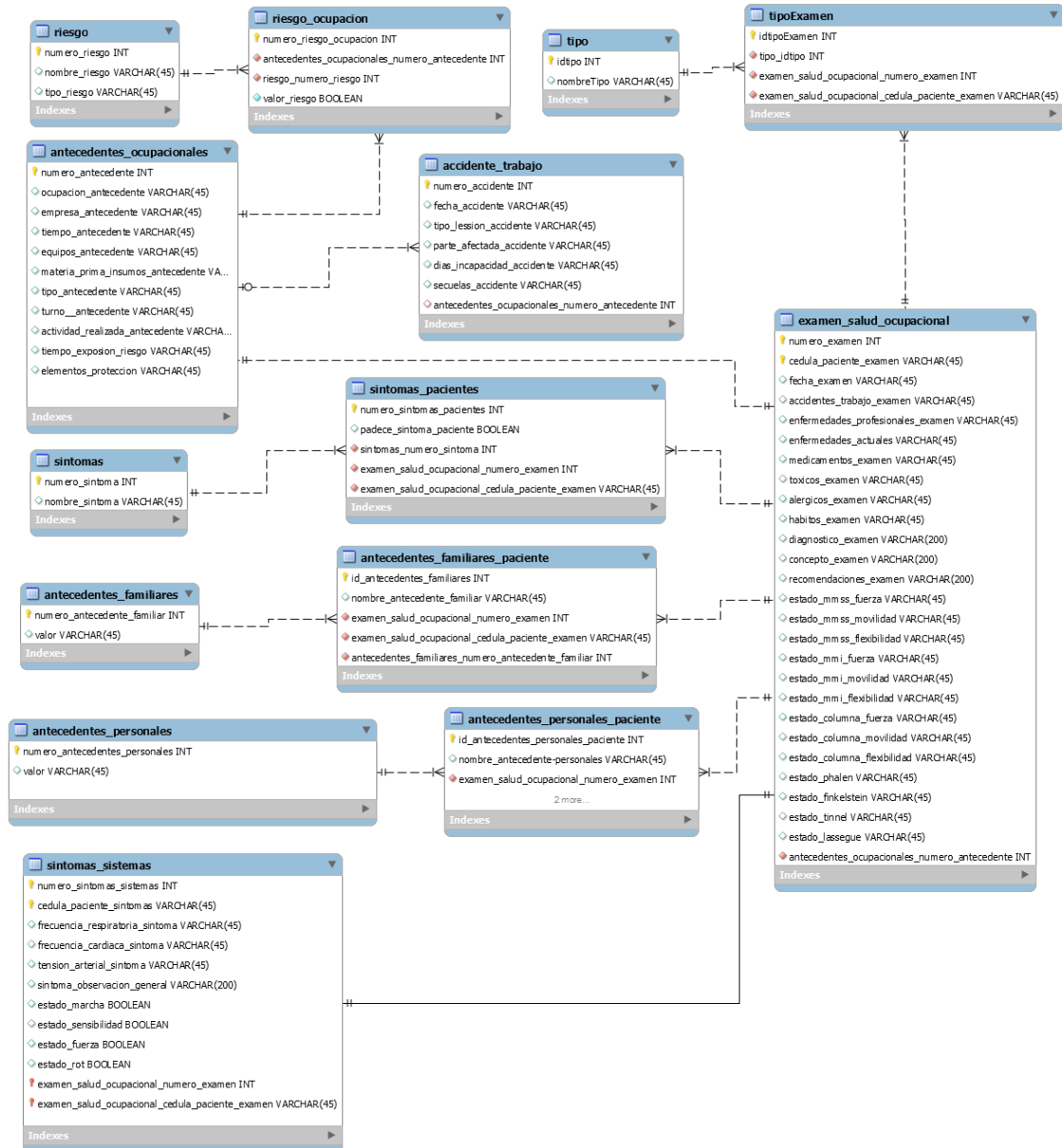


Ilustración 12 Base de Datos MS1-Usuarios

Ver:

Anexo 14 para modelo de base de datos MS1-Usuarios,

Anexo 15 para modelo de base de datos MS2-Pacientes.

Anexo 16 para MS3-Proveedores.

Anexo 17 para MS4-Historias Clinicas.

7.1.3 Disciplina Implementación

En la fase o disciplina de implementación, se codificó cada uno de los módulos de manera independiente y se crearon los servicios web para usuarios, pacientes y proveedores, teniendo en cuenta las funcionalidades de cada servicio, sus interfaces y la interacción entre cada uno de ellos.

Finalmente, para cada módulo y servicio se establecieron las pruebas de aceptación. Estas pruebas se realizaron al final de cada módulo para verificar que se cumpliera con los requerimientos de los usuarios. Las pruebas de aceptación que fallarán fueron analizadas para evaluar su corrección. La tabla 7 muestra la plantilla usada de pruebas, en donde se establecen elementos como las condiciones de ejecución y los pasos para la misma, identificados el resultado esperado y si fue positiva o no. El Anexo 18 Pruebas muestra el total de pruebas realizadas en cada módulo.

Prueba funcional	
Número de prueba: 1	Requerimiento: RF05 – Consulta de pacientes
Nombre de la prueba: Consulta de Paciente desde módulo MS4-Historias.	
Descripción: En el módulo MS4-Historias se ingresa el número de cedula del paciente. Una vez se cuente con el número se procede a consultar la información del paciente consumiendo el servicio consultar paciente del módulo MS2-Pacientes.	
Condiciones de ejecución: Red local	
Entrada/Pasos de ejecución: <ul style="list-style-type: none">• Entrada: Numero de cedula del paciente.	

<ul style="list-style-type: none"> Proceso: Consultar paciente por número de cedula.
<p>Resultado esperado: Si la consulta es válida debe cargar la información del paciente en cada uno de los campos de textos de la interfaz de historia clínica, en caso contrario, deberá aparecer un mensaje indicando que la consulta no se realizó.</p>
<p>Evaluación de la prueba: Positiva.</p>

Tabla 7 Prueba funcional RF05-MS4 Historias

7.2 HERRAMIENTAS USADAS

La ilustración 13 muestra el conjunto de tecnologías de Java EE que se utilizaron para el desarrollo del sistema de información. Todas estas tecnologías serán descritas a continuación.

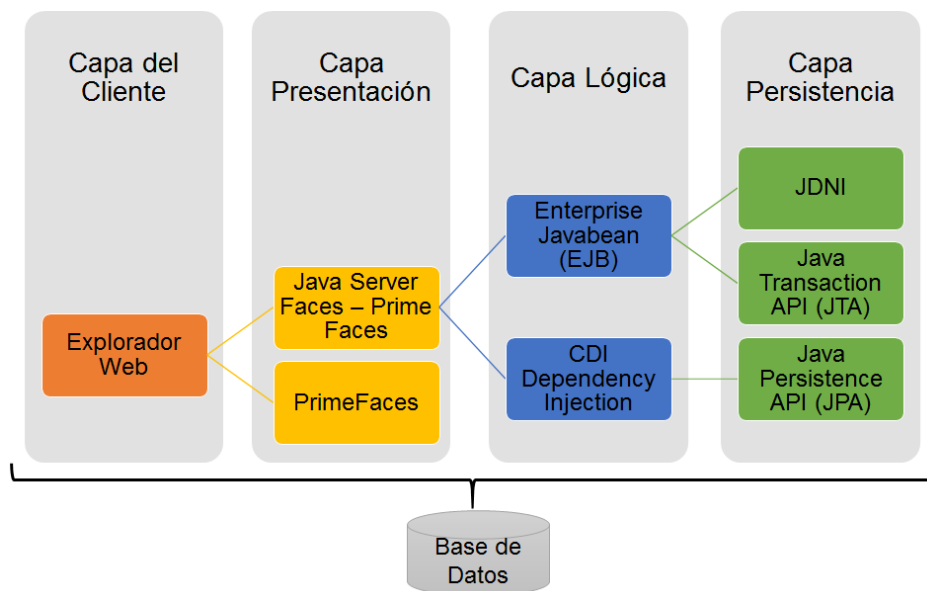


Ilustración 13 Conjunto de Tecnologías Java EE

Fuente: Elaboración Propia.

En primer lugar, desde la capa del cliente y capa presentación, se hizo uso de JSF que es un estándar de Java hacia la construcción de interfaces de usuario para aplicaciones web que simplifican el desarrollo de aplicaciones web del lado del cliente. En las páginas JSF se utilizó la extensión *.xhtml, es decir, una combinación

de XML con HTML, en donde se pudo incluir componentes como CSS, JavaScript, entre otros.

La especificación de JSF define seis fases distintas en su ciclo de vida:

1. **Restauración de la vista:** Crea un árbol de componentes en el servidor para representar la información de un cliente.
2. **Aplicar valores de la petición:** Actualiza los valores del servidor con datos del cliente.
3. **Proceso de validación:** Valida los datos del usuario y hace la conversión.
4. **Actualización de valores del modelo:** Actualiza el modelo del servidor con nuevos datos.
5. **Invocar la aplicación:** Ejecutar cualquier lógica de aplicación para cumplir con la solicitud.
6. **Procesar la respuesta:** Guarda un estado y da una respuesta al cliente.

Para facilitar el uso de los componentes se hizo uso de PrimeFaces que es una librería de componentes visuales de código abierto para el conjunto Java Server Faces 2.0 desarrollada y mantenida por Prime Technology. Su objetivo principal es ofrecer un conjunto de componentes para facilitar la creación y diseño de aplicaciones web [32]. Los componentes de PrimeFaces fueron conectados con el servidor a través de los ManagedBean que no son más que beans que reciben la información del html para enviarlos a la capa de lógica.

En segundo lugar, para la capa de lógica, se hizo uso de los JavaBeans empresariales (Enterprise JavaBeans [1, 6], EJB), que son una tecnología (API) que forma parte del estándar de Java EE. Están diseñados para desarrollo y despliegue de aplicaciones (distribuidas) de negocio basadas en componentes del lado del servidor. Una vez que se desarrolla una aplicación, ésta puede ser desplegada en cualquier servidor que soporte la especificación de EJB [2].

En tercer lugar, para la capa de persistencia se hizo uso de JPA que es, una abstracción que está por encima de JDBC lo que permitió ser independiente de SQL. Todas las clases y anotaciones de esta API se encuentran en el paquete `javax.persistence`.

Los principales componentes de JPA son:

- Mapeo de base de datos relacionales (ORM). Es el mecanismo para mapear objetos a los datos almacenados en una base de datos relacional.
- Un API administrador de entidad para realizar operaciones en la base de datos tales como crear, leer, actualizar, eliminar (CRUD).
- El Java Persistence Query Language (JPQL) que permite recuperar datos con un lenguaje de consultas orientado a objetos.
- Las transacciones y mecanismos de bloqueo cuando se accedan a los datos concurrentemente, la API Java Transaction (JTA).

El motor de base de datos usado fue MYSQL. Que sin dudas es uno de los mejores gestores en el mercado, destacando características como: su portabilidad para ejecutarse en diferentes sistemas operativos, API disponibles en los lenguajes de programación más comunes como c, c++, java, php, python o ruby, escalabilidad permitiendo base de datos robustas, seguridad y autenticación de usuarios.

La implementación de los servicios web durante el proyecto se enfocó en la comunicación cliente/servidor para realizar las tareas de persistencia en cada módulo. En ese sentido, un módulo es capaz de acceder a la base de datos de otro módulo de manera transparente, como si se tratara de su propia base de datos.

La implementación de los servicios permite consumir las operaciones CRUD como recursos, es decir, los métodos para insertar, modificar, consultar o eliminar se diseñaron para ser consumidos por el mismo u otros módulos a través del protocolo HTTP. Entre los servicios implementados podemos mencionar el servicio `findAll()` del módulo usuario, el cual permite acceder a la base de datos que contiene

almacenados los usuarios del sistema y consultar sus registros para su posterior procesamiento. Con este servicio se realizó el diseño e implementación del control de usuarios y privilegios de la aplicación facilitando que un usuario puede autenticarse en el sistema con un nivel de acceso determinado.

Todos los módulos de la aplicación se pueden comunicar entre sí, pero teniendo en cuenta los requerimientos (ver Tabla 8 Requerimientos MS1-Usuarios) se establecen reglas de funcionamiento que limitan el consumo de estos servicios, por ejemplo, el módulo Usuario es totalmente independiente con respecto a los demás módulos, mientras que el módulo de historias clínicas debe relacionarse con otros módulos. En la siguiente gráfica se ilustra la relación de cada módulo de la aplicación con respecto al consumo de los servicios implementados.

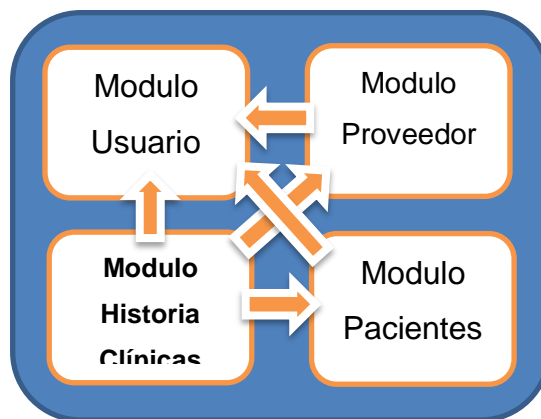


Ilustración 14 Relación de módulos con respecto al consumo de servicios

Fuente: elaboración propia

7.3 IMPLEMENTACIÓN DEL PROYECTO

Finalizados los módulos y servicios en la metodología, el siguiente paso fue la implementación del proyecto. Para ello, se hizo uso de un servidor el cual tenía instalado el motor de base de datos MYSQL y el servidor Glassfish. En las siguientes gráficas podemos observar pantallazos de la aplicación.

Fundacion IPS para la salud fisica y mental Bienvenido, Administrador

Menu Gestion de Usuarios

Agregar o actualizar
Perfiles de usuario

Datos de Usuario

Cedula * + - ✓ ✕

Nombre Apellido

Direccion Telefono

Celular Fecha de Nacimiento

Email Profesion

Cuenta de Usuario

Usuario

Contraseña

Verificar Contraseña

Perfil: GER ✓

Datos de Usuario

Buscar usuario: Ingrese un criterio de busqueda

(1 of 1) 1 5

Cedula	Nombre	Apellido	Direccion	Telefono	Celular	Correo	Profesion	
8419324	Medico 1	654	64	64	64	646	64646	✓
646546	Secretaria 1	464	64	64	64	13	313	✓
1234	Administrador	dasdasd	sadas	dsadas	dasd	asdasdas	dasd	✓

Trabajo de grado - Jose Manuel Conde

Ilustración 15 Captura de pantalla Gestion de Usuarios

Fundacion IPS para la salud fisica y mental Bienvenido, Secretaria 1

Menu Gestion de Proveedores

Volver

Datos de la empresa

Nit + - ✓ ✕

Razon social Nombre Soporte

Telefono Email Direccion

Descripcion de actividad

Listado de Empresas

Buscar empresa: Ingrese un criterio de busqueda

(1 of 1) 1 5

Nit	Razon Social	Direccion	Telefono	Email
1114819324	Reyman Pc	cra 13 #9-45	3177498512	revymanpc@hotmail.com

(1 of 1) 1 5

Ilustración 16 Captura de pantalla Gestión de proveedores, registro de empresas

Ilustración 17 Captura de patanlla Gestion de pacientes

Los pasos para montar la aplicación se muestran en el Anexo. Se utilizó el servidor de aplicaciones Glassfish que implementa las tecnologías definidas en la plataforma Java EE y permite ejecutar aplicaciones que siguen esta especificación.

CAPÍTULO 3 – RESULTADOS Y ANÁLISIS

En esta sección se presentan los resultados obtenidos por cada uno de los objetivos.

8 RESULTADOS

En la tabla 8 se muestra por cada objetivo los resultados obtenidos.

Objetivo	Resultado Obtenidos
Modelar los procesos de gestión de usuarios, pacientes, proveedores, historias clínicas y reportes.	Visión del negocio y objetivos del negocio. Requerimientos funcionales y no funcionales Casos de uso
Especificar los servicios definiendo los contratos funcionales en el desarrollo de la aplicación web orientada a servicios.	Diagramas BMP
Implementar módulo para la gestión de usuarios y privilegios en el sistema, pacientes y proveedores de servicio de la IPS.	Código fuente del módulo y servicios implementados
Implementar módulo de historia clínica que permita la gestión de información de las áreas de salud ocupacional, psicología, audiometría, optometría y medicina general.	Código fuente del módulo y servicios implementados.

Tabla 8 Objetivo vs Resultados Obtenidos

Fuente: Elaboración Propia.

Los resultados del proyecto fueron medidos desde dos puntos de vista, el primero es en cuanto a la mejora en la ejecución de los procesos en la fundación IPS FIME a partir de la implementación del proyecto. Y el segundo, desde el aporte del uso de una arquitectura orientada a servicios en el desarrollo de sistemas de información.

8.1 RESULTADOS IMPLEMENTACIÓN PROYECTO EN LA FUNDACIÓN IPS FIME

Al implementar esta aplicación basada en la arquitectura orientada a servicios se expone a la Fundación IPS FIME a un desarrollo de vanguardia y competitivo a nivel empresarial, esta arquitectura ha creado una tendencia en el desarrollo de este tipo de aplicaciones web debido a que no se limita solo al desarrollo si no que se extiende al modelo y los procesos de negocio de la empresa.

Durante el análisis y diseño de la aplicación se estudió el modelo y los procesos de negocio que se llevaban a cabo en la IPS, en dicho estudio se detectaron fallas en procesos como el registro de los pacientes, datos importantes del personal (usuarios) de la IPS, consulta de proveedores, almacenamiento de documentos clasificados, entre otros. Una vez detectadas esas fallas se diseñaron e implementaron procesos para complementar el modelo de negocio de la IPS. La atención médica.

Por lo tanto, al finalizar este proyecto se optimizaron procesos como el registro, actualización, consulta y eliminación de información, con criterios estandarizados de la propia IPS. Por ejemplo; los usuarios ya cuentan con un nivel de acceso que les permite acceder únicamente a la información autorizada. Entre los resultados obtenidos podemos listar:

- Una disminución a la mitad del tiempo la consulta de información.
- Una reducción de la redundancia en los datos registrados.
- Una transición de formatos físicos a formatos digitales y bases de datos relacionales.
- Un conjunto de restricciones como políticas de seguridad informáticas para almacenar información confidencial.

8.2 RESULTADOS USO DE ARQUITECTURA ORIENTADA A SERVICIOS SOA

El desarrollo de esta aplicación basada en la arquitectura orientada a servicios (soa) presenta un paradigma de programación. La mayoría de aplicaciones se caracteriza por base de datos centralizadas, por módulos dependientes o una estructura jerárquica que obliga a generar líneas de códigos redundantes, en cambio SOA se define en todo el sentido opuesto, exigiendo bases de datos distribuidas, módulos independientes, reutilización de código, protocolos de seguridad y el análisis del modelo de negocio de la organización.

En otras palabras, con SOA se estableció la independencia de todos sus componentes. Y se definieron fuerte criterios de comunicación empleando tecnologías como SOAP y REST para que estos se puedan comunicar y llevar a cabo procesos de negocio. La comunicación es un punto clave dentro de esta arquitectura.

Durante el desarrollo de este proyecto se tuvo en cuenta la tecnología REST para diseñar, implementar y principalmente, comunicar los servicios web de la aplicación, ya que esta trabaja sobre el protocolo más conocido de internet, HTTP. Básicamente, el resultado de esta decisión resalto la facilidad con la que un módulo o servicio establece comunicación con otro por medio de la red sin configuraciones adicionales. Por otro lado, separa realmente el cliente del servidor, permitiendo establecer capas de seguridad o persistencia adicionales en la aplicación.

CAPÍTULO 4 – CONCLUSIONES Y TRABAJOS FUTUROS

En esta sección se describen las conclusiones del trabajo realizado y se plantean algunos trabajos futuros que pueden ser abordados.

13 CONCLUSIONES

Analizar el modelo de negocio de la fundación IPS FIME, desglosarlo en actividades o procesos de negocio permitio tener una visión clara de la organización durante el desarrollo del proyecto. Esta visión no se solo se extiende a nivel de software, si no de administración.

El desarrollo de aplicaciones orientadas a servicios es un claro ejemplo de los modelos de negocio actuales. La evolución de estos, ha sido tal, que ya no se centran solamente en ofrecer productos si no, en la calidad y mejoramiento de un servicio.

Identificar y diseñar un servicio, permite explorar de cerca los procesos que se llevan a cabo en una empresa.

La arquitectura SOA no solamente implica un desarrollo de software, también incluye un restructuramiento a nivel organizacional de la empresa, ya que el objetivo principal de esta arquitectura es alinear los procesos de la empresa con las políticas de la organización.

Ciertamente SOA no es un paradigma de desarrollo nuevo, pero es una tendencia que actualmente suma más seguidores debido a su flexibilidad para integrarse con nuevos procesos y modificar los ya existentes.

El concepto de aplicación de escritorio se puede definir obsoleto, frente a las ventajas de mantenimiento y actualización que plantea una aplicación orientada a servicios. El mantenimiento de una aplicación de escritorio requiere de todos los componentes de la misma para actualizarse, mientras que, una aplicación SOA puede experimentar rutinas de mantenimiento en uno o más módulos sin acaparar toda la aplicación ya que su diseño lo permite.

14 TRABAJOS FUTUROS

Durante el modelado de negocio se encontraron procesos que no estaban contemplados en los objetivos, evidenciando que si tomamos como partida este desarrollo los alcances del proyecto serán mucho más grandes.

En otras palabras, al concluir este proyecto se podría retomar en un futuro cercano e implementar:

- Módulo de facturación.
- Módulo de contabilidad.
- Módulo de asignación de citas.
- Módulo de gestión documental.
- Una interfaz para plataformas móviles.
- Citas online.

Las implementaciones nombradas anteriormente serian viables y factibles gracias a la arquitectura y metodología de desarrollo que se aplicó en este proyecto. Ya que es una arquitectura que agrupó los servicios en módulos, donde muchos de estos servicios podrán ser reutilizados en el desarrollo de nuevos módulos.

CAPÍTULO 6 – REFERENCIAS BIBLIOGRÁFICAS

En esta sección se presenta una lista de referencias bibliográficas que soportan este trabajo.

15 REFERENCIAS BIBLIOGRÁFICAS

- [1] I. Mendoza Vázquez, “Definición De Un Framework Para Aplicaciones Web Con Navegación Sensible a Concerns.”
- [2] F. Pech-May, M. A. Gomez-Rodriguez, L. A. de la Cruz-Diaz, and S. U. Lara-Jeronimo, “Desarrollo de Aplicaciones web con JPA, EJB, JSF y PrimeFaces,” p. 9.
- [3] D. S. Manchado and D. F. Puntos, “Estudio del servidor de aplicaciones Glassfish y de las aplicaciones J2EE,” 2010.
- [4] “Usuario (Informática) - EcuRed.” [Online]. Available: [https://www.ecured.cu/Usuario_\(Informática\)](https://www.ecured.cu/Usuario_(Informática)). [Accessed: 16-Oct-2017].
- [5] O. M. de la Salud, “Marco Conceptual de la Clasificación Internacional para la Seguridad del Paciente,” 2009.
- [6] Ministerio de Salud, *Resolución numero 1995*. 1999.
- [7] “Audiometría - Medilaboral.” [Online]. Available: <http://www.medilaboral.com/audiometria/>. [Accessed: 16-Oct-2017].
- [8] “Optometría - Medilaboral.” [Online]. Available: <http://www.medilaboral.com/optometria/>. [Accessed: 16-Oct-2017].
- [9] IBM, “IBM developerWorks en español : Introducción a SOA y servicios web.” [Online]. Available: <https://www.ibm.com/developerworks/ssa/webservices/newto/service.html>. [Accessed: 16-Oct-2017].
- [10] R. De Colombia and R. Contributivo, “Sistema de Seguridad Social en Salud

Ministerio de la Protección Social Sistema de Seguridad Social en Salud.”

- [11] A. Delgado and F. Ruiz, “Metodologías de desarrollo para Service Oriented Architectures con Rational Unified Process,” *Business*.
- [12] J. L. Berenguel Gómez, *Desarrollo de aplicaciones web distribuidas UF1846*. Paraninfo, 2016.
- [13] “Classic RUP for SOMA.es.” .
- [14] J. A. Rodríguez, J. González Moreno, and T. Murillas Herrera, “Software basado en arquitectura de Java EE y estándar SOAP para la validación en línea de listas de sancionados,” *J. Technol.*, no. 1, pp. 21–26, 2008.
- [15] M. Chico Fernández *et al.*, “Seguridad del paciente y sistemas de comunicación de incidentes,” *Rev. Calid. Asist.*, vol. 20, no. 4, pp. 228–234, 2014.
- [16] A. Ca’Zorzi, “Las TIC en el desarrollo de la PyME: Algunas experiencias de América Latina,” *Cent. Int. Investig. para el Desarro. (IDRC)- BID*, pp. 1–92, 2011.
- [17] E. S. Balado, *Estrategia para la implantación de nuevas tecnologías en PYMES: obtenga el máximo rendimiento aplicando las TIC en el ámbito empresarial*. 2005.
- [18] Escuela Neijing *et al.*, *Introducción a la gestión de sistemas de información en la empresa*, vol. 2010. 2011.
- [19] “Message Queuing (MSMQ).” .
- [20] “Web Application Description Language.” .
- [21] M. A. Chaves, “Gestión de Procesos de Negocios en Función de una Arquitectura Orientada a Servicios,” no. 1, pp. 297–312, 2010.
- [22] H. del Busto and O. Enriquez, “Mapeo Objeto/Relacional (ORM).,” *Rev. Telem@tica*, vol. 10, no. 3, pp. 1–7, 2012.

- [23] O. Corporation, "Core J2EE Patterns - Data Access Object." .
- [24] O. Corporation, "The Java EE 6 Tutorial," no. January, 2013.
- [25] H. del Busto and O. Enriquez, "Mapeo Objeto/Relacional (ORM).," *Rev. Telem@tica*, vol. 10, no. 3, pp. 1–7, 2012.
- [26] O. Corporation, "The Java EE 6 Tutorial," no. January, 2013.
- [27] O. Corporation, "JavaServer Faces Technology."
- [28] Oracle, "Funciones de Sun GlassFish Enterprise Server." [Online]. Available: <https://docs.oracle.com/cd/E19879-01/821-1040/gidij/index.html>.
- [29] Oracle, "Funciones de Sun GlassFish Enterprise Server." .
- [30] "MySQL 5.0 Reference Manual."
- [31] J. Chavarriaga, H. Arboleda, and G. Lidis, "Modelo de Investigación en Ingeniería del Software: Una propuesta de investigación tecnológica," *Ing. del Softw. y Sist.*, 2004.
- [32] "Prime Faces." [Online]. Available: <http://primefaces.org/>.