



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMÁTICA Y ELECTRONICA**  
**ESCUELA DE INGENIERÍA EN SISTEMAS**

**“DESARROLLO DE UN SISTEMA E-SALUD OCUPACIONAL  
MEDIANTE MEAN STACK JAVA SCRIPT EN LA EMPRESA  
CEMENTO CHIMBORAZO”**

Trabajo de titulación presentado para optar al grado académico de:

**INGENIERO EN SISTEMAS INFORMÁTICOS**

**AUTOR: JAIRO ANDRÉS GONZÁLEZ NARANJO**

**EDISON FERNANDO GARCIA GUANGA**

**TUTOR: ING. PATRICIO MORENO.**

Riobamba-Ecuador

2016

©2015, Jairo Andrés González Naranjo; Edison Fernando García Guanga

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor

**ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO**  
**FACULTAD DE INFORMATICA Y ELECTRONICA**  
**ESCUELA DE INGENIERIA EN SISTEMAS**

El Tribunal del trabajo de titulación certifica que: El proyecto técnico: “DESARROLLO DE UN SISTEMA DE E-SALUD OCUPACIONAL MEDIANTE MEAN STACK JAVASCRIPT EN LA EMPRESA CEMENTO CHIMBORAZO”, de responsabilidad del señor Jairo Andrés González Naranjo; Edison Fernando García Guanga, ha sido minuciosamente revisado por los Miembros del Tribunal del trabajo de titulación, quedando autorizada su presentación.

Ing. Washington Luna

\_\_\_\_\_

**DECANO DE LA FACULTAD  
DE INFORMATICA Y ELECTRONICA**

Ing. Patricio Moreno.

\_\_\_\_\_

**DIRECTOR DE LA ESCUELA DE  
INGENIERIA EN SISTEMAS**

Ing. Patricio Moreno.

\_\_\_\_\_

**DIRECTOR DEL TRABAJO  
DE TITULACION**

Ing. Eduardo Villa

\_\_\_\_\_

**MIEMBRO DEL TRIBUNAL**

“Nosotros, Jairo Andrés González Naranjo Y Edison Fernando García Guanga; somos responsables de las ideas, doctrinas y resultados expuestos en este Trabajo de titulación y el patrimonio intelectual del mismo pertenecen a la Escuela Superior Politécnica De Chimborazo”

JAIRO ANDRÉS GONZÁLEZ NARANJO

EDISON FERNANDO GACÍA GUANGA

## **DEDICATORIA**

El presente es dedicado a mi familia, mis amigos y todas las personas que han contribuido con mi proceso de formación académica y personal durante este camino de aprendizaje, por medio de experiencias de vida, dándome una visión y un plan de vida para cada etapa, encaminándome al futuro cumpliendo un objetivo a la vez, para contribuir con un granito de arena a la sociedad con reciprocidad por los conocimientos adquiridos.

*Jairo*

En especial un agradecimiento a mi madre, que es la razón por la que mis objetivos y metas se han hecho realidad, su apoyo incondicional siendo un reflejo de su formación, a mis hermanos por su amor y cariño al siempre darme su mano ante difíciles situaciones que la vida presenta como obstáculos para aprender por medio de la experiencia, y mejorar en mi desempeño diario.

*Edison*

## **AGRADECIMIENTO**

Un cálido agradecimiento a la Institución que hizo posible toda nuestra formación profesional académica como es la Escuela Superior Politécnica de Chimborazo que con su personal docente nos inculco los valores y conocimientos necesarios para desenvolvemos en el ámbito profesional, al Ing. Jorge Delgado quien fue un apoyo con la parte logística y previendo los recursos necesarios en la parte administrativa para que el presente proyecto sea factible, Ing. Patricio Moreno por su apoyo incondicional en el acompañamiento de nuestra formación así como del desarrollo del proyecto, a la Dra. Blanca Maygualema por darnos las guías medicas dentro del campo de salud ocupacional para plasmar toda la teoría en el presente sistema desarrollado ,A la Unión Cementera Nacional (Empresa Cemento Chimborazo) por la apertura para el desarrollo e implementación del sistema, y a todas las personas que estuvieron apoyándonos durante el desarrollo del proyecto asesorándonos en cada etapa.

Edison, Jairo

## **CONTENIDO GENERAL**

PORTADA

CERTIFICACIÓN

DECLARACIÓN DE RESPONSABILIDAD

DEDICATORIA

TABLA DE CONTENIDOS

INDICE DE TABLAS

INDICE DE ILUSTRACIONES

RESUMEN

SUMMARY

INTRODUCCIÓN.....1

### **CAPITULO I**

#### **MARCO TEORICO REFERENCIAL**

**1.1**      **MEAN Stack JavaScript..... 9**

**1.1.1**    **Mongo DB ..... 9**

<b>2.1.2</b>	<b><i>Angular.js</i></b> .....	<b>13</b>
<b>2.1.3</b>	<b><i>Node.js</i></b> .....	<b>15</b>
<b>2.1.4</b>	<b><i>Express.js</i></b> .....	<b>19</b>
<b>1.2</b>	<b>Apis basadas en arquitectura SOA</b> .....	<b>22</b>
<b>1.2.1</b>	<b><i>Definiciones y tipos de servicios web</i></b> .....	<b>22</b>
<b>1.2.1</b>	<b><i>Consumo de servicios desde Angular.js</i></b> .....	<b>24</b>
<b>1.3</b>	<b>Estándar Clasificación Internacional de Enfermedades CIE 10</b> .....	<b>25</b>

## **CAPITULO II**

### **MARCO METODOLÓGICO**

<b>2.1</b>	<b>Metodología Scrum</b> .....	<b>26</b>
<b>2.1.1</b>	<b><i>Roles de la metodología Scrum</i></b> .....	<b>26</b>
<b>2.2</b>	<b>Método Deductivo</b> .....	<b>27</b>

## CAPITULO III

### MARCO DE RESULTADOS

<b>3.1</b>	<b>Análisis de requerimientos .....</b>	<b>29</b>
<b>3.1.1</b>	<b><i>Requerimientos funcionales.....</i></b>	<b>29</b>
<b>3.1.2</b>	<b><i>Requerimientos no funcionales.....</i></b>	<b>30</b>
<b>3.2</b>	<b>Diseño .....</b>	<b>31</b>
<b>3.2.1</b>	<b><i>Diagrama de procesos.....</i></b>	<b>31</b>
<b>3.2.1</b>	<b><i>Diseño de la Arquitectura.....</i></b>	<b>32</b>
<b>3.2.2</b>	<b><i>Diseño de la base de datos.....</i></b>	<b>33</b>
<b>3.2.3</b>	<b><i>Diseño de estándares de interfaz de usuario. ....</i></b>	<b>33</b>
<b>3.2.3</b>	<b><i>Diagrama de Secuencia.....</i></b>	<b>35</b>
<b>3.3</b>	<b>Desarrollo .....</b>	<b>37</b>
<b>3.3.1</b>	<b><i>Tareas de la metodología Scrum: .....</i></b>	<b>37</b>

<b>3.4</b>	<b>Implantación.....</b>	<b>40</b>
<b>3.5</b>	<b>Pruebas.....</b>	<b>40</b>
<b>3.5.1</b>	<b><i>Pruebas Unitarias</i>.....</b>	<b>41</b>
<b>3.5.2</b>	<b><i>Pruebas Rendimiento</i>.....</b>	<b>41</b>
<b>3.5.3</b>	<b><i>Pruebas Usabilidad</i>.....</b>	<b>43</b>
	<b>CONCLUSIONES.....</b>	<b>46</b>
	<b>RECOMENDACIONES.....</b>	<b>47</b>

**BILIOGRAFIA**

**ANEXOS**

## **INDICE DE ABREVIATURAS**

<b>UCEM</b>	Unión Cementera Nacional
<b>JS</b>	JavaScript
<b>BD</b>	Base de Datos
<b>E-Salud</b>	Salud electrónica
<b>LP</b>	Linear Programming (Programación Lineal)
<b>GUI</b>	Interfaz Gráfica de Usuario
<b>C/S</b>	Cliente-Servidor
<b>FIE</b>	Facultad de Informática y Electrónica
<b>IDE</b>	Entorno de desarrollo integrado
<b>DBMS</b>	Database Management System
<b>PNBV</b>	Plan Nacional del Buen Vivir
<b>SYSPLA</b>	Sistema de Planificación
<b>API</b>	Application Program Interface (interfaz de programación de aplicaciones).
<b>HTTP</b>	Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto).
<b>MEAN</b>	Conjunto de tecnologías Mongo DB, Express.js, Angular.js, Node.js
<b>SW</b>	Software
<b>HW</b>	Hardware
<b>W3C</b>	World Wide Web Consortium
<b>HTML</b>	HyperText Markup Language (Lenguaje de marcado de hipertexto).
<b>CSS</b>	Cascade Style Sheets (Hojas de Estilo)
<b>NPM</b>	Node Package Manager (Administrador de paquetes de Node)
<b>CDN</b>	Content Delivery Network

## INDICE DE TABLAS

<b>Tabla 3-1</b>	Requerimientos Funcionales .....	30
<b>Tabla 3-2</b>	Roles de Scrum .....	37
<b>Tabla 3-3</b>	Historias de usuario utilizadas en Scrum.....	38
<b>Tabla 3-4</b>	Tiempos por historias de usuario .....	39
<b>Tabla 3-5</b>	Valores calculados de las historia y valores ideales .....	39
<b>Tabla 3-6</b>	Pruebas unitarias de cada requerimiento .....	41
<b>Tabla 3-7</b>	Identidad .....	43
<b>Tabla 3-8</b>	Contenido .....	43
<b>Tabla 3-9</b>	Utilidad .....	44
<b>Tabla 3-10</b>	Navegación .....	44
<b>Tabla 3-11</b>	Retroalimentación.....	44

## INDICE DE ILUSTRACIONES

<b>Figura 1</b>	Funcionamiento del Mean Stack en el sistema .....	7
<b>Figura 1-1.</b>	Diagrama Relacional Estudiante – Tipo Estudiante .....	10
<b>Figura 1-2</b>	Representación tipo árbol Base de datos no relacionales documento JSON.....	10
<b>Figura 1-3</b>	Diagrama diseño referencial Base de datos no relacionales documento JSON.....	11
<b>Figura 1-4</b>	Estructura de Proyecto Angular.js.....	14
<b>Figura 1-5</b>	Sitio Web de Microsoft Internet Explorer sin JavaScript ni CSS.....	15
<b>Figura 1-6</b>	Página web Microsoft 1994 .....	16
<b>Figura 1-7</b>	Estructuras de Archivos para el proyecto en Node.js .....	17
<b>Figura 1-8</b>	Package.json.....	17
<b>Figura 1-9</b>	Ejecución de Package.json en la línea de comando .....	18
<b>Figura 1-10</b>	Declaración de las dependencias de Node.js.....	18
<b>Figura 1-11</b>	Ejecución del servicio en Node.js.....	19
<b>Figura 1-12</b>	Ejemplo de modelo Paciente .....	20
<b>Figura 1-14</b>	Archivo de ruta con los servicios expuestos .....	21
<b>Figura 1-17</b>	Consumo de Servicio con Angular.js .....	24
<b>Figura 1-18</b>	Fragmento CIE 10.....	25
<b>Figura 2-1</b>	Roles Scrum con sus interacciones.....	27
<b>Figura 3-1</b>	Diagrama de Procesos.....	31
<b>Figura 3-2</b>	Arquitectura del sistema .....	32
<b>Figura 3-3</b>	Diagrama de despliegue de la arquitectura .....	33
<b>Figura 3-4</b>	Logo Sistema Health End Point .....	34

<b>Figura 3-5</b>	Interfaz gráfica genera Health End Point .....	34
<b>Figura 3-6</b>	Formulario Health End Point con Bootstrap .....	35
<b>Figura 3-7</b>	Diagrama de secuencia.....	36
<b>Figura 3-8</b>	Burndown chart Scrum .....	40
<b>Figura 3-9</b>	Función de test Rest.....	42
<b>Figura 3-10</b>	Función de test SOAP.....	42
<b>Figura 3-11</b>	Análisis Usabilidad.....	45

## **INDICE DE ANEXOS**

**ANEXO A: DISEÑO DE LA BASE DE DATOS**

**ANEXO A1: DIAGRAMA DE LA BASE DE DATOS**

**ANEXO A2: DICCIONARIO DE DATOS**

**ANEXO B: HISTORIAS DE USUARIO SCRUM**

**ANEXO C: PRUEBAS DE UNITARIAS**

## RESUMEN

El presente trabajo de investigación contiene el desarrollo de un sistema E-Salud para la empresa Unión Cementera Nacional (UCEM) que se ubica en el cantón Riobamba, provincia de Chimborazo, cuenta con un personal de 187 empleados entre operarios y administrativos; el centro médico cuenta con un médico ocupacional y 2 enfermeras de apoyo que son las personas encargadas de la historia clínica ocupacional. Mediante tecnologías web; Mean Stack JavaScript que es una plataforma conformada por MongoDB, Express.js, Node.js y Angular.js, en conjunto con estándares como el CIE10, que utilizan los médicos ocupacionales, para la gestión de la historia clínica ocupacional. Se utilizó el IDE WebStorm para gestionar lenguajes de programación JavaScript, en conjunto con la Metodología Scrum para desarrollo de software y con el Método Deductivo se plasmó los requerimientos del departamento, estableciendo 7 requerimientos funcionales que se desarrollaron en el lapso de 21 días. La velocidad del proyecto de investigación según Scrum se la realiza mediante tarjetas conocidas como historias de usuario donde se define el tiempo estimado para la actividad y el tiempo real en la que se realizó, cada una de ellas tiene una prueba de aceptación en donde se valida o se refactorizan errores. Al final de la metodología se plasma dicha velocidad en un gráfico conocido como BurnDown Chart, en donde se observa si el proyecto de investigación está acorde a lo planificado, se realizó una prueba de rendimiento, dando como resultado la carga de 206000 datos en 1 hora con 35 min, lo cual es aceptable. Se recomienda continuar con el desarrollo de más módulos al proyecto, como lo son el de morbilidad y ausentismo laboral; y aplicar criterios de seguridad más amplios como tokens OAUTH para los servicios web.

**Palabras Clave:** <SISTEMA WEB><RECURSOS WEB><ARQUITECTURA ORIENTADA A SERVICIOS>< ENTORNO DE DESARROLLO INTEGRADO>< TECNOLOGÍA MEAN STACK><UNION CEMENTERA NACIONAL(UCEM)><RIOBAMBA(CANTÓN)>

## SUMMARY

This research develops an E-Health system for the Union Cementera Nacional (UCEM) enterprise which is located in Riobamba city, Chimborazo Province, the enterprise has 187 employees among workers and administrators; the health center has an occupational doctor and two nurses who are in charge of the occupational medical record and use the Mean Stack JavaScript web technology which is a platform composed by MongoDB, Express.js, Node.js and Angular.js in a set with standards such as CIE10, which are used by the occupational doctors for the occupational medical record management. The IDE WebStorm was used to process the JavaScript programming language together to the Scrum Methodology to develop the software and with the Deductive Method the department requirements were fulfilled by establishing 7 functional requirements which were developed in 21 days. The research project speed according to Scrum is made through cards known as user records where the estimated time and real time are defined, each card has an acceptance test in order to validate or re-factorize errors. At the end of methodology, the speed is expressed in a graph known as BurnDown Chart. Which shows if the research project presents the planned results, a performance test was applied, and the results were a load of 206000 data in 1 hour and 35 minutes, which is acceptable. It is recommended to continue with elaboration of more modules for the project, such as sickness rate and absenteeism, and apply broader safety criteria as tokens OAUTH for web services.

**Key Words:** <WEB SISTEM><RESOURCES><ARQUITECTURE SOA (SERVICES ORIENTITY ARQUITECTURE)>< IDE (INTEGRATE DEVELOPER ENVIRONMENT)><MEAN STACK>< JSON (JAVASCRIPT OBJECT NOTATION)>

## INTRODUCCIÓN

En un mundo donde la tecnología está presente en nuestras vidas, las herramientas software son fundamentales para simplificar procesos, mediante dispositivos como celulares, tablets y la computadora personal, que ayuda en cada actividad diaria, tomando específicamente en cuenta el caso de Salud Ocupacional y la gestión de la historia clínica ocupacional.

Se apoya a la empresa Unión Cementera Nacional (UCEM) que posee 187 empleados en las diferentes plantas, que realizan actividades que ponen en riesgo su integridad personal en sus puestos de trabajo, por medio de un sistema de historias clínicas ocupacionales, de esta manera se satisface y personaliza con el procesamiento y gestión de datos, debido a que actualmente lo realizan de forma manual y existe pérdida de información.

Con los cambios de política o de procesos en el Ministerio de Salud Pública y en el Instituto Ecuatoriano de Seguridad Social. La plataforma se acopla para poder crecer con el tiempo dando flexibilidad y permitiendo la integración de módulos según las necesidades del departamento médico.

El Mean Stack JavaScript es una alternativa para la solución de estos problemas, debido a que toda la tecnología es perfectamente integrable por su lenguaje, tanto en la parte de interfaz gráfica conocido como Front-End y en la parte de interacciones y procesamiento de información cuyo nombre referido es Back-End. Por último, la base de datos no relacional a utilizar es basada en archivos JSON que prácticamente son objetos.

Durante el proceso de desarrollo se toma en cuenta el estándar CIE 10 para las enfermedades, los médicos ocupacionales, para cuidar la integridad en cuanto a los síntomas que presenten los pacientes y poder agruparles para tomar decisiones futuras de salud correctiva y preventiva pro puesto de trabajo.

Todo lo mencionado anterior mente en conjunto con metodologías de desarrollo de software plasma en el documento la realización de un sistema y la automatización de los procesos requeridos por el departamento médico.

## **Antecedentes**

Tomando como inicio el cambio al cual el país se sujeta, se presentan nuevos desafíos tanto en el campo informático como en la salud, las herramientas software nos ayudan a tomar decisiones en base al procesamiento de datos, apoyando a la solidificación de las nuevas políticas de forma eficiente y confiable en este caso se plantea apoyar a lo establecido en el código de trabajo en el artículo 434 que estipula:

*“Reglamento de higiene y seguridad. - En todo medio colectivo y permanente de trabajo que cuente con más de diez trabajadores, los empleadores están obligados a elaborar y someter a la aprobación del Ministerio de Trabajo y Empleo por medio de la Dirección Regional del Trabajo, un reglamento de higiene y seguridad, el mismo que será renovado cada dos años.”*  
(MINISTERIO DEL TRABAJO, 5p)

Y el acuerdo ministerial 1404 que menciona “El reglamento para el funcionamiento de los servicios médicos de empresas. “(REGLAMENTO SALUD OCUPACIONAL, 2p.).

Definiendo salud ocupacional como: “una actividad multidisciplinaria dirigida a promover y proteger la salud de los trabajadores mediante la prevención y el control de enfermedades y accidentes y la eliminación de los factores y condiciones que ponen en peligro la salud y la seguridad en el trabajo.

Además, procura generar y promover el trabajo seguro y sano, así como buenos ambientes y organizaciones de trabajo realizando el bienestar físico mental y social de los trabajadores y respaldar el perfeccionamiento y el mantenimiento de su capacidad de trabajo.

A la vez que busca habilitar a los trabajadores para que lleven vidas social y económicamente productivas y contribuyan efectivamente al desarrollo sostenible, la salud ocupacional permite su enriquecimiento humano y profesional en el trabajo “(Definición Salud Ocupacional, 1p.).

La OMS (Organización Mundial de la Salud) plasma en el documento “Entornos Laborales Saludables: Fundamentos y Modelo de la OMS” el tratamiento adecuado de lo que es salud ocupacional y seguridad industrial con el cual se manejan toda la información sobre las formas de gestionar las historias clínicas de pacientes.

Con todo lo mencionado anteriormente se puede proceder a definir que un sistema de e-Salud ocupacional: es más que una herramienta software la cual es capaz de interactuar con los pacientes y el médico permitiendo compartir información sobre el estado de salud mediante el estándar CIE 10 para que tengan acceso mediante cualquier dispositivo móvil, web, etc.

Permita mejoras durante el proceso de manipulación de historias clínicas ocupacionales que por diferencia principal estas poseen información adicional sobre los puestos de trabajos donde se desenvuelve el empleado.

Las herramientas comunes de este tipo de sistemas utilizan base de datos relacionales que se basan en el modelo Entidad-Relación que plantea:” El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre estos objetos. Una entidad es una «cosa» u «objeto» en el mundo real que es distinguible de otros objetos” (Sudarshan, et al., 2002, 32p);

Lo cual hace que los sistemas actualmente sean fáciles de manejar, pero tienen un problema de flexibilidad lo que hace que el nuevo modelo de diseño de bases de datos que maneja el concepto NoSQL (No Structured Query Language) que se entienden como no relacionales; las mismas que están orientados a objetos y son buenas para manejar documentos conjuntos a varias tecnologías que se mencionarán en el presente documento.

Las nuevas tendencias en el desarrollo de software hacen que los lenguajes sean cada vez de más alto nivel, de esta forma se propone una mejor comunicación entre programador-lenguaje para poder desarrollar con mayor rapidez y con código más organizado y entendible; se define como stack a un conjunto de herramientas las cuales se pueden integrar de manera intrínseca sin necesidad de librerías extras.

De esta manera los programas son más rápidos y de fácil acople al momento de integrar módulos; tomando en cuenta todo lo anterior mencionado se puede entender por Main Stack a la integración de las tecnologías MongoDB como base de datos no relacional que permite flexibilidad al manipular datos orientados a objetos.

Plasmados como documentos de esta forma puede variar con el tiempo sin afectar a los datos previamente ingresados o establecidos; las búsquedas son más rápidas que las SQL tradicionales debido que maneja estructuras hash por detrás para poder obtener información mediante índices.

Node.js con las dependencias de Express.js para que permita una comunicación mediante un servidor de Web API (Application Programming Interface) la cual intercambia información mediante métodos “Post” (Petición de Servicios Web enviando parámetros) y “Get” (Petición de datos sin enviar parámetros) con cualquier aplicación que desee consumir sus servicios.

En la parte de Interfaces gráficas actúa Angular.js que es un framework el cual permite manejar dinámicamente páginas HTML5 permitiendo minificar el código y que de esta forma sean más eficientes; la integración de estas tecnologías es eficiente debido a que manejan JavaScript.

Todas y no se necesita librería alguna para poder intercambiar tipos de datos todo se maneja bajo los estándares JSON (JavaScript Object Notation) que son alternativas de comunicación a los comunes XML (extensible lenguaje de marcas) utilizadas en sistemas web programados con JAVA o .NET.

### **Formulación del problema**

La empresa Cemento Chimborazo tiene como misión “producir y comercializar cemento y productos derivados, con altos niveles de productividad y calidad para satisfacer las necesidades de sus clientes, contribuyendo además al desarrollo del país con responsabilidad socio ambiental y crecimiento sostenido” (CEMENTO CHIMBORAZO, 1p.),

Se necesita que los trabajadores se encuentren en óptimas condiciones en lo que respecta a salud ocupacional y seguridad industrial, el centro médico no cuenta con una herramienta software la cual apoye en el procesamiento de historias clínicas.

Debido a la que cuenta con 187 empleados a los que se les realiza pruebas periódicas, poseen en la unidad médica con 2 médicos y 4 enfermeras que se dedican a la atención del personal de la Institución lo cual hace que los procesos sean eficientes pero lentos sin tener el apoyo de una herramienta capaz de gestionar dichas actividades.

El sistema lo podrán utilizar los médicos mediante una aplicación web la cual gestione las historias clínicas ocupacionales, para poder eliminar el riesgo de la pérdida de datos y poder gestionar de forma adecuada la historia clínica ocupacional.

## **Justificación del Trabajo de Titulación**

A continuación, se plasma las diferentes justificaciones por las cuales se realizó el trabajo de titulación teniendo en cuenta la problematización y los contenidos que se trataran para poder resolver el problema detectado durante el proceso de análisis del problema.

## **Justificación Teórica**

El Mean Stack java script se define como la integración de varias tecnologías que se acoplan entre ellas para que puedan gestionar de forma eficiente el desarrollo de sistemas actualmente; MEAN son las siglas de las tecnologías java script que intervienen, que son:

MongoDB que es una base de datos noSQL, Express que es un framework de servidores web y servicios http que gestiona Node.js y finalmente Angular.js que es un framework desarrollado por google el cual maneja un modelo MVC (Modelo Vista Controlador) el cual se utiliza en la parte de front end (Código que se maneja en la parte de interfaz gráfica).

Las bases de datos no relacionales plantean un nuevo paradigma en el cual no todo está sujeto al manejo de relaciones que hacen que los sistemas sean ineficientes con grandes cantidades de datos; frente a esto se plantea aplicar la flexibilidad que manejan estas nuevas bases de datos para que pueda adaptarse a diferentes dispositivos.

Todo esto se plantea integrar mediante una arquitectura SOA (Service Oriented Architecture) que se define como “La arquitectura orientada a servicios es un paradigma para organizar y utilizar capacidades distribuidas que pueden estar controladas bajo diferentes propietarios e implementadas bajo diferentes tecnologías” (de la Torre & González, 2008).

La parte de servicios web que manipula Node.js es más eficiente a diferentes servicios como los da Glassfish con java o el mismo asp.net dando como pie una alternativa para implementación que tenga un lenguaje más simple como java script y gratuito.

El estándar que se hablará en este tipo de servicios web son los conocidos JSON (JavaScript Object Notation); los cuales son una alternativa ante los XML que gestionan servicios Rest(Representational State Transfer) o Restfull(una extensión del Rest utilizada por microsoft) o los mismos SOAP(Simple Object Access Protocol)que permiten comunicación entre sistemas.

La solución adoptada con el Mean Stack se integra perfectamente y permite flexibilidad durante el desarrollo especialmente al utilizar base de datos no SQL para que el sistema se pueda sujetar a cambios sin ningún inconveniente.

Uniendo esta tecnología con el documento mencionado de la OMS se puede plantear una solución sustentable para el manejo de historias clínicas ocupacionales a través de un sistema de E-Salud ocupacional web el cual puedan acceder médicos, pacientes y gerentes para obtener información competente a cada área.

### **Justificación Aplicativa**

El proyecto consta de los siguientes módulos:

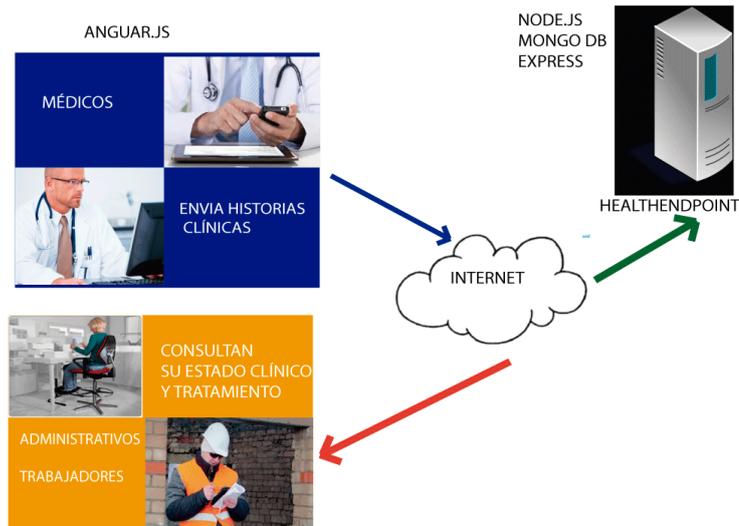
- Módulo de Autenticación
- Módulo de Historias Clínicas Ocupacionales
- Módulo de Pacientes

Se pretende plasmar mediante las nuevas tendencias sobre desarrollo de aplicaciones multidispositivo poniendo énfasis a la parte web y teniendo en cuenta la disponibilidad que da el Internet, basándose en una arquitectura SOA que no es más que la orientación de servicios web para que lo puedan consumir las aplicaciones que lo requieran y con la ayuda de JavaScript como pilar fundamental con el MEAN Stack.

Que nace de las siglas de plataformas como lo son: MongoDB como base de datos no relacional; la cual permite una gran flexibilidad y de esta forma los sistemas puedan adaptarse a cualquier medio en el cual el ambiente cambie; Express y Angular.js que es uno de los principales pilares reemplazando a Ajax como parte del Front-End.

Y como columna vertebral de Node.js que es el que administrará en la parte de Back-End todo pertinente a servicios web para alimentar a cada uno de las aplicaciones que se deseen implementar en el siguiente proyecto conjunto con metodologías ágiles las cuales nos permitan entregar de forma eficiente avances sin generar mucha documentación.

Los médicos que forman parte del departamento podrán acceder al sistema para ingresar los datos de las historias clínicas y estos después de cumplir el proceso antes mencionado estarán a disponibilidad de los pacientes, así como también del personal administrativo para ayudar a la toma de decisiones.



**Figura 1** Funcionamiento del Mean Stack en el sistema

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

## Objetivos

A continuación, se detalla los objetivos del proyecto, con cada uno de ellos se desglosan diferentes actividades integrando la investigación de la teoría y plasmar de forma adecuada en el proceso del desarrollo del sistema.

### Objetivo General

Desarrollar un sistema E-Salud ocupacional mediante Mean Stack JavaScript en la empresa Cemento Chimborazo.

## **Objetivos Específicos**

- Investigar sobre Mean Stack JavaScript para crear una arquitectura flexible y adaptable a futuros cambios en la historia clínica salud ocupacional.
- Aplicar el estándar médico CIE 10 diseño del sistema E-Salud Ocupacional que emite información necesaria para el control de historias clínicas ocupacionales tanto para médicos, trabajadores y administrativos de la empresa Cemento Chimborazo.
- Desarrollar el sistema E-Salud Ocupacional para la gestión de historias clínicas ocupacionales

# CAPITULO I

## MARCO TEORICO REFERENCIAL

### 1.1 MEAN Stack JavaScript

Dentro de las tecnologías actuales con las que se desarrolla aplicaciones web multi-dispositivo y para alto rendimiento, se plantea el uso del paradigma del MEAN Stack JavaScript que es el conjunto de frameworks que se acoplan basados en JavaScript que se mencionan a continuación:

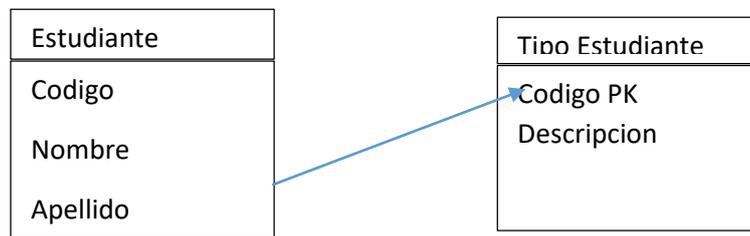
#### 1.1.1 Mongo DB

MongoDb es un gestor de base de datos no relacionales fundamental para poder proponer la arquitectura flexible; Se adapta a cualquier estándar debido a que rompe las teorías del modelo relacional, de esta manera se puede mejorar tanto los datos, como las estructuras de tablas sin necesidad de reestructuración total de las bases de datos.

Toda la teoría de base de datos no relacionales se la complementa, asociando analogías a las bases de datos no relacionales tomando en cuenta las diferencias de flexibilidad que presentan al ser objetos y ya no registros estáticos en memoria.

- Las tablas se las denomina colecciones.
- Los registros son objetos JSON
- Los campos son propiedades del objeto JSON que pueden ser otros objetos JSON dentro del mismo.

Durante el proceso de diseño de base de datos se puede tomar dos tipos de diseño los cuales son:



**Figura 1- 1.** Diagrama Relacional Estudiante – Tipo Estudiante

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

```
//Coleccion Estudiante
{
  Codigo:"cod1",
  Nombre:"Nombre",
  Apellido:"Apellido",
  Sexo:"Masculino",
  Tipo_Estudiante:{
    Codigo:"cod1",
    Descripcion:"Postgrado"
  }
}
```

**Figura 1-2** Representación tipo árbol Base de datos no relacionales documento JSON.

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

```

//Coleccion Estudiante
{ Codigo:"cod1",
  Nombre:"Nombre",
  Apellido:"Apellido",
  Sexo:"Masculino",
  Tipo_Estudiante: "cod1"
}
// Coleccion Tipo_Estudiante
{
  Codigo:"cod1",
  Descripcion:"Postgrado"
}

```

**Figura 1-3** Diagrama diseño referencial Base de datos no relacionales documento JSON

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

La figura 1-1 representa el diseño relacional el cual se llevará a convertir en base de datos no relacional.

Los documentos JSON de la figura 1-2 y 1-3 plasman los datos y la forma de almacenamiento en MONGODB, para poder comprender se debe tener en cuenta que un documento JSON siempre tendrá el formato entre llaves y cada campo serán formateados y separados por el símbolo “:” para diferenciar el nombre que se le ponga al campo y el valor que adopta el mismo ejemplo:

```

{
  "campo": "<valor>"
}

```

En la figura 1-2 el diseño en forma de árbol, permite utilizar una funcionalidad de MONGODB que es el almacenamiento de objetos dentro de objetos se puede notar que el parámetro

tipo\_estudiante es un objeto interno dentro del objeto estudiante, las ventajas de este tipo de almacenamiento son:

- Rapidez para las búsquedas sin referencias.
- Cuida la integridad de datos al encapsular.
- Flexibilidad al momento de aumentar o disminuir parámetros dentro de los objetos tanto internos como externos.

Cabe recalcar que se sacrifica un poco de redundancia por mayor rapidez al procesarla, pero no existe problema de almacenamiento debido a que al ser almacenado como documento JSON su compresión es mucha más eficiente que un registro de base de datos.

En la figura 1-3 se nota algo muy conocido en el ámbito relacional, este tipo de diseño permite referenciar tablas y trabajar con sus códigos de referencia; las ventajas que presta este tipo de diseño son las siguientes:

- Orden en el almacenamiento de datos,
- Analogías más exactas con el modelo Referencial del base de datos relacionales.
- Flexibilidad al aumentar o disminuir campos con respecto al diseño original.

En cuanto a concurrencia y simultaneidad no existe problemas porque utiliza procesos en paralelo para atender las peticiones que se realicen a la base de datos cumpliendo su objetivo de almacenamiento cuidando parámetros esperados para el desarrollo de sistemas informáticos.

Es relevante mencionar que los tipos de dato son variados, se tiene los tipos de datos nativos como son:

- String
- Bit
- Boolean
- Integer
- Double
- Float

Pero además de ellos lo que le hace una estructura robusta para grandes cantidades de datos es el almacenamiento de bloques de datos en:

- Arrays (Vectores).
- JSON (JavaScript Objects)

### 2.1.2 Angular.js

En la parte de presentación del cliente se tiene una gran herramienta que ayuda en la interacción con Node.js como es Angular.js, que permiten a las aplicaciones a más de ser multidispositivo interactuar y dar formatos más amigables a nuestras aplicaciones, permitiéndoles adaptarse a cualquier dispositivo sea móvil o Pc.

Angular.js es un framework creado por google el cual permite un mejor manejo del patrón MVC (Modelo Vista Controlador) permitiendo generar e inyectar dependencias programadas previamente;

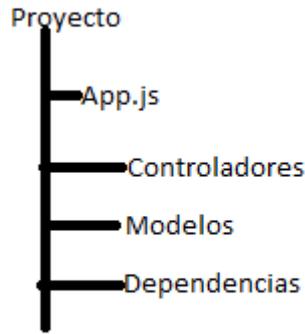
La vista es compatible con HTML5 que es el formato actual para páginas web normado por la W3C (World Wide Web Consortium) que es la encargada de normar el despliegue diseño y normativas de la web actual, cabe recalcar que el lenguaje que utiliza es JavaScript lo que es perfecto para paginas dinámicas y para reutilización de código, en conjunto a su rendimiento de velocidad de carga e iteración con los eventos.

Existen 2 formas de implementar Angular.js la primera es con el CDN (Content Delivery Network):

```
(<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.4.5/angular.min.js"></script>)
```

que Angular y algunas empresas proveen, no es más que un link de referencia a las librerías base para que reconozca el uso de Angular.js en el proyecto, para comprender más a fondo no se debe olvidar que en la web todos son archivos planos que pueden ser modificados según se desee.

La segunda forma para obtener Angular.js es descargarlo de la página oficial y agregarlo a las dependencias del proyecto que se utilizarán para que reconozca las librerías base, en angular se puede manejar la siguiente estructura de archivos de la figura 1-4:



**Figura 1-4** Estructura de Proyecto Angular.js

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

Dentro de la carpeta que almacena el proyecto se tiene las páginas HTML5 referenciadas a los archivos app.js y en la carpeta de controladores se tendrá todos los archivos JavaScript que manejen la iteración con el Back-End y en caso de necesitarlo permite manejar modelos los cuales son objetos JSON formateados para poder implementar automatizaciones en la vista con respecto al controlador.

Otra funcionalidad importante de manejar de Angular.js es las inyecciones de dependencias, esto permite manejar el patrón Factory que permite tener centralizado tanto servicios web que consume de cualquier api Rest o Full Rest, o ser utilizado como archivo configurable para centralización de variables de configuración.

Angular.js interactúa con la tecnología web de ngStorage que actualmente es fundamental para el uso temporal debido a políticas de cookies a nivel mundial es una buena opción para guardar temporalmente en el cache interno del navegador como un hash con una clave se puede llegar a obtener el contenido de cualquier objeto o tipo de dato que se quiera almacenar.

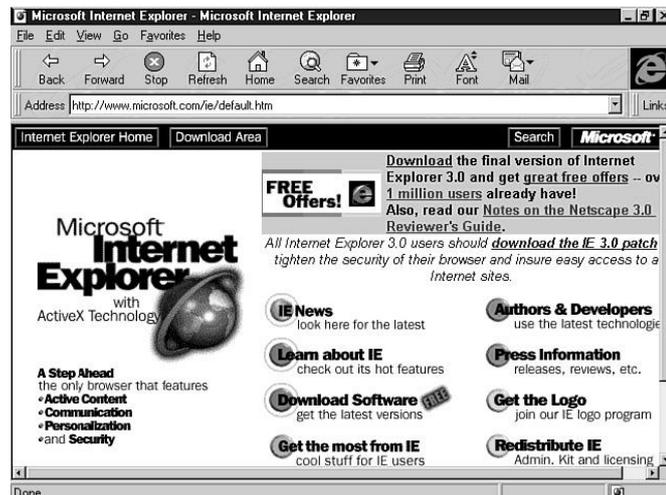
Lo más relevante es que permite manejar los datos en caso que existan problemas de red guardando temporalmente los formularios llenos para posteriormente cuando se encuentre en modo online envíe la información recolectada y libere ese espacio de memoria.

Dentro de la vista Angular.js utiliza un framework el cual permite el manejo de interfaz responsive que significa que se adapta a la pantalla de cualquier dispositivo el cual es conocido como BOOTSTRAP que maneja la interfaz en forma de grid (columnas) con las cuales se puede referenciar los componentes y decidir cuantos se renderizarán dependiendo del dispositivo.

Para agregar este framework al proyecto existen las dos formas igual que con Angular.js que es con el CDN o descargando el archivo minificado y referenciándolo y su único requerimiento es tener jquery dentro del proyecto.

### 2.1.3 Node.js

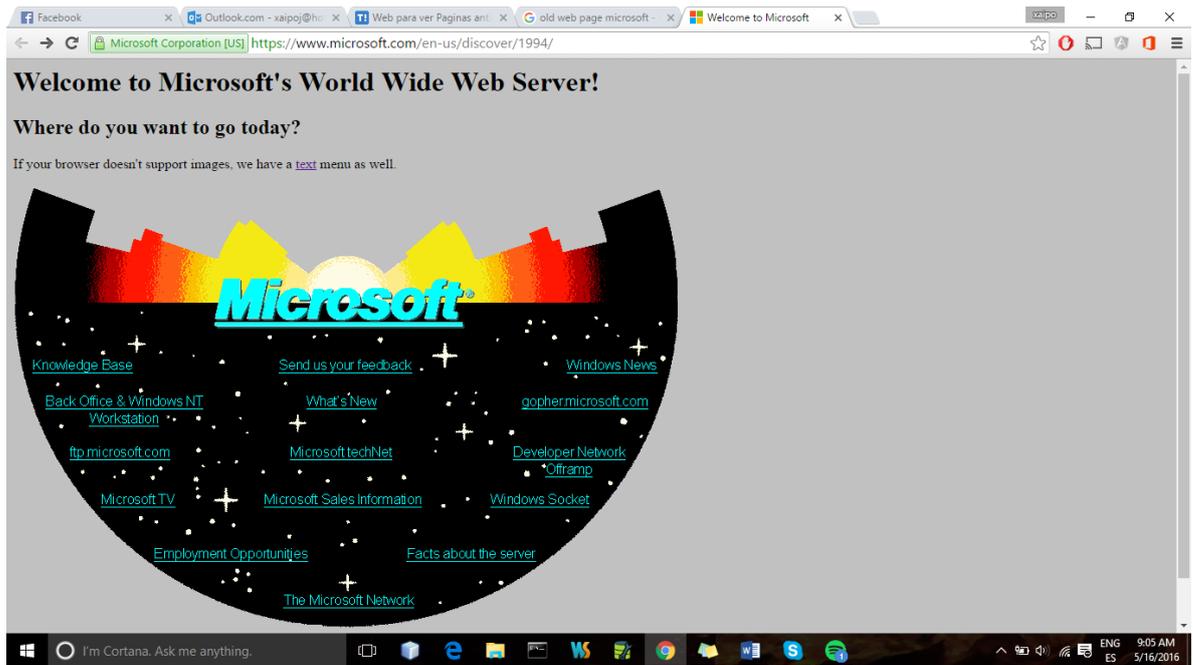
Para nombrar Node.js es necesario nombrar a JavaScript, que es un lenguaje de programación que se acostumbra a estar presente en la parte de interfaces gráficas, hace que la web de hoy posea funcionalidades agradables a la vista del usuario final, en la antigüedad la web 1.0 como se la conoce era solo procesamiento de texto sin animaciones como se lo puede observar en la figura 1-5 que es la página web de Microsoft Internet Explorer sin el uso de JavaScript a esta tecnología se la conoce como la web 1.0



**Figura 1-5** Sitio Web de Microsoft Internet Explorer sin JavaScript ni CSS

Tomado de: [http://hitech.vesti.ru/news/view/id/6650\\_2016](http://hitech.vesti.ru/news/view/id/6650_2016)

Con el paso del tiempo y la integración de JavaScript como lenguaje interpretado por los navegadores web permitió la integración y animación de sitios web, haciendo que la interfaz gráfica sea agradable al usuario incluyendo colores con el nacimiento de CSS que son hojas de estilo para las páginas HTML estándar, como se pudo ver en la figura 1-6 la diferencia de la página web de Microsoft en el año 1994, a esto se le conoce como el nacimiento de la web 2.0



**Figura 1-6** Página web Microsoft 1994

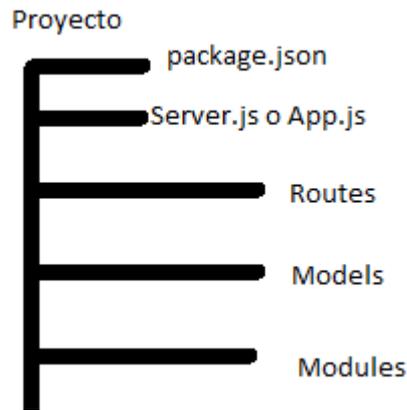
**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

En la actualidad el surgimiento de los nuevos paradigmas web para multidispositivos ha hecho que la web evolucione a lo más conocido como web 3.0 que integra varias funcionalidades tanto gráficas como de procesamiento de la información.

Node.js es un conjunto de herramientas que gestionan por medio de motor V8 de JavaScript que maneja google, el cual está diseñado para ejecutar código JavaScript de manera muy eficiente y extremadamente rápida, del lado del servidor.

Una de sus funcionalidades es que puede gestionar tareas de forma asíncrona, teniendo en su estructura un pool de threads (piscina de hilos) las cuales se delegaran las tareas a cada uno de los hilos para manejo asíncrono de cada actividad y esta tendrá una respuesta.

Para empezar a comprender la tecnología como en el caso de Angular.js se plasma la estructura de los archivos para su correcto funcionamiento como lo muestra la figura 1-7:



**Figura 1-7** Estructuras de Archivos para el proyecto en Node.js

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

El archivo package.json es uno de los más importantes que existen en esta estructura en el guarda todas las dependencias externas que se instalen en node.js para su posterior implementación, así como la información del nombre de la aplicación y el tipo de licencia que se le aplique, el archivo tendrá el siguiente formato de la figura 1-8:

```

{
  "name": "appCompras",
  "main": "server.js",
  "dependencies": {
    "body-parser": "^1.14.2",
    "cors": "^2.7.1",
    "express": "^4.13.3",
    "mongoose": "^4.3.4",
    "node-restful": "^0.2.4"
  }
}
  
```

**Figura 1-8** Package.json

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

El paquete puede instalar todas las dependencias que constan en el archivo abriendo la consola y en la ubicación donde se encuentre el archivo ejemplo:

C:\Proyecto\Package.json> escribir npm install

En la figura 1-9 el comando llama a un módulo interno de node.js el encargado de gestionar los paquetes:

```
C:\WINDOWS\system32\cmd.exe
C:\Users\xaipo>cd..
C:\Users>cd..
C:\>cd tesis
C:\tesis>cd tesisSaludOcupacional
C:\tesis\tesisSaludOcupacional>cd HttpServices
C:\tesis\tesisSaludOcupacional\HttpServices>dir
Volume in drive C has no label.
Volume Serial Number is 461E-86B7

Directory of C:\tesis\tesisSaludOcupacional\HttpServices

05/03/2016  09:33 PM    <DIR>          .
05/03/2016  09:33 PM    <DIR>          ..
04/13/2016  09:38 AM    <DIR>          Models
01/21/2016  08:53 PM    <DIR>          node_modules
01/21/2016  08:53 PM                208 package.json
05/04/2016  07:16 PM    <DIR>          Routes
05/03/2016  09:33 PM                4,328 server.js
                2 File(s)      4,536 bytes
                5 Dir(s)    30,671,781,888 bytes free

C:\tesis\tesisSaludOcupacional\HttpServices>npm install
```

**Figura 1-9** Ejecución de Package.json en la línea de comando

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

Una vez instalados los paquetes se puede observar generada una carpeta con el nombre node\_modules y dentro cada una de las dependencias descargadas con el npm dentro de la misma, estas dependencias se las usara de la siguiente forma en el archivo app.js o server.js, que prácticamente vendrían a ser los archivos centrales para la ejecución del servicio.

En ellos se encontrarán 3 partes fundamentales la primera todos los módulos que se ocuparán declarados como muestra la figura 1-10:

```
var express = require('express');
var mongoose= require('mongoose');
var bodyParser= require('body-
parser');
var cors = require('cors');
```

**Figura 20** Declaración de las dependencias de Node.js

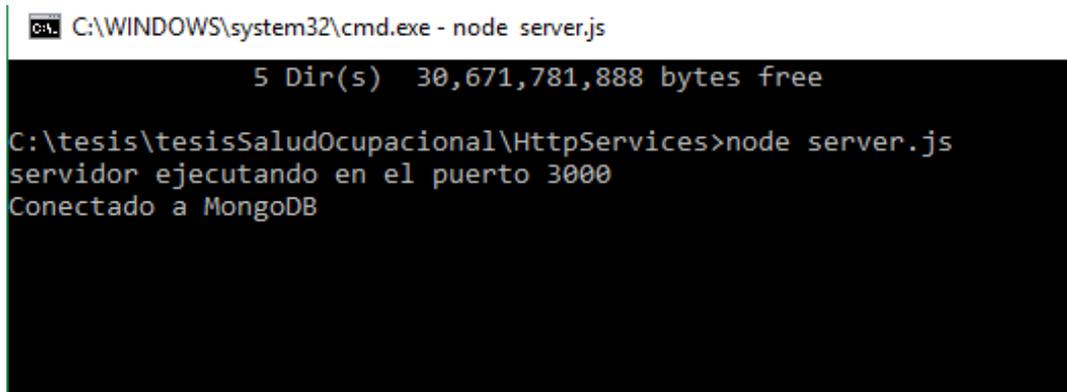
Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

Para ejecutar el programa se lo puede realizar desde la consola similar a la instalación de paquetes con npm ubicándose en el lugar de ejecución del archivo ejecutando el comando “node” ejemplo:

node app.js

node server.js

El nombre app.js o server.js son indistintos, pero se recomienda usarlos por cuestión de estandarización de estructura como muestra la figura 1-11.



```
C:\WINDOWS\system32\cmd.exe - node server.js
5 Dir(s) 30,671,781,888 bytes free
C:\tesis\tesisSaludOcupacional\HttpServices>node server.js
servidor ejecutando en el puerto 3000
Conectado a MongoDB
```

**Figura 311** Ejecución del servicio en Node.js

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

A continuación, se explicará más detallada la parte de la creación del servidor web con Express.js

#### 2.1.4 Express.js

Es un framework creado para node.js basado en JavaScript el cual permite ejecutar servidores web http tanto para publicaciones de interfaces web como para servicios web en este caso JSON, para adquirir express en el proyecto se debe ejecutar en la dirección donde se desea agregar el siguiente comando:

```
npm install express --save
```

esto permite que express se descargue y se guarde al dependencia en el archivo package.json mencionado anteriormente; a continuación se explica la estructura de express.js utilizado como servidor de web api.

Para utilizar express dentro del servidor web se lo debe llamar al archivo server.js, en donde se declara una variable que contendrá todos los accesos a sus funcionalidades con la siguiente línea:

```
var express = require('express');
```

luego de debe declarar los módulos necesarios para que los servicios web puedan funcionar con las siguientes líneas:

```
var app= express();
app.use(bodyParser.urlencoded({extended:true}));
app.use(bodyParser.json());
app.use(cors());// permite angular interactuar
```

Como se puede observar se utiliza un módulo llamado body parser para poder entender los modelos que ya se mencionarán más adelante y también el módulo de cors que permite la interacción de los servicios JSON get, post, push ,etc.

Luego de definir todo se aplica otro modulo que automatiza ese proceso de post y get se llama node-restfull el cual gestiona automáticamente los modelos con los servicios web se lo instala mediante el comando igualmente en la consola:

```
npm node-restfull --save
```

con todo esto está listo el aplicativo para poder generar los servicios web para ello se necesita los modelos, que no son más que archivos .js que tiene los mismos campos y tipo de dato de la base a continuación en la figura 1-12 un ejemplo de modelo:

```
var restful= require('node-restful');
var mongoose= restful.mongoose;

var categoriaSchema = new mongoose.Schema({
  cedula: String,
  puesto_trabajo: mongoose.Schema.ObjectId,
  primer_nombre: String,
  segundo_nombre: String,
  primer_apellido: String,
  segundo_apellido: String,
  sexo : String,
  fecha_nacimiento: String,
  ciudad: mongoose.Schema.ObjectId,
  edad: Number,
  telefono:String,
  estado_civil: mongoose.Schema.ObjectId,
  nivel_estudio: mongoose.Schema.ObjectId,
});

module.exports=
restful.model('pacientes',categoriaSchema);
```

**Figura 4** Ejemplo de modelo Paciente

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

Como se puede observar en la figura 1-12 existe la llamada a la librería de node-restful y a otra llamada mongoose, la cual es la encargada de gestionar el modelo para su instalación igualmente ejecutar en la línea de comandos:

```
npm install mongoose --save
```

A continuación, se realiza la petición de un esquema con la forma del objeto JSON con sus respectivos tipos de dato, y por ultimo necesita ser exportado al servidor para su utilización en la parte de web api para ello se crea rutas tanto en el archivo server.js como en cada una de las rutas de los servicios a implementar.

```
app.use('/api',require('./Routes/apiUsuarios'));
app.use('/api',require('./Routes/apiEmpresa'));
app.use('/api',require('./Routes/apiDependencia'));
app.use('/api',require('./Routes/apiJornada'));
app.use('/api',require('./Routes/apiMateriaPrima'));
```

**Figura 1-13** Rutas en el archivo server.js

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

Como se puede observar en la figura 1-13 la función app.use recepta como parámetros la ruta que en este caso es “/api” que puede ser cualquier nombre que se desee y en la segunda parte con la función require asocia el archivo de cada una de las APIS con los servicios expuestos en su dirección.

```
// <editor-fold defaultstate="collapsed" desc="Usuario">
var express= require('express');
var router= express.Router();
// </editor-fold>
// <editor-fold defaultstate="collapsed" desc="Modelos">
var Productos = require('./Models/modeloUsuario');
// </editor-fold>
Productos.methods(['get','put','post','delete','search']);
Productos.register(router,'/usuarios');

// </editor-fold>
//Return route
module.exports=router;
```

**Figura 1-14** Archivo de ruta con los servicios expuestos

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

Como se puede observar en la figura 1-14 en la parte superior del archivo de ruta es muy similar, incluye la llamada a Express.js en la siguiente parte le llama al modelo y finalmente con la librería de rest-ful se pide los métodos a construir de ese modelo, exportando una particular ruta similar a la de server.js que debe tener un nombre adecuado.

## **1.2 Apis basadas en arquitectura SOA.**

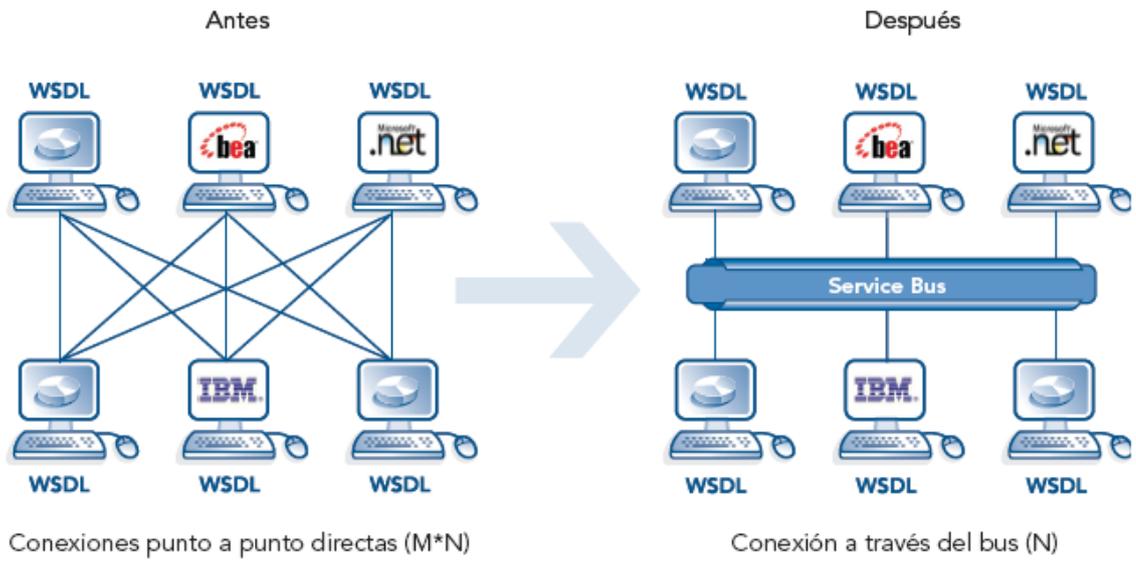
Durante el desarrollo de software lo primordial es la decisión de la arquitectura a utilizar, en la actualidad todo está orientada a la web y que mejor que SOA (Service Oriented Architecture); la arquitectura orientada a servicios para manejar de forma desacoplada este tipo de aplicaciones para que puedan ser consumidas por los clientes web que se deseen.

### **1.2.1 Definiciones y tipos de servicios web**

Para poder entender la arquitectura y la estructura que maneja el proyecto es necesario tener en cuenta las siguientes definiciones:

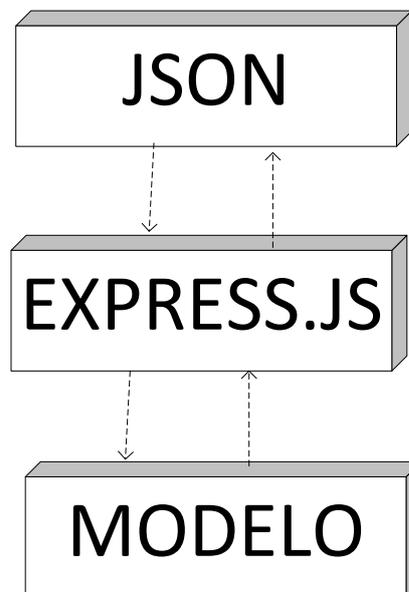
**Servicio Web.** - Es una interfaz de comunicación expuesta por un protocolo para el transporte de información con formato definido como XML o JSON para poder interactuar con clientes que deseen obtener dicha información.

**Arquitectura SOA.** - Arquitectura orientada al servicio eso quiere decir que el servicio web es lo primordial y lo que más énfasis se le tendrá ya que posee aplicaciones que exponen dichas funciones a cualquier aplicación externa que lo solicite como muestra la figura 1-15:



**Figura 1-15** Arquitectura Orientada a Servicios Evolución

Tomado de: <http://temariotic.wdfiles.com/local--files/tema52aes/ESB.PNG> 2016



**Figura 1-16** Arquitectura Rest JavaScript

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

El mensaje se lo realiza por JSON no con XML, el WSDL y UDDI se ven reemplazados solo por las rutas que maneja Express.js y los protocolos SOAP se transfieren directamente por el protocolo HTTP como lo muestra la figura 1-16 donde se ve la interacción de cada uno de estos componentes.

### 1.2.1 Consumo de servicios desde Angular.js

Existen varias formas de consumir desde los diferentes dispositivos, una de ellas son las ocupadas en el proyecto que es con Angular.js, que ofrece diferentes formas para consumir dichos servicios reemplazando las funciones que comúnmente se las realizaba con Ajax.

Para realizar el proceso necesitamos seguir los pasos anteriores de la sección que menciona Angular.js y se implementa la siguiente función que se muestra en la figura 1-17:

```
$http({
  method: 'GET',
  url: myProvider.getMateriaPrima() + '?_id'+aux,

  headers: {
    'Content-Type': 'application/json'
  }

}).then(function successCallback(response) {
  var n = response.data.length;
  // console.log(n);
  if (n == 0) {
    alert('no se encontro el materia prima);
  } else {
$scope.pacienteEncontrado.puesto_trabajo.materia_prima.push(response.data[0]);
  }
}, function errorCallback(response) {
  $scope.mensaje = response.mensaje;
});
}
```

**Figura 1-17** Consumo de Servicio con Angular.js

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

Con el objeto \$http es el que permite acceder a cualquier tipo de servicio rest o resfull, para ello hay que describir los parámetros como la URL que sería la ruta de descripción del consumo, el tipo de servicio en el método (method), en el header se hace referencia el tipo de mensaje que se enviará en este caso es JSON y con dos funciones de tipo call backs se puede realizar procesos tanto cuando fue exitoso como cuando fallo el servicio.

### 1.3 Estándar Clasificación Internacional de Enfermedades CIE 10.

El tener estándares para poder definir los datos enriquece a un sistema informático porque permite cuidar la integridad y llevar un mismo idioma con los sistemas externos, este estándar se basa en las enfermedades que son causales de muerte y los síntomas principales de la mayoría de enfermedades y epidemias desde el año 1893, comprende de un código propio que define la enfermedad y su descripción.

Comprende de 22 capítulos tomados como subdivisiones macros de las enfermedades y cada uno de ellos se subdividen en 30 a 50 enfermedades que se puede diagnosticar, teniendo como un total un aproximado de 1200 enfermedades para seleccionar, a continuación, en la figura 1-18 un fragmento del CIE 10 tomado de una página Excel con la que actualmente trabaja el departamento médico.

46	<b>CAPITULO I</b>
47	<b>CIERTAS ENFERMEDADES INFECCIOSAS Y PARASITARIAS</b>
48	<b>Enfermedades infecciosas intestinales</b>
49	A00 Cólera
50	A01 Fiebres tifoidea y paratifoidea
51	A02 Otras infecciones debidas a Salmonella
52	A03 Shigelosis
53	A04 Otras infecciones intestinales bacterianas
54	A05 Otras intoxicaciones alimentarias bacterianas
55	A06 Amebiasis
56	A07 Otras enfermedades intestinales debidas a protozoarios
57	A08 Infecciones intestinales debidas a virus y otros organismos especificados
58	A09 Diarrea y gastroenteritis de presunto origen infeccioso
59	<b>Tuberculosis</b>
60	A15 Tuberculosis respiratoria, confirmada bacteriológica e histológicamente
61	A16 Tuberculosis respiratoria, no confirmada bacteriológica o histológicamente
62	A17 Tuberculosis del sistema nervioso
63	A18 Tuberculosis de otros órganos
64	A19 Tuberculosis miliar
65	<b>Ciertas zoonosis bacterianas</b>
66	A20 Peste
67	A21 Tularemia
68	A22 Carbunco [ántrax]
69	A23 Brucelosis

**Figura 1-18** Fragmento CIE 10

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

## CAPITULO II

### MARCO METODOLÓGICO

#### 2.1 Metodología Scrum

Durante el desarrollo de software existen problemas al momento de gestionar cada actividad, para ello siempre se sugiere utilizar una metodología la cual vaya de la mano, permitiendo aplicar técnicas para que los tiempos estimados sean más reales y el proyecto se entregue en el tiempo estimado, Scrum es una metodología ágil que permite dividir las tareas en pequeñas actividades conocidas como sprints que generan pequeños entregables al final de cada uno de estos ciclos.

Trabaja directamente con el usuario final permitiéndole verificar y validar cada una de las tareas realizadas por los desarrolladores, en caso de que alguna falle permite una refactorización con el fin de cumplir su funcionamiento correcto, cada rol y estimación en scrum es importante para ello se debe tener en claro las actividades que conlleva cada persona del equipo de trabajo.

##### 2.1.1 Roles de la metodología Scrum

Existen varios roles en la metodología de los que se mencionarán los principales que serán los utilizados en el desarrollo del proyecto, cada uno de ellos tienen un papel fundamental, sea para mediar ideas entre mandos superiores o para validar las actividades realizadas permitiendo entregar en el tiempo estimado cada una de las actividades.

**Product Manager.-** Esta persona es la encargada de plantear los requerimientos, es conocido como usuario experto, es un delegado del cliente para poder expresar cada una de las historias de usuario plasmando como requerimientos y validando cada una de ellas para obtener el resultado deseado, es quien tiene la última palabra y quien pedirá cambios durante el proceso si existieran.

**Scrum Manager.-** Es la persona que trabaja directamente con el product manager, es como el gerente del proyecto que permite la interacción de las ideas plasmadas en las historias de usuario

hacia los desarrolladores, es la primera persona en validar cada requerimiento y permite mediar retrasos o cambios con el usuario final.

**Desarrolladores.** - Son las personas encargadas de plasmar los requerimientos previamente revisados por el scrum manager y probarlos en conjunto realizando pruebas unitarias de aceptación bajo las condiciones que plantee el product manager para validar cada actividad.

En la figura 2-1 se puede observar todo el funcionamiento en conjunto de la metodología en torno a los procesos y las actividades que cada uno desempeña, teniendo en cuenta los procesos establecidos y sus respectivas retroalimentaciones:



**Figura 2-1** Roles Scrum con sus interacciones.

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

## 2.2 Método Deductivo

Mediante este método se aplican los principios descubiertos a casos particulares a partir de la vinculación de juicios. El papel de la deducción en la investigación es doble:

- Primero consiste en encontrar principios desconocidos, a partir de los conocidos. Una ley o principio puede reducirse a otra más general que la incluya. Si un cuerpo cae decimos que pesa porque es un caso particular de la gravitación
- También sirve para descubrir consecuencias desconocidas, de principios conocidos. Si sabemos la fórmula para calcular la velocidad, podremos calcular entonces la velocidad de un avión. La matemática es la ciencia deductiva por excelencia; parte de axiomas y definiciones. (DANIEL, 2008, p39).

En base a todas las teorías aprendidas en la carrera y los proyectos desarrollados en conjunto a metodologías de desarrollo de software se puede aplicar en el proyecto de desarrollo de software, particularizando en base a las necesidades de la empresa Unión Cementera Nacional, y adaptar a la realidad acorde a la observación realizada haciendo que el método deductivo sea más directo, El Mean Stack JavaScript permite la flexibilidad para desarrollar el software, es la herramienta fundamental para plasmar la metodología en el proyecto y hacer que el sistema de gestión se trabaje en conjunto con el usuario final, personalizando su proceso y permitiendo la validación y verificación de cada tarea terminada, menorando el riesgo de fallos y contribuyendo con la calidad al momento de desarrollo.

Durante el proceso es importante comprender cada uno de los requerimientos en base al proceso que realiza es de donde se puede obtener la premisas de los datos conocidos actuales de la empresa como, en este caso se manifestó que se realiza la gestión de las historias de forma manual sin contemplar los errores que conlleva y la pérdida de información que actualmente sufre la dependencia, con esa premisa se puede plasmar la necesidad de desarrollar un sistema informático que gestione la historia clínica para poder mejorar la gestión de la información, en este caso con esta conclusión se puede generar más premisas que sería la necesidad de gestionar el estándar CIE10 para sintomatologías, y tener un control de todos los pacientes y los puestos de trabajo donde se desempeñan.

La Unión Cementera Nacional requiere que se maneje un acceso para que solo el médico ocupacional pueda gestionar los datos de sus pacientes, tomando en cuenta que es información a la que no puede acceder cualquier persona.

## **CAPITULO III**

### **MARCO DE RESULTADOS**

#### **3.1 Análisis de requerimientos**

Durante el levantamiento de requerimientos de software se realizó un proceso de observación y entrevista al Dr. Pulla en la Cemento Chimborazo. En el centro médico trabajan 2 médicos y 3 enfermeras, existen 187 trabajadores directos y 1500 indirectos, por lo que el manejo de la historia clínica ocupacional requiere una ayuda de una herramienta software, que permitirá automatizar el proceso y disminuir la pérdida de información.

Durante la reunión en conjunto con el personal médico en base a la exposición del proceso y preguntas para aclarar documentos que manejan en el departamento médico, se pudo recolectar los requerimientos funcionales y no funcionales.

##### **3.1.1 Requerimientos funcionales**

Este tipo de requerimientos tienen que ver con el funcionamiento intrínseco de la aplicación, permitiendo la interacción directa con el sistema y son los que están plasmando el proceso de los que se tiene la siguiente lista de requerimientos funcionales en la tabla 3-1:

**Tabla 3-1** Requerimientos Funcionales

Código	Requerimiento	Tipo
Req01	Ingreso Paciente	Ingreso
Req02	Modificación Paciente	Modificación
Req03	Autenticación	Consulta
Req04	Ingreso Historia Clínica	Ingreso
Req05	Modificación Historia Clínica	Modificación
Req06	Ingreso CIE 10	Ingreso
Req07	Modificación CIE 10	Modificación

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

### 3.1.2 Requerimientos no funcionales

Este tipo de requerimientos tiene que ver con las funcionalidades que se tuvieron en cuenta en el sistema para el momento que se desee realizar su implantación y para definir la arquitectura y diseño adecuadamente para cumplir cada uno de ellos:

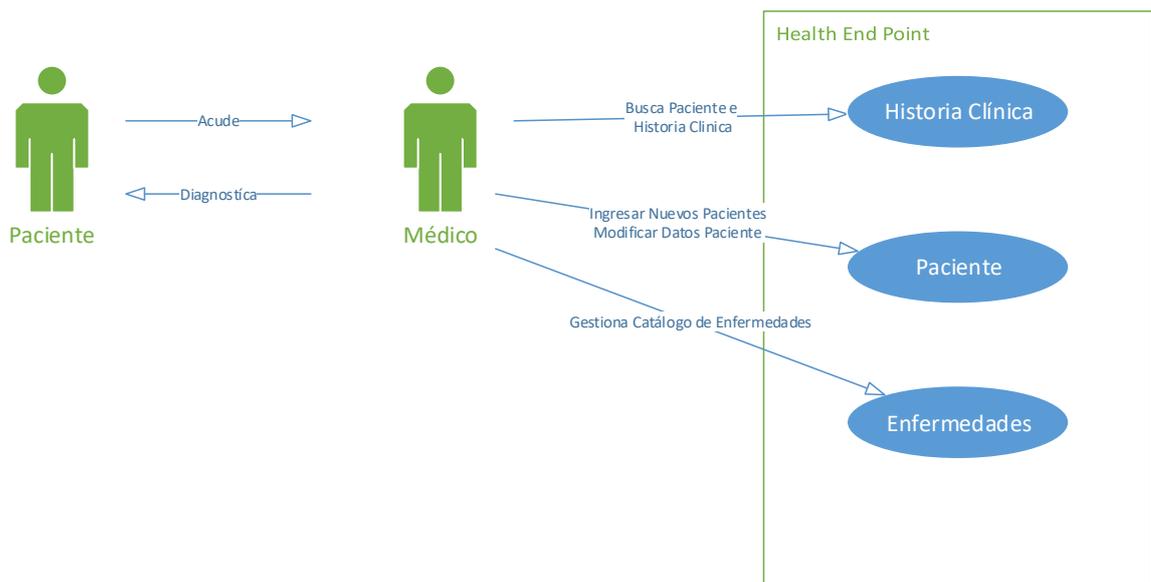
- El sistema siempre se encuentre disponible para que puedan acceder dentro de la institución.
- El sistema se ajuste a cambios de políticas y se adapte con el tiempo a mas tecnologías.
- Cumpla con el Estándar CIE 10 médico para sintomatologías de los pacientes para que esté acorde con el IESS (Instituto Ecuatoriano de Seguridad Social).
- El Sistema se lo pueda ejecutar en cualquier computadora con navegador o en teléfonos celulares o tablets.
- Los usuarios puedan acceder al mismo tiempo desde diferentes dispositivos para poder utilizar la aplicación.
- El diseño Agradable al usuario
- El rendimiento sea bueno ante grandes cantidades de datos
- Sea fácil para dar mantenimiento.

## 3.2 Diseño

Durante el desarrollo de software se tomó en consideración el diseño de todos los componentes que interactúen en la solución, basados en los requerimientos tanto funcionales como no funcionales se plasmaron los siguientes diseños tanto de interacción del sistema como interfaces gráficas

### 3.2.1 Diagrama de procesos.

El proceso explicado por el personal del centro médico durante las reuniones de trabajo se lo representa en un diagrama para poder comprender los flujos de trabajo y de esta forma poder realizar diseños adaptados a la realidad; en la figura 3.1

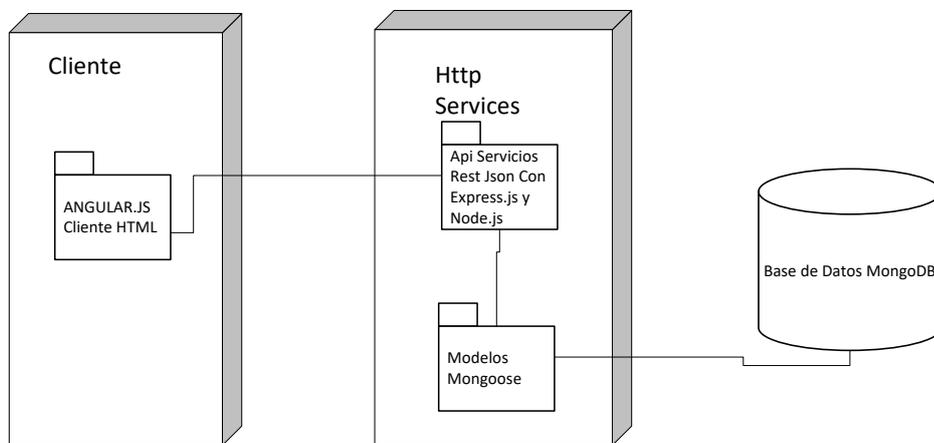


**Figura 3-1** Diagrama de procesos

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

### 3.2.1 Diseño de la Arquitectura

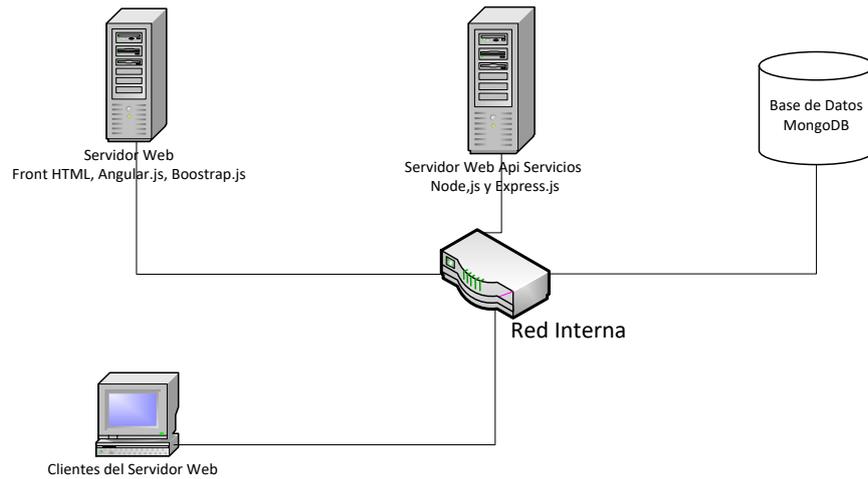
La figura 3-1 describe la arquitectura que del sistema de acuerdo a la teoría SOA( Arquitectura Orientada a Servicios) que implementa módulos publicados en servicios web Rest para que cualquier cliente que desee consumir las funciones y datos no tenga que ser programado desde cero permitiéndole flexibilidad y escalabilidad, el cliente siempre se conecta a los servicios web expuestos para realizar las peticiones pertinentes a la base de datos, dependiendo del modelo que se utilice interactúa directamente con las colecciones y hace que el flujo de datos sea el esperado.



**Figura 3-2** Arquitectura del sistema

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

En este tipo de diseños es muy importante tomar en cuenta la forma adecuada y óptima para que funcione el sistema el cual se conoce como diagrama de despliegue que está presente en la figura 3-2 en el que se observa los dos servidores unidos a un switch en conjunto con el de base de datos que interactúan con los clientes web para realizar las peticiones al sistema.



**Figura 3-3** Diagrama de despliegue de la arquitectura

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

### 3.2.2 Diseño de la base de datos

Durante el diseño se aplicó la metodología de Arrays guardando como referencia los identificadores únicos de tabla y en el caso de relaciones n-n o cardinalidades que requieran encapsular muchos campos, se utilizó esta estructura de datos, el diagrama se encuentra en el ANEXO A 1 y en él se plasman todas las tablas que interactúan en el proceso transaccional tanto de datos de paciente como de la historia clínica, permitiendo tener una visión general de la asociación de datos dentro del proyecto.

### 3.2.3 Diseño de estándares de interfaz de usuario.

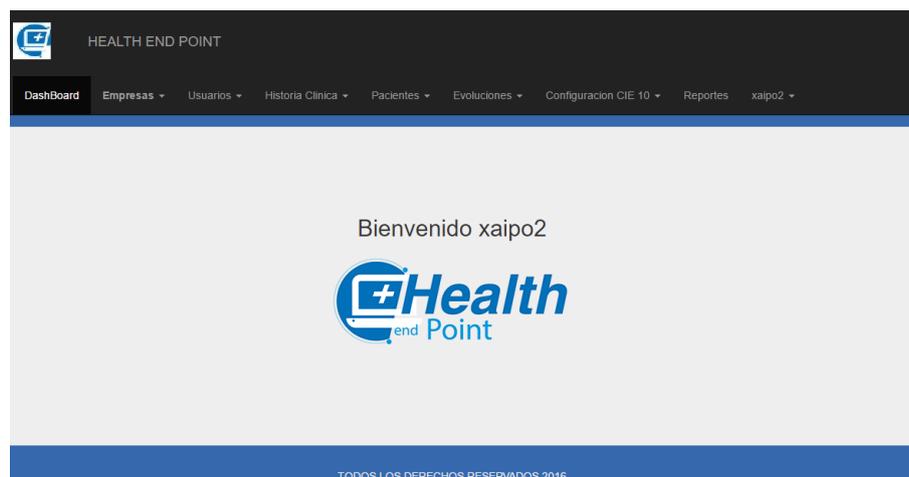
Se tomó en cuenta los colores del logo a utilizar para realizar toda la interfaz de un sistema de salud, que representen el centro médico con tonalidades bajas de azul permitiendo contrastar con colores como gris negro y blanco con los que se pueda crear los formularios de forma agradable a la vista del usuario final.



**Figura 3-4** Logo Sistema Health End Point

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

Los formularios de navegación y tamaños de letra del formulario son por defecto el de Bootstrap.js como en la figura 3-3 y 3-4 que se observa la pantalla de inicio del sistema con los colores antes mencionados y la creación de los formularios con sus respectivas ubicaciones para que se puedan adaptar a cualquier dispositivo



**Figura 3-5** Interfaz gráfica genera Health End Point

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

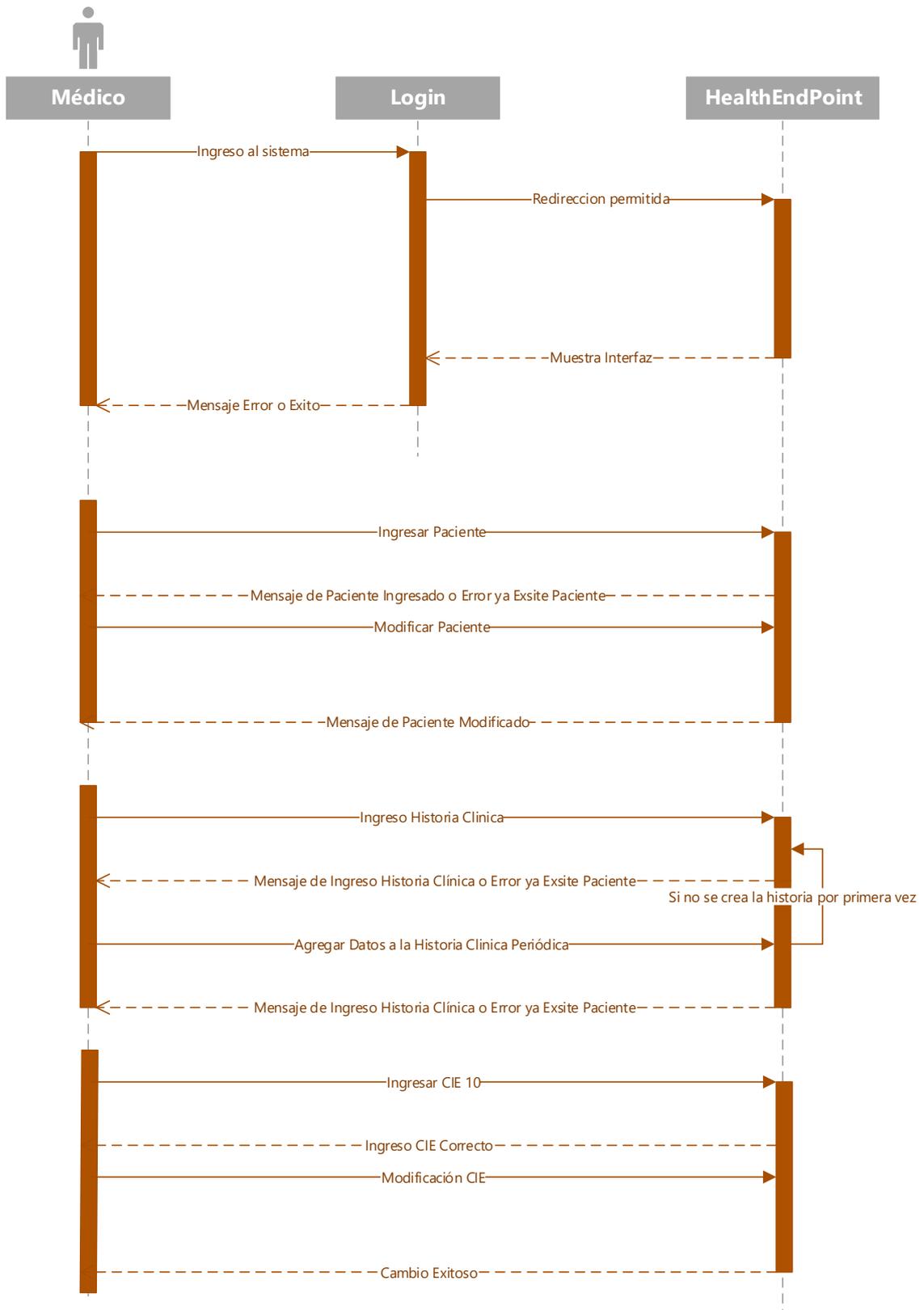
The image shows a web-based form for patient registration. The header includes a logo and the text 'HEALTH END POINT'. Below the header is a navigation menu with items like 'Dashboard', 'Empresas', 'Usuarios', 'Historia Clínica', 'Pacientes', 'Evoluciones', 'Configuración CIE 10', 'Reportes', and 'xaipo2'. The main section is titled 'Ingreso Paciente' and contains two main parts: personal data and job information. The personal data section includes fields for Cédula (ID number), names (First and Second), surnames (First and Second), birth date, sex, province, city, age, marital status, and education level. The job information section is titled 'Ingreso Puesto Trabajo' and includes fields for job description, province, position, start date, machinery, and selected machinery.

**Figura 3-6** Formulario Health End Point con Bootstrap

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

### 3.2.3 Diagrama de Secuencia.

Mediante el diagrama de secuencia se comprende el proceso de interacción de las dependencias entre el usuario final y los módulos del sistema de esta forma permite que al programar se tenga muy en cuenta las entradas y salidas de las diferentes interacciones como procesos globales como se observa en la figura 3-7.



**Figura 3-7** Diagrama de Secuencias

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

### 3.3 Desarrollo

Una vez realizados todos los diseños se puede proceder a plasmarlos dividiéndolos en tareas, y aplicando la metodología propuesta para la gestión del proyecto, se establecieron tiempos adecuados corroborando con el desempeño en cada una de las actividades realizadas.

#### 3.3.1 Tareas de la metodología Scrum:

La metodología que se utilizó en el desarrollo del proyecto es Scrum en la que cada persona ocupa un rol, los más importantes que son los utilizados en el proyecto son el product manager, que es la persona que pide y valida los requerimientos, el scrum manager es la persona que realiza las mediaciones entre el product manager y les informa de las novedades a todo el equipo de trabajo.

Finalmente, los desarrolladores quienes realizan el proyecto y validan las tareas en conjunto con el product manager y el scrum manager haciendo de este procedimiento cíclico, a continuación, la tabla 3-2 muestra los roles de scrum de cada persona en el proyecto.

**Tabla 3-2** Roles de Scrum

Nombre	Rol
Dr. Pablo Pulla	Product Manager
Ing. Patricio Moreno	Scrum Manager
Jairo González, Edison García	Desarrolladores

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

El siguiente paso después de tener bien claro los roles de las personas es la generación de las historias de usuario que no son más que las actividades para llegar a la solución del proyecto, se hicieron 8 historias de usuario una para cada requerimiento teniendo en cuenta sus tiempos los cuales tienen el formato de la tabla 3-3 las demás se encuentran en el ANEXO B, en ellas se encuentra información importante como el número de historia de usuario.

El usuario que realizó dicha historia, en ocasiones se utiliza el rol desempeñado en la empresa, el nombre de la historia que consta con la actividad que se realizó, la prioridad, además de analizar el riesgo que conlleva dentro del proyecto, el responsable es la persona que se encarga de

desarrollar la tarea y siempre debe llevar las horas estimadas en las que se realizara la actividad para luego poder comparar con las horas reales en las que se realizó.

En la descripción el usuario pone lo que quiere que se solucione con esa tarea y finalmente en la validación se describe la funcionalidad del sistema frente a pruebas para poder ser aprobada y continuar con la siguiente tarea.

Las historias se repartieron en base a los módulos que se desarrollaron teniendo en cuenta el esfuerzo que conllevan en base a experiencias pasadas y autoevaluaciones por parte de los desarrolladores.

**Tabla 3-3** Historias de usuario utilizadas en Scrum

<b>HISTORIA DE USUARIO</b>		
<b>NUMERO:</b> 1	<b>USUARIO:</b> Dr Departamento Médico	
<b>NOMBRE HISTORIA:</b> Ingreso Paciente		
<b>PRIORIDAD:</b> ALTA	<b>RIESGO:</b> MEDIO	<b>HORAS ESTIMADAS:</b> 18
<b>ITERACION:</b> 1	<b>RESPONSABLE:</b> EDISON GARCÍA	<b>HORAS REALES:</b> 16
<b>DESCRIPCION:</b>		
Como Dr del departamento médico quiero que la información de los pacientes se almacene acorde a su cédula y el puesto de trabajo en donde se desenvuelven.		
<b>VALIDACION:</b>		
Se ingresará un usuario con sus respectivos datos verificando que se guarde en la base de datos de la forma correcta, al ingresar datos erróneos emitir los respectivos mensajes de error.		

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

Cuando se desee se puede realizar un análisis del proyecto en cuanto a tiempo de trabajo con respecto al trabajo planificado ideal se puede tomar las historias de usuario y listar las horas efectivas en un cuadro como el de la tabla 3-4 para poder realizar los cálculos y mostrar un diagrama denominado burndown chart presente en la figura 24.

**Tabla 3-4** Tiempos por historias de usuario

	TIEMPO ESTIMADO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	TIEMPO REAL TOTAL
1	18	8	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16
2	18	0	4	4	4	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14
3	18	0	0	0	0	6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16
4	40	0	0	0	0	0	2	8	8	8	8	8	1	0	0	0	0	0	0	0	0	0	45
5	25	0	0	0	0	0	0	0	0	0	0	0	7	8	8	7	0	0	0	0	0	0	30
6	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	8	2	0	0	0	0	11
7	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	5	0	0	0	11
8	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	8	8	8	25

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

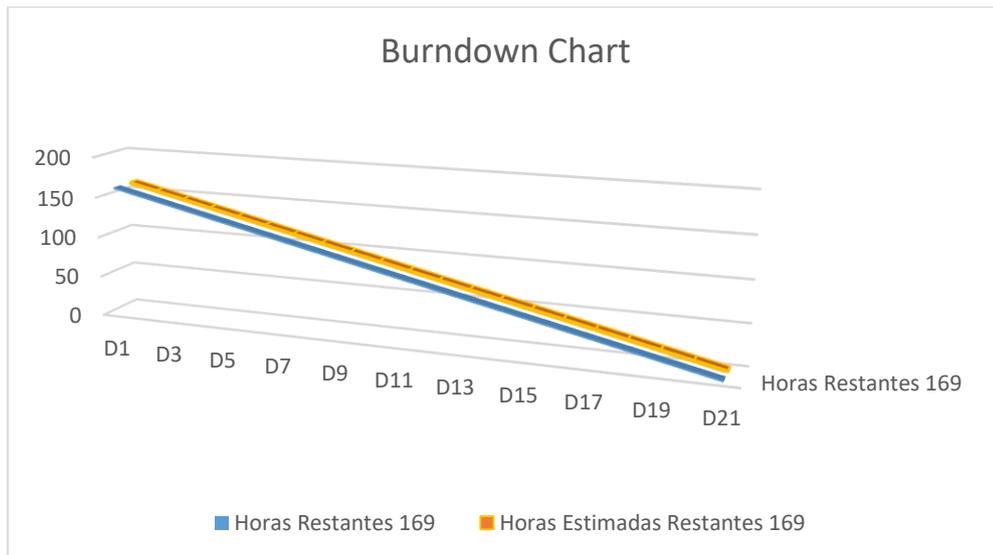
La tabla 3-5 posee los cálculos que son las horas restantes que es la sumatoria de cada columna de los números de días y para sacar el tiempo ideal se divide el tiempo estimado para el número de días del análisis restándole al total de las horas estimadas, ambos tiempos siempre tienden a ser 0 debido a que existen menos horas pendientes.

**Tabla 3-5** Valores calculados de las historia y valores ideales

	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15	D16	D17	D18	D19	D20	D21		
Horas Restantes	169	161	153	145	137	129	121	113	105	97	89	81	73	65	57	49	41	33	25	17	9	1	suma
Horas Estimadas Restantes	169	161.0	152.9	144.9	136.8	128.8	120.7	112.7	104.6	96.6	88.5	80.5	72.4	64.4	56.3	48.3	40.2	32.2	24.1	16.1	8.0	0.0	ideal

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

El en el burndown chart de la figura 3-6 se observa la relación entre las horas restantes del proyecto reales con respecto a las horas estimadas del proyecto, en el eje de las x se encentra la cantidad de días que se ocupan para entregar el proyecto y en el eje y la escala de tiempo con respecto a cada día, si ambas líneas en el grafico están prácticamente una encima de la otra quiere decir que el proyecto se está llevando dentro de lo planificado , caso contrario presentará picos en la parte inferior si esta adelantado y en la parte superior si está retrasado.



**Figura 3-8** *Burndown chart Scrum*

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

### 3.4 Implantación

El sistema se entregó al departamento médico de la Cemento Chimborazo en un CD con los respectivos instaladores para que sean configurados por los técnicos de la empresa en una computadora HP 6300 con procesador i7 y 8 Gb en memoria RAM, un disco duro de 1 TB, una vez que se establecieron los accesos pertinentes en la computadora del médico ocupacional se accedió para realizar pruebas de funcionamiento y funcionalidad que se detallan a continuación.

### 3.5 Pruebas

Las pruebas son un pilar fundamental para la validación de las actividades previas a su implantación o más conocida como el paso a producción, en el documento se plasman las pruebas realizadas tanto en el proceso de desarrollo conocidas como pruebas unitarias como las de rendimiento para que el sistema funcione a toda su capacidad.

### 3.5.1 Pruebas Unitarias

Durante el desarrollo se validó cada actividad para cumplir el requerimiento por ello, manejar pruebas unitarias es una alternativa para ver el desempeño de cada módulo para poder controlar errores, tomando en cuenta este precedente a continuación un ejemplo de prueba de aceptación en la Tabla 3-6:

**Tabla 3-6** Pruebas unitarias de cada requerimiento

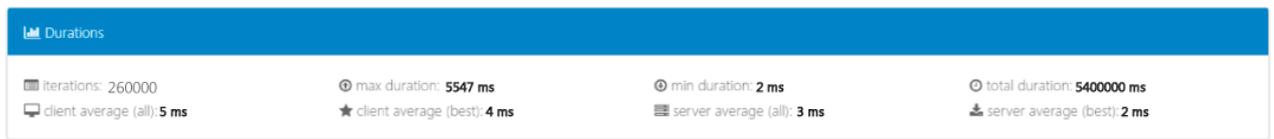
<b>Prueba : 1</b>
<b>Descripción:</b> Prueba de Ingreso a Paciente
<b>Condiciones:</b>  Se realiza bajo el ambiente de desarrollo, se ingresarán datos correctos e incorrectos
<b>Resultado Esperado:</b>  Que ingrese en la base de datos y si existen datos mal ingresados emitan las alertas
<b>Resultado Obtenido:</b>  Una parte de los datos se perdió en el proceso de ingreso debido a la interfaz se debe refactorizar y en cuanto a las alertas funcionan bien

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

En cada una de ellas se especificó bajo qué ambiente se realizaron las pruebas y los resultados obtenidos junto con los esperados, de esta forma se pueden corregir errores durante el proceso de desarrollo de software todas las demás pruebas adjuntas en el ANEXO C.

### 3.5.2 Pruebas Rendimiento.

Para probar la tecnología una carga de estrés para notar la velocidad de respuesta de los servicios web con respecto a la cantidad de datos ingresados, se realizaron 260000 iteraciones de envío al servicio de pacientes teniendo los resultados en la a figura 3-8 con la herramienta RestFul Stress:



**Figura 3-9** Función de test Rest

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

Durante la carga de 206000 datos de la tabla paciente se ejecutaron en 1 hora 35 minutos que equivalen a 5400000 milisegundos a través de un servicio web, cabe recalcar que es bajo condiciones normales con una computadora con procesador core i7 de sexta generación 4Gb de memoria RAM y en un solo hilo de proceso teniendo como mayor tiempo de respuesta 5547 milisegundos y el promedio de 4 milisegundos por parte del cliente y 3 milisegundos por parte del servidor con una mejora del 80% con respecto a los servicios SOAP que se muestra en la figura 3109 con respecto a la 3-9.



**Figura 3-10** Función de test SOAP

**Realizado por:** GONZALEZ, Jairo; GARCÍA, Edison.2016

### 3.5.3 Pruebas Usabilidad.

La evaluación de la usabilidad del sistema se la realizó mediante un test en el cual se evaluó identidad, contenido, navegación, utilidad y retroalimentación.

**Tabla 3-7:** Identidad

<b>Identidad</b>	<b>Si</b>	<b>No</b>
¿Se plasma en la pantalla de inicio la identidad corporativa de la empresa expresada en colores?	1	0
¿Hay algún elemento gráfico o de texto que le haya ayudado a entender más claramente a qué empresa pertenece?	1	0
¿El sistema cumple con todos los requerimientos exigidos por el departamento médico?	1	0
¿De los elementos que muestra esta pantalla, hay algo que usted crea que no está fuera de contexto?	1	0
<b>TOTAL:</b>	<b>5</b>	<b>0</b>

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

**Análisis:** Se capacitó a la persona que utilizará el sistema y en base a lo respondido se obtiene que 4 de 4 preguntas que posee el sistema fueron identificadas de forma fácil, lo que equivale a un 100 %, y evidencia que cumple el sistema con identificarse con la empresa.

**Tabla 3-8:** Contenido

<b>Contenido</b>	<b>Si</b>	<b>No</b>
¿La aplicación cumple con los contenidos previstos?	1	0
¿Al ver el sistema, pudo distinguir de una sola mirada cuál era el contenido más relevante que se ofrecía?	1	0
¿Se identifica de forma clara el ingreso de datos correcto?	1	0
¿Los datos a ser ingresados son suficientemente descriptivos?	1	0
¿No se repite información en el sistema?	1	0
<b>TOTAL:</b>	<b>5</b>	<b>0</b>

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

**Análisis:** En la tabla se tiene un total de 5 respuestas positivas lo que equivale al 100% del cumplimiento del contenido previsto fácil de entender y descriptivo para el usuario, se plasma que el sistema se realizó de forma personalizada y en conjunto con el usuario.

**Tabla 3-9: Utilidad**

Utilidad	Si	No
¿Al observar la pantalla de inicio se puede avizorar de lo que trata el sistema? y ¿Qué ofrece?	1	0
¿Cree que los contenidos y servicios que se ofrecen en este sistema son de utilidad para su caso personal?	1	0
<b>TOTAL:</b>	2	0

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

**Análisis:** La utilidad del sistema se basa en la solución de problemas que se plante solucionar durante el proceso de facturación por ello tiene una valoración del 100% de cumplimiento

**Tabla 3-10: Navegación**

Navegación	Si	No
¿La forma en que se navega por el sistema y sus opciones, es fácil de realizar?	1	0
¿Existen confusión de elementos que entorpezcan la navegación?	0	1
¿Logra distinguir gráficamente los datos que ya han sido registrados?	1	0
<b>TOTAL:</b>	2	1

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

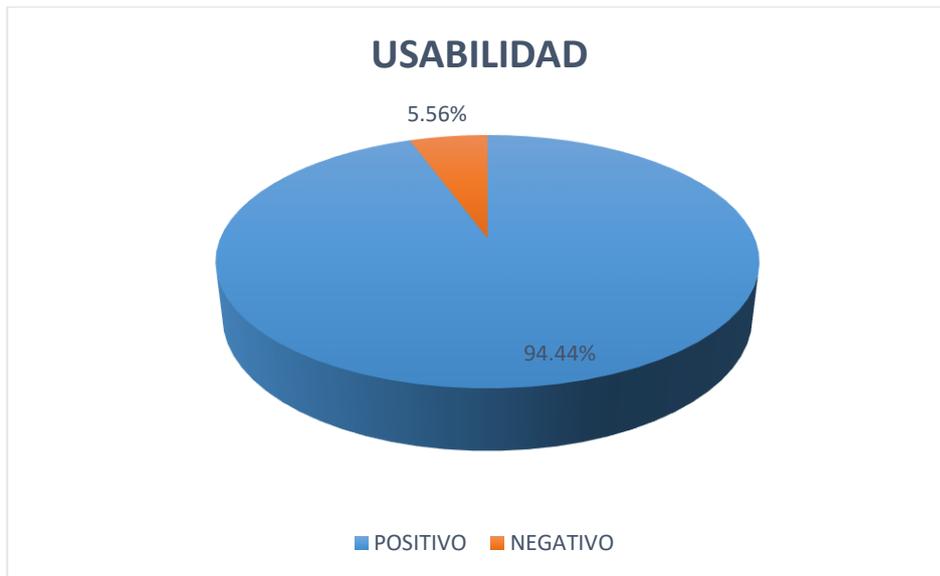
**Análisis:** En cuanto a navegación se muestra un 66.66% lo que es bastante bueno para cumplir dicho criterio permitiendo al usuario poder desempeñar las funciones con el software de forma satisfactoria.

**Tabla 3-11: Retroalimentación**

Retroalimentación	Si	No
¿La aplicación cumple con sus expectativas?	1	0
¿Al momento de registrar los datos en el sistema ¿Ocurrió algún error?	0	1
¿Al momento de generar un comprobante electrónico en el sistema ¿Ocurrió algún error?	0	1
<b>TOTAL:</b>	1	2

Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

**Análisis:** Con las respuestas de estas preguntas se nota que cumple con el 100% de aceptación el sistema en cuanto a retroalimentación.



**Figura 3-11** Análisis usabilidad  
Realizado por: GONZALEZ, Jairo; GARCÍA, Edison.2016

**Análisis:** Tomando en cuenta los factores positivos y negativos identificados en las encuestas se obtuvieron 17/18 factores positivos en cuanto usabilidad y solo un 1 factor negativo por lo que el sistema cumple con los parámetros básicos de usabilidad para el usuario.

## CONCLUSIONES

- Al realizar el test de usabilidad en base a preguntas de navegación, identidad, contenido, usabilidad, utilidad y retroalimentación se llegó a un 94.44% de aspectos positivos determinando un alto índice de usabilidad y experiencia del usuario al momento de utilizar el sistema.
- Los servicios basados en JSON sometidos a test resultaron 80% más eficientes en cuanto a respuestas y procesamiento de información con respecto a los servicios web SOAP, que tienen como base XML lo que determina que son muy eficientes para grandes cantidades de datos.
- El sistema permitió apoyar en la gestión de los datos de la historia clínica de forma satisfactoria permitiendo cuidar la integridad de los datos con el estándar CIE 10 de enfermedades, prestando una universalización de sintomatologías gracias al código y la descripción únicos de cada síntoma.
- El manejo de tecnologías como Bootstrap permite que el sistema sea responsive y se adapte a todo tipo de resolución de pantalla haciéndolo que funcione de forma multidispositivo, incrementando las oportunidades de uso en cuanto a la herramienta cliente del usuario final que puede ser una pc, Tablet o teléfono móvil.
- EL implantar una arquitectura orientada a servicios brinda flexibilidad a cambios y permite exponer servicios para otros sistemas que deseen integrarse a la plataforma con solo conceder los accesos pertinentes, tomando en cuenta las leyes que rigen el país actualmente.

## RECOMENDACIONES

- Es mejor utilizar Express en su versión 4.0 debido a las prestaciones y la estructura flexible que tiene tanto para definir las rutas de los servicios web como para agregar librerías y módulos específicas para esa versión.
- Se recomienda agregar seguridades con token OAUTH para consumir los servicios web y administrar el login en la aplicación, ya que al tener el token se puede utilizar varios criterios de autenticación y permisos a las partes futuras que puedan ser desarrolladas en la aplicación y realizar un trabajo conjunto con encriptaciones de los datos más importantes.
- Se recomienda desarrollar un módulo administrable para las demás tablas permitiendo configurar cada uno de los tipos e incrementar la flexibilidad por parte del usuario final, con ello al existir cambios futuros dentro de los configurables se puede adaptar el sistema.
- Se puede incluir al sistema módulos de inteligencia artificial para investigación permitiéndole realizar minería de datos y convertirse en un sistema experto, de esta forma se puede contribuir con esta información a planteles educativos, así como tomar decisiones para realizar campañas preventivas de salud por sectores de la empresa.
- Se recomienda incrementar los módulos de ausentismo y morbilidad para que trabaje en conjunto con la historia clínica ocupacional para que de esta forma el sistema se fortalezca con reportes para los médicos ocupacionales.
- Es mas recomendable utilizar los archivos minificados de angular.js y jquery.js debido a que el uso de CDN requiere el uso permanente de internet al ser servidores ajenos a la aplicación y no depende de la disponibilidad del servidor para ello mejor referenciar a la aplicación a que utilice archivos locales en el servidor que se implanta.

## **BILIOGRAFIA**

**ARGENTINA. Ministerio de Salud Argentina.** Definición Salud Ocupacional. Buenos Aires-Argentina.2015. [En línea]

[ Consulta: 5 marzo del 2015]

<http://www.msal.gov.ar/index.php/home/salud-ocupacional>

**BRETT MacLaughlin.** What is Node.js. 1ª ed. California-Estados Unidos,O'Reilly .2011. pp 15-200.

**BRETZ Adam; & IHRIG Colin.** Full Stack JavaScript develop with Mean. 1ª ed. Pittsburgh-Estados Unidos. Sitepoint .2014. pp 20-305.

**BROWN Ethan.** Web Development with Node and Express. 1ª ed. California-Estados Unidos. O'Reilly. 2014. pp 500-730.

**DE LA TOREE Cesar.** Arquitectura SOA con tecnología Microsoft. 1ª ed. Madrid-España. Kransis Press. 2008. pp 4-32.

**DE LUCA Damián.** Apps HTML 5 para móviles. 1ª ed. Buenos Aires-Argentina. Alfaomega. 2016. pp 5-220.

**ECUADOR. UNIÓN CEMENTERA NACIONAL (UCEM).** Misión y visión. Riobamba-Ecuador.2015. [En línea]

[ Consulta: 5 marzo del 2015]

[http://www.cementochimborazo.com/index.php?option=com\\_content&view=article&id=50&Itemid=64](http://www.cementochimborazo.com/index.php?option=com_content&view=article&id=50&Itemid=64)

**ECUADOR. MINISTERIO DEL TRABAJO.** Código del trabajo. Quito-Ecuador.2015.

[En línea]

[Consulta: 3 enero del 2015]

<http://www.trabajo.gob.ec/wp-content/uploads/downloads/2012/11/C%C3%B3digo-de-Trabajo-PDF.pdf>

**ECUADOR. MINISTERIO DEL TRABAJO.** Reglamento Salud Ocupacional. Quito-Ecuador.2015. [En línea]

[Consulta: 3 enero del 2015]

<http://www.trabajo.gob.ec/wp-content/uploads/downloads/2012/12/Reglamento-para-el-Funcionamiento-de-Servicios-M%C3%A9dicos-Acuerdo-Ministerial-1404.pdf>

**ESTADOS UNIDOS. ANGULAR JS.** Documentación Angular.js. California-Estados Unidos. 2015. [En línea]

[ Consulta: 9 agosto del 2015]

<https://docs.angularjs.org/guide/concepts>

**ESTADOS UNIDOS. API NODE RESFUL.** Documentación api node restfull. California-Estados Unidos. 2015. [En línea]

[ Consulta: 15 noviembre del 2015]

[https://www.tutorialspoint.com/nodejs/nodejs\\_restful\\_api.htm](https://www.tutorialspoint.com/nodejs/nodejs_restful_api.htm)

**ESTADOS UNIDOS. CODECADEMY.** Cursos AngularJS, California-Estados Unidos. 2015. [En línea]

[ Consulta: 9 marzo del 2015]

<https://www.codecademy.com/learn/learn-angularjs>

**ESTADOS UNIDOS. CODESCHOOL.** Became Mean Stack JavaScript. California-Estados Unidos. 2015. [En línea]

[ Consulta: 10 abril del 2015]

<https://www.codecademy.com/learn/learn-angularjs>

**ESTADOS UNIDOS. EXPRESS JS.** Documentación y Manuales ExpressJS. California-Estados Unidos. 2015. [En línea]

[ Consulta: 10 junio del 2015]

<http://expressjs.com/en/guide/routing.html>

**ESTADOS UNIDOS. MEAN.IO.** Documentation, California-Estados Unidos. 2015. [En línea]

[ Consulta: 10 mayo del 2015]

<http://learn.mean.io/>

**ESTADOS UNIDOS. MEAN.JS.** Documentación Mean Stack. California-Estados Unidos. 2015. [En línea]

[ Consulta: 10 diciembre del 2015]

<https://meanjs.org/docs.html>

**ESTADOS UNIDOS. MICROSOFT VIRTUAL ACADEMY.** Tutoriales Jump Mean Stack JavaScript. California-Estados Unidos. 2015. [En línea]

[ Consulta: 6 marzo del 2016]

, [https://mva.microsoft.com/en-us/training-courses/mean-stack-jump-start-8442?l=eovSb3Vz\\_4604984382](https://mva.microsoft.com/en-us/training-courses/mean-stack-jump-start-8442?l=eovSb3Vz_4604984382)

**ESTADOS UNIDOS. MONGO DB.** Documentación MongoDB. California-Estados Unidos. 2015. [En línea]

[ Consulta: 15 abril del 2015]

<https://docs.mongodb.com/manual/>

**ESTADOS UNIDOS. NODE JS.** Documentación Node.js. California-Estados Unidos. 2015. [En línea]

[ Consulta: 20 agosto del 2015]

<https://nodejs.org/en/docs/>

**ESTADOS UNIDOS. UDEMY.** Curso Mean Stack JavaScript. California-Estados Unidos. 2015. [En línea]

[ Consulta: 15 agosto del 2015]

<https://www.udemy.com>

**DEYLEY Brad.** Write Modern Web Apps with the MEAN Stack. 1ª ed. California-Estados Unidos. Peachpit Press. 2014. pp 50-320.

**DICKEY Brad.** Node.js, MongoDB and AngularJS Web Development. 1ª ed. Boston-Estados Unidos. Addison-Wesley. 2014. pp 100-205.

**FORBES Alan.** The joy of bootstrap. 2ª ed. Boston-Estados Unidos. Plum Island Publishing. 2014. pp 20-100.

**HERNÁNDEZ Sampieri.** Metodología de la Investigación. Buenos Aires-Argentina. 2015.

[En línea]

[ Consulta: 5 marzo del 2015]

<http://www.msal.gov.ar/index.php/home/salud-ocupacional>

**HERNÁNDEZ Rick.** MEAN.IO Guide Building Real World Applications From Scratch. 1ª ed. California-Estados Unidos. Lenapub .2015. pp 205-508.

**KURNIAWAN Agus.** AngularJS Programming by Example. 1ª ed Berlin-Alemania. Lenapub. 2014. pp 10-400.

**KROL Jason.** Anatomy of a web application using node.js, ExpressJS, MongoDB & Backbone.js. 1ª ed. Teneese-Estados Unidos. Lenapub. 2015. pp 10-100.

**ORNBO, GEORGE.** Node.JS. 1ª ed. Madrid-España. GRUPO ANAYA. 2013. pp 5-200.

**OZ, Effy.** Administración de sistema de Información. 2.ª ed. Pensylvania-Estados Unidos. Thomson Learning. 2001. pp 5-300.

**PRESSMAN Roger.** Software Engieneerin. 7ª ed. Boston-Estados Unidos. MacGraw-Hill. 2012. pp 5-300.

**SUDARSHAN Subrao.** Fundamentos de base de datos. 1ª ed. Madrid-España. MacGraw-Hill. 2013. pp 4-26.



**ANEXO A 2: Diccionario de Datos**

<b>Tabla</b>	<b>Campo</b>	<b>Tipo de Dato</b>	<b>PK</b>	<b>FK</b>
<b>Paciente</b>	Cedula	Texto	x	
	PuestoTrabajo	ObjectID		x
	PrimerNombre	Texto		
	SegundoNombre	Texto		
	PrimerApellido	Texto		
	SegundoApellido	Texto		
	Sexo	Texto		
	FechaNacimiento	Texto		
	Ciudad	ObjectID		x
	Edad	Numero		
	Telefono	Texto		
	EstadoCivil	ObjectID		x
	NivelEstudio	ObjectID		x
<b>Historia Clinica</b>	Paciente	ObjectID		x
	Codigo	ObjectID	x	
	TipoHistoriaClinica	ObjectID		x
	Proceso	Texto		
	RiesgosOcupacionales	Array ObjectID		x
	Ausentismo	Array ObjectID		x
	AccidentesTrabajo	Array ObjectID		x
	EnfermedadesActualesHistoricas	Array ObjectID		x
	AntecedentesSalud	Array ObjectID		x
	GinecoObstetra	ObjectID		x
	RevisionSistemas	Array ObjectID		x
	Inmunizacion	Array ObjectID		x
	HabitosToxicos	Array ObjectID		x
	ExamenFisico	Array ObjectID		x
	OrganosSistemaSeleccionado	Array ObjectID		x
ExamenesLaboratorioPracticados	Array ObjectID		x	

	ExámenesParaclínicos	Array ObjectID		x
	SospechaEnfermedadOcupacional	Texto		
	DescripciónSospecha	Texto		
	AptitudCargo	Booleana		
	Recomendaciones	Array ObjectID		x
	Limitaciones	Array ObjectID		x
	FechaExamen	Texto		
	Estado	Boolean		
<b>Recomendaciones</b>	CódigoRecomendaciones	ObjectID	x	
	DescripciónRecomendaciones	Texto		
<b>Limitaciones</b>	CódigoLimitaciones	ObjectID	x	
	Descripción	Texto		
<b>Tipo Historia</b>	Código	ObjectID	x	
	Descripción			
<b>Empresa</b>	CódigoEmpresa	ObjectID	x	
	NombreEmpresa	Texto		
<b>Dependencia</b>	CódigoDependencia	ObjectID	x	
	Empresa	ObjectID		x
	NombreDependencia			
<b>PuestoTrabajo</b>	CódigoPuestoTrabajo	ObjectID	x	
	Dependencia	ObjectID		x
	Protocolos	Array ObjectID		x
	Protección	Array ObjectID		x
	MateriaPrima	Array ObjectID		x
	Herramientas	Array ObjectID		x
	Maquinarias	Array ObjectID		x
	NombrePuestoTrabajo			
	Estado	Texto		
	Jornada	ObjectID		x
	Fecha	Texto		
	PuestoTrabajo	Texto		
<b>Cargo</b>	CódigoCargo	ObjectID	x	

	NombreCargo	Texto		
<b>Jornada</b>	CodigoJornada	ObjectID	x	
	NombreJornada	Texto		
<b>Maquinaria</b>	CodigoMaquinaria	ObjectID	x	
	DescripcionMaquinaria	Texto		
<b>Herramientas</b>	CodigoHerramienta	ObjectID	x	
	Descripcion	Texto		
<b>MateriaPrima</b>	Codigo]MateriaPrima	ObjectID	x	
	Descripcion	Texto		
<b>Proteccion</b>	CodigoProteccion	ObjectID	x	
	Descripcion	Texto		
<b>Protocolos</b>	CodigoProtocolos	ObjectID	x	
	Descripcion	Texto		
	Preguntas	Array ObjectID		x
<b>Preguntas</b>	CodigoPregunta	ObjectID	x	
	Preguntas	Text		
	RespuestasSeleccionadas	Array ObjectID		
<b>Provincias</b>	CodigoProvincia	ObjaectID	x	
	Descripcion	Text		
<b>Ciudad</b>	CodigoCiudad	ObjectID	x	
	Descripcion	Text		
	Provincia	ObjectID		x
<b>EstadoCivil</b>	CodigoEstadoCivil	ObjectID	x	
	Descripcion	Text		
<b>NivelEstudio</b>	CodigoNivelEstudio	ObjectID	x	
	Descripcion	Text		
<b>RiesgosOcupacionales</b>	CodigoRiesgo	ObjectID	x	
	NombreEmpresa	Texto		
	CargoEmpresa	Texto		
	Actividades	Texto		

	TipoActividad	ObjectID		x
	TipoAñosExposicion	Numerico		
	FactoresRiesgo	Array ObjectID		x
	Cualificacion	Array ObjectID		x
	Alimentos	Array ObjectID		x
	Sintomatología Individual	Texto		
	Sintomatología Grupal	Texto		
	EPP	Texto		
	AccidentesTrabajo	Array ObjectID		x
	EnfermedadesActualesHistoricas	Array ObjectID		x
<b>TipoActividad</b>	CodigoTipoActividad	ObjectID	x	
	Descripcion	Texto		
<b>FactoresRiesgo</b>	CodigoFactoresRiesgo	ObjectID	x	
	Descripcion	Texto		
<b>Cualificacion</b>	CodigoCualificacion	ObjectID	x	
	Descripcion	Texto		
<b>Alimentos</b>	CodigoAlimentos	ObjectID	x	
	Descripcion	Texto		
<b>AccidentesTrabajo</b>	CodigoAccidente	ObjectID	x	
	Fecha	Texto		
	NombreEmpresa	Texto		
	Naturaleza Lesion	Texto		
	ParteCuerpo	Texto		
	DiasIncapacidad	Numerico		
	Secuelas	Texto		
<b>EnfermedadesActualesHistoricas</b>	CodigoEnfermedades	ObjectID	x	
	Descripcion	Text		
	EfermedadesCIE10	Array ObjectID		x
	Diagnostico	Texto		
	Fecha	Texto		
<b>CIE 10</b>	CodigoCIE10	ObjectID	x	
	Codigo	Texto		
	Sintomas	Texto		
	TipoCIE10	ObjectID		

	TratamientoCIE10	Array ObjectID		x
<b>TipoCIE10</b>	CodigoTipoCIE10	ObjectID		
	Descripcion	Texto		
<b>TratamientoCIE10</b>	CodigoTratamientoCIE10	ObjectID	x	
	Descripcion	Texto		
<b>Ausentismo</b>	CodigoAusentismo	ObjectID	x	
	Causa	Texto		
	Tiempo	Texto		
	EnfermedadCIE10	Array ObjectID		x
<b>AntecedentesSalud</b>	CodigoAntecedentes	ObjectID	x	
	Familiares	Array ObjectID		x
	Personales	Array ObjectID		x
<b>Familiares</b>	CodigoFamiliares	ObjectID	x	
	PatologiaCIE10	ObjectID		x
	Parentezco	ObjectID		x
<b>Personales</b>	CodigoPersonales	ObjectID	x	
	EnfermedadCIE10	ObjectID		x
	TipoPersonales	ObjectID		x
	Observaciones	Texto		
<b>Parentezco</b>	CodigoParentezco	ObjectID	x	
	Descripcion	Text		
<b>TipoPersonales</b>	CodigoTipoPersonales	ObjectID	x	
	Descripcion	Text		
<b>GinecoObstetra</b>	CodigoGinecoObstetra	ObjectID	x	
	Normal_Anormal	Texto		
	FechaUltimaRegla	Texto		
	Gestaciones	Boolean		x
	Partos	Boolean		
	Abortos	Boolean		
	HijosVivos	Boolean		
	Embarazos	Boolean		
	FechaUltimaCitologia	Texto		
	ResultadosCitologia	Texto		

	PlanificacionFamiliar	Boolean		
	MetodosPlanificacion	Array ObjectID		x
	Observaciones	Texto		
<b>MetodosPlanificacion</b>	CodigoMetodoPlanificacion	ObjectID	x	
	Nombre	Texto		
<b>RevisionSistemas</b>	CodigoRevisionSistemas	ObjectID	x	
	Descripcion	Texto		
	TipoRevisionSistemas	Array ObjectID		x
	Observaciones	Texto		
<b>TipoRevisionSistemas</b>	CodigoTipoRevision	ObjectID	x	
	Descripcion	Texto		
<b>Inmunizacion</b>	CodigoInmunizacion	ObjectID	x	
	Vacuna	ObjectID		x
	Fecha	Texto		
	Dosis	Numerico		
<b>Vacuna</b>	CodigoVacuna	ObjectID	x	
	NombreVacuna	Texto		
<b>HabitosToxicos</b>	CodigoHABitoToxico	ObjectID	x	
	TipoHabitoToxico	ObjectID		x
	TipoConsumidor	ObjectID		x
	Frecuencia	Texto		
	Años de Consumo	Numerico		
<b>TipoHabitoToxico</b>	CodigoTipoHabito	ObjectID	x	
	Descripcion	Texto		
<b>TipoConsumidor</b>	CodigoTipoConsumidor	ObjectID	x	
	Descripcion	Texto		
<b>ExamenFisico</b>	CodigoExamenFisico	ObjectID	x	
	TensionArterial	Numeric		
	FrecuenciaCardiaca	Numeric		
	FrecuenciaRespiratoria	Numeric		
	Dieztro	Boolean		
	Zurdo	Boolean		
	Ambidiestro	Boolean		
	Talla	Numeric		
	Peso	Numeric		
	IndiceMasaCorporal	Numeric		

	Interpretacion IMC	ObjectID		x
<b>Interpretacion IMC</b>	CodigoInterpretacionIMC	ObjectID	x	
	Descripcion	Text		
<b>OrganosSistemaSeleccionado</b>	CodigoOrganosSistema	ObjectID	x	
	SistemaOrgano	Array ObjectID		x
	Normal_Anormal	Texto		
	Hallazgo	Texto		
<b>SistemaOrgano</b>	CodigoSistemaOrgano	ObjectID	x	
	Descripcion	Texto		
	Array ObjectID	ObjectID		
<b>TipoOrgano</b>	CodigoTipoOrgano	ObjectID	x	
	Descripcion	Texto		
<b>ExámenesLaboratorioPracticados</b>	CodigoExámenes	ObjectID	x	
	ExámenesLaboratorio	ObjectID		x
	Normal	Texto		
	Fecha	Texto		
	Observaciones	Texto		
<b>ExámenesLaboratorio</b>	CodigoExamenLaboratorio	ObjectID	x	
	Descripcion	Texto		
<b>ExámenesParaclínicos</b>	CodigoParaclínicos	ObjectID		
	ResultadoParaclinico	Array ObjectID		x
	Observaciones	Texto		
	ImpresionDiagnosticoCIE10	Array ObjectID		x
	NombreExamen	Texto		
<b>ResultadoParaclinico</b>	CodigoResultadoParaclinico	ObjectID	x	
	Descripcion	Texto		

**ANEXO B: Historias de Usuario Scrum**

<b>HISTORIA DE USUARIO</b>		
<b>NUMERO:</b> 1	<b>USUARIO:</b> Dr Departamento Médico	
<b>NOMBRE HISTORIA:</b> Ingreso Paciente		
<b>PRIORIDAD:</b> ALTA	<b>RIESGO:</b> MEDIO	<b>HORAS ESTIMADAS:</b> 18
<b>ITERACION:</b> 1	<b>RESPONSABLE:</b> EDISON GARCÍA	<b>HORAS REALES:</b> 16
<b>DESCRIPCION:</b>  Como Dr. del departamento médico quiero que la información de los pacientes se almacene acorde a su cédula y el puesto de trabajo en donde se desenvuelven.		
<b>VALIDACION:</b>  Se ingresará un usuario con sus respectivos datos verificando que se guarde en la base de datos de la forma correcta, al ingresar datos erróneos emitir los respectivos mensajes de error.		

<b>HISTORIA DE USUARIO</b>		
<b>NUMERO:</b> 2	<b>USUARIO:</b> Dr Departamento Médico	
<b>NOMBRE HISTORIA:</b> Modificación Paciente		
<b>PRIORIDAD:</b> ALTA	<b>RIESGO:</b> MEDIO	<b>HORAS ESTIMADAS:</b> 18
<b>ITERACION:</b> 1	<b>RESPONSABLE:</b> EDISON GARCÍA	<b>HORAS:</b> 14
<b>DESCRIPCION:</b>  Como Dr del departamento médico quiero modificar la información de los pacientes se almacenada acorde a su cédula y el puesto de trabajo en donde se desenvuelven.		
<b>VALIDACION:</b>  Se modificará un usuario con sus respectivos datos verificando que se guarde en la base de datos de la forma correcta, al ingresar datos erróneos emitir los respectivos mensajes de error.		

<b>HISTORIA DE USUARIO</b>		
<b>NUMERO:</b> 3	<b>USUARIO:</b> Dr Departamento Médico	
<b>NOMBRE HISTORIA:</b> Autenticación		
<b>PRIORIDAD:</b> ALTA	<b>RIESGO:</b> MEDIO	<b>HORAS ESTIMADAS:</b> 18
<b>ITERACION:</b> 1	<b>RESPONSABLE:</b> JAIRO GONZÁLEZ	<b>HORAS:</b> 16
<p><b>DESCRIPCION:</b></p> <p>Como Dr. del departamento médico quiero ingresar por medio un nombre de usuario y contraseña al sistema para cuidar la seguridad</p>		
<p><b>VALIDACIÓN:</b></p> <p>Ingrese si es correcto si no es correcto muestre el mensaje de error.</p>		

HISTORIA DE USUARIO		
<b>NUMERO:</b> 4	<b>USUARIO:</b> Dr Departamento Médico	
<b>NOMBRE HISTORIA:</b> Ingreso Historia Clínica		
<b>PRIORIDAD:</b> ALTA	<b>RIESGO:</b> MEDIO	<b>HORAS ESTIMADAS:</b> 40
<b>ITERACION:</b> 1	<b>RESPONSABLE:</b> JAIRO GONZÁLEZ	<b>HORAS:</b> 45
<b>DESCRIPCION:</b>  Como Dr del departamento médico quiero ingresar la historia clínica al sistema con sus respectivos controles		
<b>VALIDACION:</b>  Ingreso correcto de la Historia Clínica de pasar algo en el proceso mostrar un mensaje de error.		

HISTORIA DE USUARIO		
<b>NUMERO:</b> 5	<b>USUARIO:</b> Dr Departamento Médico	
<b>NOMBRE HISTORIA:</b> Modificación Historia Clínica		
<b>PRIORIDAD:</b> ALTA	<b>RIESGO:</b> MEDIO	<b>HORAS ESTIMADAS:</b> 25
<b>ITERACION:</b> 1	<b>RESPONSABLE:</b> JAIRO GONZÁLEZ	<b>HORAS:</b> 30
<b>DESCRIPCION:</b>  Como Dr. del departamento médico quiero modificar solo datos respecto a exámenes históricos y evolutivos que se presenten con el tiempo		
<b>VALIDACION:</b>  Modificación Correcta de la Historia clínica de no emitir un mensaje de error.		

<b>HISTORIA DE USUARIO</b>		
<b>NUMERO:</b> 6	<b>USUARIO:</b> Dr Departamento Médico	
<b>NOMBRE HISTORIA:</b> Ingreso CIE 10		
<b>PRIORIDAD:</b> ALTA	<b>RIESGO:</b> MEDIO	<b>HORAS ESTIMADAS:</b> 10
<b>ITERACION:</b> 1	<b>RESPONSABLE:</b> EDIOSN GARCÍA	<b>HORAS:</b> 11
<b>DESCRIPCION:</b>  Como Dr. del departamento médico quiero manejar las sintomatologías mediante el estándar CIE 10 y que sea de forma administrable		
<b>VALIDACION:</b>  Ingresar Enfermedades del CIE 10 de forma correcta		

<b>HISTORIA DE USUARIO</b>		
<b>NUMERO:</b> 7	<b>USUARIO:</b> Dr Departamento Médico	
<b>NOMBRE HISTORIA:</b> Modificación CIE 10		
<b>PRIORIDAD:</b> ALTA	<b>RIESGO:</b> MEDIO	<b>HORAS ESTIMADAS:</b> 10
<b>ITERACION:</b> 1	<b>RESPONSABLE:</b> EDIOSN GARCÍA	<b>HORAS:</b> 11
<b>DESCRIPCION:</b>  Como Dr del departamento médico quiero modificar las sintomatologías mediante el estándar CIE 10 y que sea de forma administrable		
<b>VALIDACION:</b>  Modificar Enfermedades del CIE 10 de forma correcta		

### ANEXO C: Pruebas Unitarias

<b>Prueba : 1</b>
<b>Descripción:</b> Prueba de Ingreso a Paciente
<b>Condiciones:</b>  Se realiza bajo el ambiente de desarrollo, se ingresarán datos correctos e incorrectos
<b>Resultado Esperado:</b>  Que ingrese en la base de datos y si existen datos mal ingresados emitan las alertas
<b>Resultado Obtenido:</b>  Una parte de los datos se perdio en el proceso de ingreso debido a la interfaz se debe refactorizar y en cuanto a las alertas funcionan bine

<b>Prueba : 2</b>
<b>Descripción:</b> Prueba de Ingreso a Paciente
<b>Condiciones:</b>  Ingresar los datos completos
<b>Resultado Esperado:</b>  Ingrese todos los datos completos del paciente en la base de datos
<b>Resultado Obtenido:</b>  Se realizaron satisfactoriamente

<b>Prueba : 3</b>
<b>Descripción:</b> Modificación de Paciente
<b>Condiciones:</b>  Se realiza bajo el ambiente de desarrollo, se ingresarán datos correctos e incorrectos
<b>Resultado Esperado:</b>  Modifiquen los datos de forma correcta
<b>Resultado Obtenido:</b>  Presento error en la llamada al servicio web necesita refactorización

<b>Prueba : 4</b>
<b>Descripción:</b> <b>Modificación de</b> Paciente
<b>Condiciones:</b>  <b>Se realiza</b> bajo el ambiente de desarrollo, se ingresarán datos correctos e incorrectos
<b>Resultado Esperado:</b>  <b>Modifiquen los datos</b> de forma correcta
<b>Resultado Obtenido:</b>  <b>Se modificó de</b> forma correcta

Prueba : <b>5</b>
<b>Descripción: Autenticación</b>
<b>Condiciones:</b>
<b>Se realiza</b> bajo el ambiente de desarrollo, se ingresarán datos correctos e incorrectos
Resultado <b>Esperado:</b>
<b>Mensajes de error</b> al iniciar sesión y que no deje ir a ninguna parte del aplicativo sin que este loggeado
Resultado <b>Obtenido:</b>
<b>Aprovada la prueba</b> satisfactoriamente solo refactorizar el mensaje de bienvenida

Prueba : <b>6</b>
<b>Descripción: Ingreso Historia Clínica</b>
<b>Condiciones:</b>
<b>Se realiza</b> bajo el ambiente de desarrollo, se ingresarán datos correctos e incorrectos
Resultado <b>Esperado:</b>
<b>Se ingrese satisfactoriamente</b> todos los controles de datos en la historia clínica ocupacional.
Resultado <b>Obtenido:</b>
<b>Se realizaron satisfactoriamente.</b>

Prueba : <b>7</b>
<b>Descripción: Modifiación Historia Clinica</b>
<b>Condiciones:</b>
<b>Se realiza</b> bajo el ambiete de desarrollo, se ingresarán datos correctos e incorrectos
<b>Resultado Esperado:</b>
<b>Se modifique satisfactoriamente</b> todos los controles de datos en la historia clínica ocupacional.
<b>Resultado Obtenido:</b>
<b>Se realizaron satisfactoriamente.</b>

Prueba : <b>8</b>
<b>Descripción: Ingreso CIE 10</b>
<b>Condiciones:</b>
<b>Se realiza</b> bajo el ambiete de desarrollo, se ingresarán datos correctos e incorrectos
<b>Resultado Esperado:</b>
<b>Ingrese las enfermedades</b> y sintomas en la base de datos
<b>Resultado Obtenido:</b>
<b>Se ingresaron satisfactoriamente</b>

Prueba : 9
<b>Descripción:</b> Modificación CIE 10
<b>Condiciones:</b>  Se realiza bajo el ambiente de desarrollo, se ingresarán datos correctos e incorrectos
<b>Resultado Esperado:</b>  Modifique las enfermedades y síntomas en la base de datos
<b>Resultado Obtenido:</b>  Se modificaron satisfactoriamente

<b>Prueba : 10</b>
<b>Descripción:</b> Reportes
<b>Condiciones:</b>  Se visualuce la historia clinica de un paciente
<b>Resultado Esperado:</b>  Se encuentren los datos que necesita el médico
<b>Resultado Obtenido:</b>  Se realizo satisfactoriamente