MODULO DE DIGITALIZACIÓN Y SISTEMATIZACIÓN DE LA ATENCIÓN INTEGRADA A LAS ENFERMEDADES PREVALENTES DE LA INFANCIA (AIEPI) PARA LA CLÍNICA DE LA COSTA

HAROLD RANGEL CHARRIS

CORPORACIÓN UNIVERSITARIA DE LA COSTA CUC

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS

BARRANQUILLA, COLOMBIA

2012

MODULO DE DIGITALIZACIÓN Y SISTEMATIZACIÓN DE LA ATENCIÓN INTEGRADA A LAS ENFERMEDADES PREVALENTES DE LA INFANCIA (AIEPI) PARA LA CLÍNICA DE LA COSTA

HAROLD RANGEL CHARRIS

Trabajo de tesis para optar por el título de Ingeniero de Sistemas

Asesor: Ing. Jorge Hernández Mercado

CORPORACIÓN UNIVERSITARIA DE LA COSTA CUC

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS

BARRANQUILLA, COLOMBIA

2012

NOTA DE ACEPTACION

_____

_____

_____

_____

_____

_____

_____

_____

JURADO 1

_____

_____

_____

_____

JURADO 2

Barranquilla 17 de Marzo de 2012

Señores: Corporación Universidad de la Costa (CUC)
Atención: Departamento Admisiones y Registros y Biblioteca
Asunto: Recepción de proyecto de grado

El día 17 de Marzo de 2012, el estudiante **Harold Rangel Charris** identificado con **C.C. 72357278**, hizo la entrega de los documentos y monografía de proyectos de grado junto con los anexos respectivos al programa de ingeniería de sistemas. A continuación se relaciona la información del proyecto de grado:

Titulo del proyecto: Modulo de Digitalizacion y Sistematizacion de la Atencion Integrada a las Enfermedades Prevalentes de la Infancia (AIEPI) Para la Clinica de la Costa
Calificación: 4.5
Decisión del jurado evaluador: Aprobado
Tutor: Jorge Hernandez Mercado
Cotutor: Roberto Morales Ortega

El jurado califico el proyecto presentado como un proyecto destacable dado los objetivos alcanzados con el desarrollo del trabajo de grado. Esta calificación incluye el desempeño durante todo el proyecto y la sustentación final asignada por los jurados que por su director de proyecto de grado lo que incluye:

Atentamente

Paola Ariza Colpas

Ing. Paola Patricia Ariza Colpas
Director de Programa
Ingeniería de Sistemas

A Dios,

a mi familia.

AGRADECIMIENTOS

A mi familia por su apoyo durante toda la carrera por creer en mí y el ayudarme a seguir luchando por lo que quería.

A mis amigos por enseñarme a luchar por las cosas.

A todos aquellos que me ayudaron durante todo este tiempo.

Gracias.

CONTENIDO.

pág.

INTRODUCCION. 12

1. MARCO CONTEXTUAL. 14

1.1 ANTECEDENTES. 14

1.2 PLANTEAMIENTO DEL PROBLEMA. 16

1.3 JUSTIFICACIÓN 17

1.4 OBJETIVOS DE LA INVESTIGACIÓN. 18

1.4.1 Objetivo General. 18

1.4.2 Objetivos Específicos. 18

1.5 ESTRUCTURA DE LA TESIS. 19

2. MARCO TEÓRICO. 20

2.1 INTRODUCCIÓN 20

2.2 ¿QUÉ ES LA AIEPI? 21

2.2.1 ¿Cómo Esta Conformado La AIEPI? 22

2.2.2 ¿Cómo Funciona El AIEPI? 23

2.3 PROGRAMACIÓN ORIENTADA A LA WEB. 24

2.4 INICIOS DE LA PROGRAMACIÓN WEB. 25

2.5 TECNOLOGÍAS Y ESTÁNDARES PARA LA PROGRAMACIÓN WEB. 26

2.6 LAS BASES DE LA PROGRAMACIÓN WEB. 26

2.6.1 El Modelo Cliente-Servidor. 26

2.6.2 Estructura, Presentación Y Contenidos. 28

2.7 TECNOLOGÍAS DE SCRIPTING / LENGUAJES DE PROGRAMACIÓN DE SCRIPTS. 46

2.7.1 Programación Del Lado Del Cliente. 46

2.7.2 Programación Del Lado Del Servidor. 53

2.8 APLICACIONES WEB ENRIQUECIDAS. 65

2.8.1 Ajax. 66
2.9 FRAMEWORK WEB. 70
2.9.1 Tipos de Frameworks Web. . 71
2.10 W3C. 72
2.11 EL PATRÓN MVC. 73
2.11.1 Las Capas De La Arquitectura MVC. 74
2.11.2 FrameworksQue Utilizan Este Modelo. 74
2.12 CONCLUSIÓN. 75
3. MODELO DE ANALISIS Y DISEÑO. 76
3.1 INTRODUCCIÓN. 76
3.2 PLANIFICACIÓN. 77
3.2.1 Etapas, Actividades Y Duración. 77
3.3 REQUERIMIENTOS. 78
3.3.1 Definición De Actores De Negocio. 79
3.3.2 Necesidades Genéricas Detectadas En Los Actores De Negocio. 79
3.3.3 Requerimientos Del Sistema. 79
3.4 ANÁLISIS Y DISEÑO. 81
3.4.1 Análisis Del Sistema. 82
3.4.2 Decisiones De Diseño Del Sistema. 96
3.5 CONCLUSIÓN. 98
4. IMPLEMENTACIÓN Y PUESTA EN MARCHA. 99
4.1 INTRODUCCIÓN. 99
4.2 IMPLEMENTACIÓN. 100
4.2.1 Lenguajes, Tecnologías Y Servicios De Base Datos Utilizados. 100
4.3 MODELO DE DESPLIEGUE. 108
4.4 INTERFAZ DE USUARIO GUI. 108
4.5 PUESTA EN MARCHA. 111
4.5.1 Tecnología Y Requerimientos Del Nodo Cliente. 111
4.5.2 Tecnología Y Características Del Nodo Servidor. 112
4.5.3 Fusión Del Modulo De Digitalización Y Sistematización De La Estrategia AIEPI Al S.I. SAIM. 112

4.5.4 Capacitación De Los Usuarios.....113
4.6 CONCLUSIÓN.....114
5. CONCLUSIÓN.....115
BIBLIOGRAFÍA.....116

LISTA DE TABLAS.

Tabla 1: Diferentes lenguajes XML según su orientación 35
Tabla 2: Cronograma de actividades a realizar en este proyecto.....77
Tabla 3: Requerimientos del sistema para el proyecto propuesto 79
Tabla 4: Descripción caso de uso creación de formato.....83
Tabla 5: Descripción caso de uso realización de la encuesta 86
Tabla 6: Descripción caso de uso reportes de la encuesta 87
Tabla 7: Descripción caso de uso reportes estadisticos.....89

LISTA DE GRAFICAS.

Ilustración 1: Ejemplo en formato plano de un área de la estrategia AIEPI 23
Ilustración 2: Condición de un diagnostico de la estrategia AIEPI.....23
Ilustración 3: Modelo Cliente – Servidor.....27
Ilustración 4: Vista de una página web en un explorador IExplorer 31
Ilustración 5: Pagina web con un estilo css aplicado.....44
Ilustración 6: Pagina web solo con la estructura HTML.....45
Ilustración 7: Pagina web con un estilo css aplicado.....45
Ilustración 8: Secuencia de la validación realizada con JavaScript.....50
Ilustración 9: Árbol de la estructura DOM de una página web.....52
Ilustración 10: Proceso realizado por el lenguaje JSP para mostrar una página web 57
Ilustración 11: Estructura del modelo vista controlador MVC propuesto por JSP ..59
Ilustración 12: Tecnologías del AJAX 66
Ilustración 13: Comparación del proceso normal de una aplicación web con el proceso realizado por AJAX 67
Ilustración 14: Proceso síncrono y asíncrono en una aplicación web.....68
Ilustración 15: Estructura del patrón Modelo Vista Controlador.....74
Ilustración 16: Diagrama de caso de usos del S:I desarrollado en este proyecto..91
Ilustración 17: Diagrama relacional.....92
Ilustración 18: Diagrama de secuencia creacion de un area 93
Ilustración 19: Diagrama de secuencia ralizacion de la encuesta 95
Ilustración 20: Diagrama de secuencia reportes de encuestas 95
Ilustración 21: Interfaz encuesta estrategia AIEPI.....109
Ilustración 22: Interfaz encuesta estrategia AIEPI, desplegando area 110
Ilustración 23: Interfaz creación formato AIEPI, creación de preguntas 110

Ilustración 24: Interfaz creación formato AIEPI, creación de preguntas: desplegando preguntas realizadas 111

LISTA DE ANEXOS.

Anexo : Carta de entrega y documentacion de los autores para la consulta, la reproducción parcial o total, y publicación electrónica del texto completo de tesis y trabajos de grado.....117
Anexo 2: Formulario de la descripción de la tesis o del trabajo de grado.....118
Anexo 3: Pagina web oficial de las diferentes tecnologías y estándares utilizados en la programación web.....120

RESUMEN

Desde 1996 la Organización Mundial de la Salud (OMS) y el Fondo de las Naciones Unidas para la Infancia (UNICEF) presento una estrategia para mejorar la salud en la niñez, *la Atención Integrada a las Enfermedades Prevalentes de la Infancia (AIEPI)*, la cual enfoca la atención de los menores de cinco años en su estado de salud más que en las enfermedades que ocasionalmente pueden afectarlos. De este modo, reduce las oportunidades perdidas de detección precoz y tratamiento de enfermedades que pueden pasar inadvertidas para los padres y para el personal de salud, con el consiguiente riesgo de agravarse y de presentar complicaciones, pero en nuestro país está esta estrategia no se encuentra completamente implementado.

Dado esta necesidad de seguir implementando y reconociendo la importancia que la AIEPI para la sociedad, la motivación de este proyecto es el poder **Digitalizar y Sistematizar de la Atención Integrada a las Enfermedades Prevalentes de la Infancia (AIEPI)**, mostrando completamente una **AIEPI** de forma digital, con el cual se buscaría una forma más cómoda y rápida de poder desarrollarla.

Con la Digitalización y Sistematización no solo se agiliza el proceso de realización de estrategia, sino que también se bajan los riesgos de errores en la realización de la misma si esta fuese llenada de forma manual, -el cual es error más concurrente que se presenta al momento de poder llenar un formulario- así como de:

- Contar con históricos de todas las encuestas realizadas y tener acceso rápido a los mismos.

- Al ser automatizada proporciona preguntas que varían de acuerdo a la edad del paciente y las respuestas suministradas en otras preguntas. Lo que disminuye el error de equivocación por parte del realizador
- Mejorar la organización y el funcionamiento de los servicios de salud para así poder brindar una mejor y más rápida atención a los infantes.
- Mejorar el desempeño del personal de salud para la atención a los menores y sus familiares.
- Permitir a los médicos ir enfocados a una impresión diagnostica del paciente.
- Llevar un histórico de los AIEPI realizados a los pacientes.

INTRODUCCION.

El uso de la informática ha evolucionado completamente a los sistemas de información, inicialmente estos eran considerados como un instrumentos amplificador de las distintas actividades de una organización, posteriormente el desarrollo de la informática y las telecomunicaciones permitieron incrementar la eficacia en la realización de las tareas, ahorrar tiempo en el desarrollo de las actividades y almacenar la mayor cantidad de información en el menor espacio posible, lo cual aumentó en las organizaciones el interés en los sistemas de información.

Dado a esta evolución de la informática y los sistemas de información, en la década de los 60 esta se fue introduciendo en la medicina, para proporcionar asesoramiento en la toma de decisiones dentro de la práctica de la medicina clínica y la cual ha extendido su uso en la actualidad, encontrando actualmente diferentes organizaciones de la salud que hacen uso de los sistema de información hospitalaria "HIS" (por sus siglas en ingles) para la administración de la información de los procesos clínicos dados en estas.

Esta tesis se enfoca en el desarrollo del un nuevo modulo para el HIS SAIM desarrollado por la clínica de la costa, y el cual pretende ser una solución para el uso simple y correcto de la estrategia AIEPI desarrollado por la PAHO y que en los últimos años se ha venido exigiendo en el país, pero que no ha podido ser implementada completamente por su gran complejidad de su uso.

A lo largo del desarrollo de la tesis se identificarán que es y en qué consiste esta estrategia y cuáles son los problemas que se presentan en la Clínica de la Costa que hace que no se pueda poner en práctica, se expondrá la mejor solución a los problemas identificados. Otro de los aspectos que se desarrollará a lo largo de la

tesis es el de identificar como está desarrollado el S.I SAIM para poder hacer uso de esas tecnologías y técnicas usadas, por último se expondrá la documentación que conlleva a la realización de un software, documentación regida por la ingeniería de software.

1. MARCO CONTEXTUAL.

1.1 ANTECEDENTES.

Desde la aparición de la máquina de Hollerith usada para tabular el censo de los Estado Unidos y cuya maquina demostró que realizando un proceso automatizado en la lectura de la encuesta se podía reducir significativamente el tiempo que se necesitaba para procesar la información de estas.

Luego con el avance de la tecnología y de las ciencias computacionales se ha demostrado que no solo se puede hacer uso de la tecnología para el procesamiento de los datos en una encuesta sino que realizando estas de una forma digital la reducción del tiempo iba a ser mayor tanto para el usuario quien realiza la encuesta como para la organización que la dirige.

En la actualidad existen compañías encargadas de proveer a organizaciones software para la realización de encuestas digitales permitiendo desde la configuración de la encuesta hasta los reportes estadísticos de los datos obtenidos de esta, entre esas compañías se encuentran:

- NetQuest: con el software Survey Manager con el que permiten cubrir todas las fases del proceso de creación, publicación, envío y análisis de encuestas online.
- TricTrac: con el software del mismo nombre, que presenta las mismas características de Survey Manager.

Otro de las organización que presentan desde la creación hasta el análisis de encuestas son las plataformas Moodle y BlackBoard, las cuales son plataformas

académicas y que permiten la creación de estos formatos para exámenes académicos.

Adicionalmente a las fases de proceso de estas aplicaciones como son: creación, publicación, análisis, el *Modulo de digitalización y sistematización de la estrategia a las enfermedades prevalentes de la infancia AIEPI* contendrá un motor de inferencia que analizará las respuestas obtenidas, mostrándole al usuario un diagnostico dependiendo de las respuesta dadas.

1.2 PLANTEAMIENTO DEL PROBLEMA.

Dos problemas se unen para plantear la necesidad de este proyecto, el primero es el poder reforzar, afianzar y poner en marcha la estrategia AIEPI desarrollada desde 1996 y teniendo varias actualizaciones siendo la última en el 2010 y que hasta la fecha no ha sido completamente desarrollada y aplicada en nuestro país.

Uno de los más grandes problemas y que se tienen en casi todo tipo de situación y aspectos, son los errores que se cometen al momento de realizar expedientes de forma manual, estos errores generalmente parecieran que no fuesen muy grandes pero que al momento de recolectar toda esa información y generar informes se convierten en un gran problema.

Para la **Clínica de la Costa**, espacio de estudio de la investigación estas situaciones se presentan, para el primero, aunque tienen desarrollado la estrategia AIEPI de forma manual, no se está poniendo en práctica debido a su complejidad y la demora que trae esta al momento de realizarla; por otra parte es comúnmente ver errores escritura, de selección de las respuestas, entre otros; los cuales se ven reflejados al momento de recopilar la información.

Entonces, ***¿De qué forma se puede fortalecer y poner en marcha la estrategia AIEPI en la clínica y que está a la vez disminuya los errores generados al momento de realizarla?***

1.3 JUSTIFICACIÓN.

El procesamiento automatizado de datos, desde sus comienzos vistos con la máquina de Hollerith utilizada en el censo de 1890 de los Estados Unidos, ha demostrado que es la mejor forma de poder analizar, revisar y arrojar resultados de diferentes medios de obtención de datos.

En estos momentos con los avances tecnológicos y el uso cada vez más de los computadores, la obtención de datos ha pasado no solo de ser realizados en documentos planos a documentos digitales, permitiendo así que el procesamiento automatizado de datos sea cada vez más necesario.

Trayendo múltiples ventajas tales como: agilización en el proceso de la realización de encuestas, disminución de los riesgos de errores en la realización de estas, confiabilidad de los datos obtenidos, facilidad de adquirir históricos de los datos, entre otras.

Dado todas estas ventajas que traen la digitalización y el procesamiento automatizados de datos, se escoge como solución a la pregunta problema de este trabajo la digitalización y sistematización de la estrategia AIEPI permitiendo generar una estrategia de manera sencilla, rápida, enfocada a disminuir los errores que se presentan en la clínica y sobre todo el poder obtener de una manera instantánea los resultados que arrojan esta estrategia al ser realizada.

1.4 OBJETIVOS DE LA INVESTIGACIÓN.

1.4.1 Objetivo General.

- Digitalizar y sistematizar la estrategia de la atención integrada a las enfermedades prevalentes de la infancia (AIEPI).

1.4.2 Objetivos Específicos.

- Analizar y entender cómo funciona la estrategia AIEPI para poder determinar cuál es la mejor forma de sistematización a aplicar.
- Estudiar y comprender tecnologías y estándares que forman la base para el desarrollo de proyectos Web.
- Analizar detalladamente las necesidades de los actores de negocio involucrados para poder definir todos los requerimientos necesarios que ayuden en la construcción de SI.
- Modelar dicho sitio y sus interfaces, en base a la aplicación de los patrones de arquitectura de software MVC, considerando las tecnologías y estándares estudiados.
- Desarrollar los diferentes módulos que comprenden el sistema (encuesta, reporte)
- Fusionar el nuevo sistema con el sistema existente en la clínica
- Implementación y puesta en marcha del sistema digital del AIEPI
- Desarrollo del manual del usuario y la documentación del sistema
- Capacitar al personal de la clínica para el uso del sistema

1.5 ESTRUCTURA DE LA TESIS.

Esta tesis está dividida en 5 capítulos. El capitulo 2 se centra en identificar y entender todos aquellos temas que servirán de base de conocimiento para los capítulos posteriores. En el capítulo 3 se basa en la generación de un modelo de análisis y diseño, en el cual se desarrollara las bases para la implementación y concreción del S.I.

Posteriormente, en el capítulo 4 se concreta el modelo de análisis y diseño realizado, dando como resultado el desarrollo del S.I. de la Digitalización y Sistematización de la estrategia AIEPI. Finalmente, el trabajo de la tesis culmina con la exposición de las conclusiones en el capítulo 5.

2. MARCO TEÓRICO.

2.1 INTRODUCCIÓN.

Una vez identificado y estudiado el problema de estudio, se empezara a identificar como está conformado y cómo funciona la AIEPI, así como de buscar cuales son las mejores técnicas con que se puede dar solución a dicho problema, dando cumplimiento al objetivo principal de este proyecto; a la vez este marco teórico se desarrolla para que puedan conseguir más adelante en la realización de nuevos proyectos una fuente de información confiable.

En el desarrollo de esta etapa se buscaran cumplir los objetivos:

- Analizar y entender cómo funciona la estrategia AIEPI para poder determinar cuál es la mejor forma de sistematización a aplicar.
- Estudiar y comprender tecnologías y estándares que forman la base para el desarrollo de proyectos Web.

El enfoque dado a este marco teórico es la programación orientada a la web, con la que se definirán todas las tecnologías, estándares y tendencia que se refleja a esta programación.

Con lo cual se definirán las bases para poder desarrollar un modelo de analisis y diseño, etapa siguiente en el desarrollo de este proyecto.

2.2 ¿QUÉ ES LA AIEPI?

La Atención Integrada a las Enfermedades Prevalentes de la Infancia (AIEPI) es una estrategia elaborada por la Organización Mundial de la Salud (OMS) y el Fondo de las Naciones Unidas para la Infancia (UNICEF), que fue presentada en 1996 como la principal estrategia para mejorar la salud en la niñez. Enfoca la atención de los menores de cinco años en su estado de salud más que en las enfermedades que ocasionalmente pueden afectarlos. De este modo, reduce las oportunidades perdidas de detección precoz y tratamiento de enfermedades que pueden pasar inadvertidas para los padres y para el personal de salud, con el consiguiente riesgo de agravarse y de presentar complicaciones. La AIEPI incorpora, además, un fuerte contenido preventivo y de promoción de la salud como parte de la atención. Así contribuye, entre otros beneficios, a aumentar la cobertura de vacunación y a mejorar el conocimiento y las prácticas de cuidado y atención de los menores de cinco años en el hogar, de forma de contribuir a un crecimiento y desarrollo saludables.

La implementación de la AIEPI contempla la participación tanto de los servicios de salud como de la comunidad y la familia, y se lleva a cabo por medio de tres componentes. El primero está dirigido a mejorar el desempeño del personal de salud para la prevención de enfermedades en la niñez y para su tratamiento. El segundo, se dirige a mejorar la organización y funcionamiento de los servicios de salud para que brinden atención de calidad apropiada. El tercer componente está dirigido a mejorar las prácticas familiares y comunitarias de cuidado y atención de la niñez.

La AIEPI es una estrategia que integra todas las medidas disponibles para la prevención de enfermedades y problemas de salud durante la niñez, para su

detección precoz y tratamiento efectivo, y para la promoción de hábitos de vida saludables en la familia y la comunidad.

AIEPI puede ser aplicada tanto por el personal de salud como por otras personas que tienen a su cargo el cuidado y atención de niños y niñas menores de cinco años, entre ellos, los padres.

AIEPI brinda los conocimientos y habilidades para evaluar en forma secuencial e integrada la condición de salud del niño y, de esta manera, detectar las enfermedades o problemas que más frecuentemente la afectan, según el perfil epidemiológico de cada lugar. A partir de esta evaluación, la AIEPI brinda instrucciones claras sobre la clasificación de las enfermedades y problemas, y establece el tratamiento que debe administrarse para cada una de ellas. La estrategia también contiene las indicaciones para controlar la evolución del tratamiento, para identificar la necesidad de aplicar medidas de prevención y aplicarlas, y para informar y educar a los padres sobre la prevención y promoción de la salud infantil.

Sobre esta base, la AIEPI es considerada en la actualidad como la estrategia más eficiente para reducir la carga de enfermedad y discapacidad en la población, y contribuir a un crecimiento y desarrollo saludables durante los primeros cinco años de vida.

2.2.1 ¿Cómo Esta Conformado La AIEPI?

La AIEPI, son tres formatos de encuestas dividido según un rango de edades, AIEPI para Recien nacidos, AIEPI para niños de 0 a 2 meses y AIEPI para niños de 2 meses a 5 años, en cada uno de esos formatos existen unas areas con un encabezado, cada area contiene un grupo de preguntas y unos diagnosticos. Los

diagnosticos son seleccionados dependiendo de las respuestas a las preguntas del area.

Ilustración 1: Ejemplo en formato plano de un área de la estrategia AIEPI

2.2.2 ¿Cómo Funciona El AIEPI?

Como se mencionó anteriormente, la AIEPI está conformado por unas áreas que contienen preguntas y diagnósticos, para poder indicar el diagnostico apropiado se hace necesario que las respuestas dadas coincidan con la condición dada para cada diagnósticos, las condiciones de los diagnósticos son estudios que han realizado la PAHO y la UNICEF para estandarizar los problemas que se presentan en los menores de 5 años, por ejemplo: a continuación se muestra un cuadro de condiciones para el diagnóstico de ***NEUMONIA GRAVE***.

Uno de los siguientes signos: • Cualquier signo general de peligro • Tiraje subcostal • Saturación de O2 < 92% (o <90% si vive en una altura superior a 2500 msnm)	NEUMONÍA GRAVE

Ilustración 2: Condición de un diagnostico de la estrategia AIEPI

La PAHO y La UNICEF no solo han desarrollado las condiciones de cada diagnósticos, sino, que ha cada diagnostico lo han clasificado según una semaforización (rojo, amarillo, verde) ubicando a cada diagnostico en la semaforización según sea su nivel de gravedad.

Ya habiendo entendido como está conformado y como es el funcionamiento de la AIEPI, el enfoque siguiente del marco teórico es entender todo lo relacionado a la programación web, el cual es la otra parte del objetivo general de este proyecto, en esta parte se pretende analizar, estructuras, lenguajes y técnicas de la programación web, así como el de enseñar un poco de la historia de cada una ellas.

2.3 PROGRAMACIÓN ORIENTADA A LA WEB.

Desde la aparición de la Web en el internet, esta ha venido influenciando enormemente al mundo informático, como a la sociedad en general tanto es así que en poco menos de 20 años la Web ha roto las barreras físicas, económicas y lógicas y ha abierto todo un abanico de nuevas posibilidades.

Una de las áreas que más expansión la Web son las aplicaciones web, permitiendo la generación automática de contenido, la creación de páginas personalizadas según el perfil del usuario o el desarrollo del comercio electrónico, por otra parte una aplicación web permite interactuar con los sistemas informáticos de gestión de una empresa, como puede ser gestión de clientes, contabilidad o inventario, a través de una página web.

Con esas aplicación web han surgido diferentes lenguajes de programación así como de tecnologías y estándares relacionados con el desarrollo de estas aplicaciones, dándose lugar así a lo que es conocido como programación orientada a la web o programación web.

2.4 INICIOS DE LA PROGRAMACIÓN WEB.

El inicio de la programación web, se puede remontar al desarrollo de Tim Berners-Leede las estructuras fundamentales de la web, el HTML.
(**H**yper**T**ext **M**arkup **L**anguage), el protocolo HTTP (**H**yper**T**ext **T**ransfer **P**rotocol) y el sistema de localización de objetos en la web URL (**U**niform **R**esource **L**ocator), dada la necesidad de poder distribuir e intercambiar información acerca de sus investigaciones de una manera más efectiva y que hasta la fecha de hoy siguen siendo las estructuras fundamentales de la web.

Con el inicio de estas estructuras el HTML se convierte en el primer lenguaje de programación web, sin embargo este es un lenguaje de programación web estático, sin embrago, a medida que pasó el tiempo, nuevas tecnologías fueron desarrollándose y a la vez, surgieron nuevos problemas que habían que darles solución, dando lugar al desarrollo de lenguajes de programación para la web dinámicos, que permitieran interactuar con los usuarios y utilizaran sistemas de Bases de Datos.

2.5 TECNOLOGÍAS Y ESTÁNDARES PARA LA PROGRAMACIÓN WEB.

Con la aparición de la estructuras fundamentales de la web, y el inicio de la programación orientada a la web, fueron dándose solución a necesidades puntuales, pero a medida que fue evolucionando el internet y se fueron desarrollando nuevas tecnologías, aparecieron nuevas necesidades que habían que darles solución, es aquí cuando entonces se empiezan a crear nuevos lenguajes, técnicas y estándares para poder darles solución a todas esas necesidades.

Dada la necesidad del lector de poder conocer y de entrarlos a las técnicas y estándares que se han desarrollado, a continuación se explicaran las técnicas y estándares empezando con una perspectiva histórica y observando los avances tecnológicos que estas han tenido hasta llegar al alcance actual que tienen.

2.6 LAS BASES DE LA PROGRAMACIÓN WEB.

2.6.1 El Modelo Cliente-Servidor.

El modelo Cliente servidor es una aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes, en él, el cliente envía un mensaje solicitando un determinado servicio a un servidor, y este envía uno o varios mensajes con la respuesta.

Esta modelo permite distribuir físicamente los procesos y los datos en forma más eficiente lo que en computación distribuida afecta directamente el tráfico de la red, reduciéndolo grandemente.

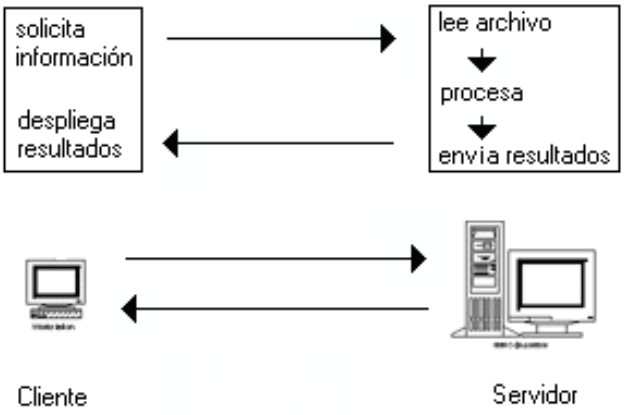


Ilustración 3: Modelo Cliente – Servidor

2.6.1.1 Cliente.

El cliente es el proceso que permite al usuario formular los requerimientos y pasarlos al servidor, se le conoce con el término front-end, este normalmente maneja todas las funciones relacionadas con la manipulación y despliegue de datos, por lo que están desarrollados sobre plataformas que permiten construir interfaces gráficas de usuario, además de acceder a los servicios distribuidos en cualquier parte de una red.

Las funciones que lleva a cabo el proceso cliente se resumen en los siguientes puntos:

- Administrar la interfaz de usuario.
- Interactuar con el usuario.
- Procesar la lógica de la aplicación y hacer validaciones locales.
- Generar requerimientos de bases de datos.

- RecibirresultadosdelServidor.
- Formatearresultados.

2.6.1.2 Servidor.

Es el proceso encargado de atender a múltiples clientes que hacen peticiones de algún recurso administrado por él. Al proceso servidor se le conoce con el término back-en, normalmente maneja todas las funciones relacionadas con la mayoría de las reglas del negocio y los recursos de datos.

Las funciones que lleva a cabo el proceso servidor se resumen en los siguientes puntos:

- Aceptar los requerimientos de bases de datos que hacen los clientes.
- Procesar requerimientos de bases de datos.
- Formatear datos para trasmitirlos a los clientes.
- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos

2.6.2 Estructura, Presentación Y Contenidos.

2.6.2.1 HTML y su evolución.

HTML es considerado el primer lenguaje de programación web, su aparición radica en 1991 con las estructuras fundamentales de la web, desarrollados por Tim Berners-lee, este lenguaje como su nombre lo indica (lenguaje de marcado de

hipertexto) consta de una series de etiquetas - por ejemplo: Esto es un párrafo - que se usan para describir la estructura y el contenido forma de texto, así como para complementar el texto con objetos tales como imágenes.

Las etiquetas o como su nombre en inglés Markup del lenguaje HTML son fragmentos de texto destacado de una forma especial que permiten la definición de las distintas instrucciones del lenguaje, tanto los efectos a aplicar sobre el texto como las distintas estructuras. Estas etiquetas son la base principal del lenguaje HTML.

A continuación se mostrara un ejemplo de una estructura de una página Web utilizando diferentes etiquetas, y como se ve esta estructura en un navegador.
Ese ejemplo se irá evolucionando a medida que se pase por el resto del contenido del marco teórico, no se especificara ni se detallara cada etiqueta siendo que ese no es el objeto de estudio de este proyecto, sin embargo, se les brindara las direcciones de las páginas web oficiales en donde puedan encontrar toda la información relacionada con el tema (Anexo I).

El código siguiente corresponde a la página de inicio de una red social, el cual fue un proyecto realizado por el mismo autor durante su semestre de ingeniería de software.

```
<! DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
    <Html>
        <Head>
        <metahttp-equiv="Content-Type"content="text/html; charset=utf-8" /><title>My
        Friend's</title>
    </head>
    <Body>
        <div id="contenedor">
            <div id="cabecera">
                <div id="header">
                    <div id="logo"></div>
                    <ulid="menu">
                        <li id="registro"><a href="#">Registrate</a></li>
```

```
                    <li id="login"><a href="#">Ingresar</a></li>
                </ul>
                <div id="formLogin">
                    <form method="post"name="frm">
                        <div id="divLogin">
                            <input                    type="text"id="loginEmail"name="inputEmail"
                        value="email"/>
                            <input type="password"id="loginPass"name="inputPass"
                        Value="password"/>
                            <input type="submit"id="login Submit"value="Ingresar"/>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    <div id="contenido"class="clear">
        <div id="principal"></div>
        <div id="secundario">
            <div id="form">
                    <div id="header2">
                        <p id="txtheader2">Inscribete ahora</p>
                    </div>
                    <div id="contform">
                       <form method="post"name="form"/>
                        <fieldset>
                           <label class="txtformulario"for="nombre">Nombre :<
                        /label>
                           <input type="text"class="inputForm"name="nombre"/>
                        </fieldset>
                        <fieldset>
                           <label
                        class="txtformulario"for="apellido">Apellido:</label
                        >
                           <input
                        type="text"class="inputForm"name="apellidos"/>
                        </fieldset>
                        <fieldset>
                           <label
                        class="txtformulario"for="correo">email:</label>
                           <input
                        type="text"class="inputForm"name="email"autocomplete
                        =OFF/>
                           <span class="hint"></span>
                        </fieldset>
                        <fieldset>
                           <label
                        class="txtformulario"for="pass">Password:</label>
                           <input
                        type="password"class="inputForm"name="pass"id="input
                        Pass"maxlength="8"/>
                           <span class="hint"></span>
                           <span  id="pass">  *  minímo  4  caracteresmaximo  8
                        </span>
                        </fieldset>
                        <div id="submit">
```

```
                        <input
                       id="submittForm"type="submit"align="center"value="Re
                       gistrarse"name="enviar" />
                       </div>
                    </form>
                  </div>
               </div>
            </div>
            <div class="clear"></div>
         </div><div id="pie">
         <div id="leyenda">
            <p  id="txtLeyenda">Al  hacer  clic  en  Registrarse,  estarás
      indicando que has leído, entendido y accedes a las
               <aid="linkPoliticas"href="#">Condiciones de uso y Política de
            privacidad</a> de
               <a id="homepage"href="#">myFriend's</a>
            </p>
         </div>
      </div>
    </div>
   </body>
</html>
```

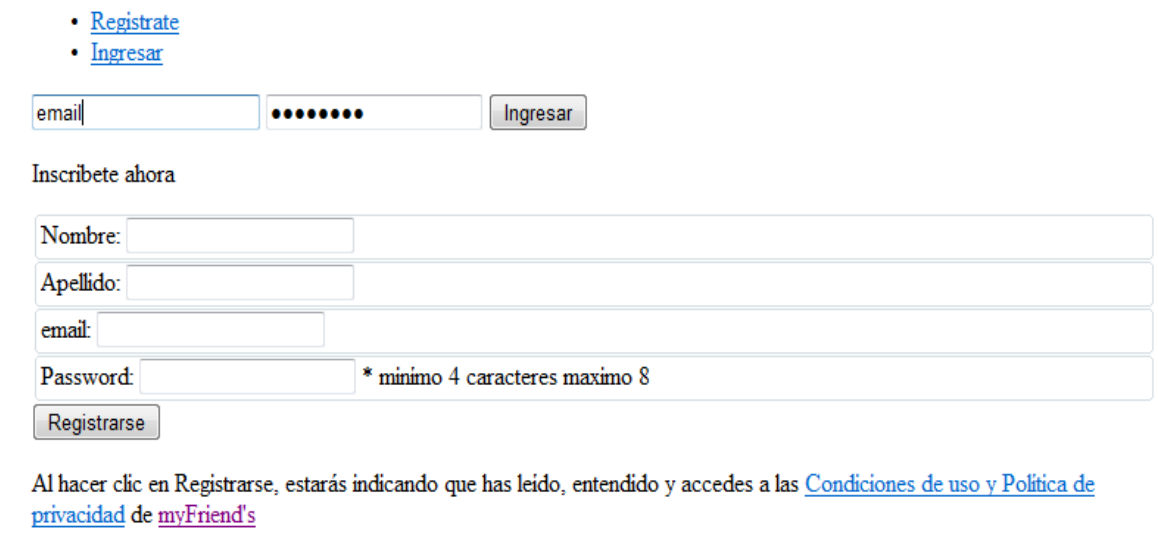


Ilustración 4: Vista de unapágina web en un explorador IExplorer

LA EVOLUCIÓN DEL HTML.

HTML ha evolucionado desde su aparición hasta la fecha incluyendo nuevas estructuras y estilos, así como pasar de una estructura completamente de texto hasta llegar a poder a tener un espacio de dibujo y reproducción de video.

- **HTML:** el inicio de este lenguaje, como se mencionó anteriormente, radica en 1991 cuando el físico Tim Berners-Lee, con la necesidad de poder distribuir e intercambiar información acerca de sus investigaciones desarrollo las estructuras de la web HTML, HTTP y URL. Esta primera versión del lenguaje solo mostraba texto con formato, contenía etiquetas para títulos, párrafos, listas y viñetas, entre otras características, y se publicó bajo el nombre de HTML Tags.

- **HTML 2.0:** con la necesidad de volver el lenguaje HTML en un estándar en 1993 el Grupo Especial sobre Ingeniería de Internet (IETF por su sigla en inglés), empieza a estandarizar el lenguaje creando el HTML+, pero no es hasta 1995 que la misma entidad, crea un grupo de trabajo para estandarizar el HTML dando como resultado el HTML 2.0, éste lograba su compatibilidad con los navegadores y adicionaba las etiquetas para lograr una presentación de imágenes, tablas y formularios

- **HTML 3.2:** en 1996 la W3C (*World Wide Web Consortium)*dirigido por Tim Berners-Lee desarrolla otro estándar conocido como el HTML 3.2, en esta publicación del lenguaje se incorporan los *applets* de Java y texto que fluye alrededor de las imágenes.

- **HTML 4.0:** en 1998 dada a una revisión a la versión publicada en 1997, se publica el HTML 4.0 incorporando nuevas estructuras y tecnologías con las hojas de estilo CSS, pequeños programas o scriptsen las páginas web, mejora en la accesibilidad de las páginas diseñadas, tablas complejas y en los formularios.

- **HTML 4.01:** en 1999 nace el 4.01 siendo una revisión y actualización del estándar 4.0, esta versión no incluye novedades significativas y fue el último estándar desarrollado por la W3C, dado a que sus objetivos se enfocaron en el estudio y desarrollo del XHTML.
- **HTML 5.0:** en el 2004 la WHATWG (*Web HypertextApplicationTechnologyWorkingGroup*), grupo conformado por Apple, Mozilla y Operaexpresa en un comunicado seguir estudiando y el poder desarrollar un nuevo estándar del HTML el cual había sido dejado a un lado por la W3C; en 2006 la W3C queriendo continuar con la estandarización del lenguaje pide ser parte del grupo de trabajo a estudiar el nuevo estándar, pero no es hasta el 2008 que el WHATWG y la W3C lanza un borrador de lo que sería el HTML 5.0, el lanzamiento oficial del HTML 5.0 será en el 2012, trayendo novedades como canvas, reproducción de audio y videos, así como de también ampliación en las propiedades de las etiquetas como de nuevas etiquetas incluidas a este nuevo estándar.

2.6.2.2 XML

Debido al crecimiento descontrolado que tuvo el HTML y dado al no cumplimiento de los problemas que planteaba la sociedad global de internet, en 1998 la W3C

empezó en el desarrollo de XML (Extended MarkupLanguage), pretendiendo solucionar las carencias del HTML en lo que se respecta al tratamiento de la información.

El XML es un lenguaje en base a tags que dando soporte a una gran variedad de aplicaciones y sin tener algún tipo de tags predefinidos, sino que se basa en las definiciones del programador, por lo que es considerado un “meta lenguaje”, este ha sido ampliamente usado en los últimos años, siendo uno de los pilares de las tecnologías de Internet.

Aunque fue lanzado por la W3C en 1998, el XML es un subgrupo simplificado del SGML (Standard GeneralizedMarkupLanguage) -lanzado por la IBM y estandarizado por la ISO- el cual es ampliamente utilizado para compartir e integrar información entre aplicaciones.

Los lenguajes basados en XML son a su vez razonablemente legibles y entendibles para el humano, ya que los tags para cada caso son en general elegidos de tal forma que mantengan una relación con el contenido que encierran dentro de un documento.

A continuación se mostrara un ejemplo de una estructura realizada en XML, en este ejemplo consiste enun menú donde se puede organizar el nombre, precio, descripción y las calorías de cada comida.

```
<menu_almuerzo>

<comida>
    <nombre>Waffles</nombre>
    <precio>$2.00</precio>
    <descripcion>Waffles baratos de McDonalds</descripcion>
    <calorias>650</calorias>
</comida>
```

```
<comida>
    <nombre>Hamburguesa</nombre>
    <precio>$5.00</precio>
    <descripcion>La hamburguesa más común de McDonald's</descripcion>
<calorias>1500</calorias>

</comida>
</menu_almuerzo>
```

Desde que fue lanzado, se han desarrollado muchos lenguajes en base a este, clasificándolos en orientados a aplicaciones y orientados a tecnología.

Tabla 1: Diferentes lenguajes XML según su orientación

Orientación	Lenguajes
Intercambio y extracción de información	SOAP, WSDL, XQuery, XPath, SAX, DOM
Formación de "vocabularios" específicos de negocio	MathML, MusicML, OTA, HL7, XBRL
Presentación de la información	XHTML, XForms, WML, SVG
Tratamiento y transformación del propio XML	XSLT, XSL-FO, XML-Schema, RelaxNG, XLink, XPointer 9

Si desean encontrar toda la información relacionada con el tema se pueden referir a la sección de anexos (Anexo I).

2.6.2.3 XHTML.

Dada la necesidad de poder crear un lenguaje de marcado más robusto y ya que empezaban a proliferar otros dispositivos, además de los ordenadores, capaces de reproducir la información de la web, XHTML Extensible HypertextMarkupLanguage (lenguaje extensible de marcado de hipertexto), este no es más que la fusión del XML y el HTML.

El XHTML tiene básicamente las mismas funcionalidades del HTML, pero cumple las especificaciones, más estrictas, del XML. Su objetivo es avanzar en el proyecto del World Wide Web Consortium de lograr una web semántica, donde la información, y la forma de presentarla estén claramente separadas. En este sentido, XHTML serviría únicamente para transmitir la información que contiene un documento, dejando para las hojas de estilo y JavaScript su aspecto y diseño en distintos medios.

DIFERENCIAS ENTRE HTML Y XHTML.

Se eliminan elementos no semánticos, como se ha mencionado anteriormente, uno de los objetivos del XHTML es poder separar la información del documento del aspecto y diseño, por tanto:

- Desaparecen las etiquetas de HTML <font>, .
- Desaparecen varios atributos de formato.
- El aspecto del documento se describe únicamente a través de hojas de estilo.

Al Ser XML, Se Exige:

- Incluir siempre la etiqueta "doctype" apropiada.

- Todas las etiquetas deben cerrarse, aunque sea poniendo una barra "/", como, por ejemplo: <br> pasa a ser <br />.
- Sólo pueden incluirse datos en formatos admitidos por XML. Esto da problemas para incluir JavaScript en los documentos directamente.
- Todos los atributos deben tener un valor, y meterlo entre comillas.

La primera versión de XHTML se denomina XHTML 1.0 y se publicó el 26 de Enero de 2000 (y posteriormente se revisó el 1 de Agosto de 2002).

La versión XHTML 1.1 ya ha sido publicada en forma de borrador. También ha sido publicado el borrador de XHTML 2.0, que supondrá un cambio muy importante respecto de las anteriores versiones de XHTML.

Si desean encontrar toda la información relacionada con el tema se pueden referir a la sección de anexos (Anexo I).

2.6.2.4 CSS.

CSS *(*Cascading Style Sheets*)*es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML, fue desarrollado por HåkonWium Lie y BertBos el cual fue luego estandarizado por la W3C en 1996 conociendo la primera versión como CSS nivel 1.

CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "documentos semánticos"). Además, mejora la accesibilidad del documento, reduce la

complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.
Continuando con el ejemplo de la página de inicio de la red social, a continuación se le aplicara diferentes propiedades CSS que le dará un diseño completamente diferente a que se obtuvo solo con la estructura realizada en HTML, como se ha mencionada anteriormente no se profundizara sobre las propiedades aplicadas, si desean encontrar toda la información relacionada con el tema se pueden referir a la sección de anexos (Anexo I).

Propiedades CSS aplicadas a la página de inicio de la red social.

```
UL
{
        list-style-type: none;
}
#contenedor
{
        width: 900px;
        margin: 15px auto 0;
}
#cabecera
{
        width: 900px;
        position: relative;
        height: 60px;
        margin: 0;
        padding: 0;
        color: #FFF;
}
#contenido
```

```
{
	width: 890px;
	color: #FFF;
}
#contenido #principal
{
	float: left;
	width: 560px;
	margin-top: 0px;
	padding: 0 0 0 20px;
	background: #FFF;
	height: 430px;
	background: url(image.jpg) no-repeat;
}
#contenido #secundario
{
	float: left;
	width: 280px;
	height: 393px;
	margin: 15px 0 0;
	padding-left: 10px;
	padding-right: 10px;
	padding-top: 9px;
}
#pie
{
	clear: both;
	height: 60px;
	margin-bottom: 50px;
	color: #6685CC;
	position: relative;
	padding-left: 48px;
	margin-top: 15px;
}
.clear
{
	clear: both;
}
/*----cabecera----------------------------------------------------------------*/
#header
{
	background: url(img_header_cont.png) no-repeat;
	height: 60px;
}
#logo
{
	float: left;
	height: 39px;
	width: 146px;
	margin-right: 60px;
	background: url(logo.gif) no-repeat;
	padding-left: 0px;
	margin-left: 9px;
	margin-top: 1px;
```

```
        background-position-y: -1px;
        /*[empty]background-position-x:;*/
}
#menu
{
        width: 180px;
        list-style-type: none;
        margin: 0;
        padding: 0;
        float: left;
}
#registro, #login
{
        float: left;
}
#registro
{
        width: 90px;
}
#login
{
        width: 80px;
}
#registro A, #login A
{
        padding-top: 10px;
        padding-left: 10px;
        display: block;
        text-decoration: none;
        color: #596699;
        height: 30px;
        font-weight: bold;
}
#registro A:hover, #login A:hover
{
        background-color: #DDF0F8;
}
/*----contenido secundario-------------------------------------------------------*/
#contenido #secundario #form
{
        width: 280px;
        height: 290px;
        background: url(img_mod_insc.gif) no-repeat;
}
#header2
{
        width: 280px;
        height: 27px;
        background: url(img_header_principal2.png) no-repeat;
        padding: 2px 0 0;
}
#txtheader2
{
        margin: 0 0 0 11px;
```

```
	padding: 0;
	font-size: 120%;
	font-weight: bold;
}
#contform
{
	margin: 17px 0px 20px 8px;
	padding: 0;
}
LABEL.txtformulario
{
	color: #000000;
	width: 96px;
	float: left;
	text-align: right;
	margin-left: -25px;
	padding-top: 5px;
	padding-right: 10px;
}
#submit
{
	margin-left: 86px;
	margin-top: 8px;
	/*[empty]background-image:;*/
}
#submittForm
{
	height: 33px;
	font-weight: bolder;
	background: url(boton.gif) no-repeat;
	border: 0;
	cursor: hand;
	width: 100px;
	margin-top: -1px;
}
#leyenda
{
	border-top: 1px solid #D3D3D3;
	border-bottom: 1px solid #D3D3D3;
	height: 32px;
	width: 800px;
}
#txtLeyenda
{
	margin-top: 8px;
	color: #A5A0A0;
	font-size: 13px;
	text-align: center;
}
A#linkPoliticas
{
	text-decoration: none;
	color: #4594BC;
}
```

```
A#homepage
{
	text-decoration: none;
	color: #4594BC;
}
#formLogin
{
	float: left;
	width: 472px;
	height: 38px;
	margin-left: 27px;
}
#divLogin
{
	/*[empty]width:;*/
	height: 37px;
}
#divLogin #loginEmail
{
	margin-top: 9px;
	width: 182px;
	margin-left: 15px;
	color: #686363;
}
#divLogin #loginPass
{
	width: 97px;
	margin-left: 3px;
	color: #686363;
}
#divLogin #loginSubmit
{
	background: url(botonLogin.gif) no-repeat;
	border-right: 0;
	border-top: 0;
	border-left: 0;
	border-bottom: 0;
	height: 25px;
	width: 63px;
	text-align: center;
	font-weight: bolder;
	font-size: 11px;
	margin-left: 9px;
	margin-bottom: -1px;
	cursor: hand;
}
.inputForm
{
	margin-top: 4px;
}
FIELDSET
{
	height: 44px;
	width: 265px;
```

```
        padding-top: 4px;
}
FIELDSET.welldone
{
        background: transparent url(bg-fieldset-welldone.gif) no-repeat;
        height: 44px;
        border-top: 0px solid #000000;
        border-left: 0px solid #000000;
        border-bottom: 0px solid #000000;
        border-right: 0px solid #000000;
        width: 265px;
        padding-top: 4px;
        background-position-x: 240px;
}
FIELDSET.kindagood
{
        background: transparent url(bg-fieldset-kindagood.gif) no-repeat;
        height: 44px;
        width: 265px;
        border-top: 0px solid #000000;
        border-left: 0px solid #000000;
        border-bottom: 0px solid #000000;
        border-right: 0px solid #000000;
        padding-top: 4px;
        background-position-x: 240px;
}
#inputPass
{
        width: 149px;
}
SPAN.hint
{
        width: 27px;
        position: absolute;
        display: none;
        height: 31px;
        /*[disabled]background:transparenturl(bg-fieldset-welldone.gif) no-repeat;*/
        margin-top: -2px;
}
FIELDSET.error
{
        background: transparent url(bg-fieldset-error.gif) no-repeat;
        height: 44px;
        width: 265px;
        border-top: 0px solid #000000;
        border-left: 0px solid #000000;
        border-bottom: 0px solid #000000;
        border-right: 0px solid #000000;
        padding-top: 4px;
        background-position-x: 240px;
}
SPAN#pass
{
        font-size: 10px;
```

```
        position: absolute;
        margin-top: 26px;
        margin-left: -160px;
        color: #4A98BF;
}
```

Vinculando la hoja de estilo con la estructura HTML.

```
<head>
    <metahttp-equiv="Content-Type"content="text/html; charset=utf-8" />
    <link rel="stylesheet" type="text/css" href="style.css" media="screen"/>
</head>
```

Vista de la estructura HTML y el diseño en CSS anterior en un navegador IExplorer.

Ilustración 5: Pagina web con un estilo css aplicado

Se puede observar como la evolución de las tecnologías de la programación web ha evolucionado de mostrar un texto estático a mostrar un texto con un diseño más amigable, la ilustración 6, muestra la vista de una página web solo con la estructura HTML, la ilustración 7, muestra la misma página pero esta vez con un estilo css aplicado

Ilustración 6: Pagina web solo con la estructura HTML

Ilustración 7: Pagina web con un estilo css aplicado

2.7 TECNOLOGÍAS DE SCRIPTING / LENGUAJES DE PROGRAMACIÓN DE SCRIPTS.

Una de las ventajas que ha traído las nuevas tecnologías y técnicas a la programación web, es poder desarrollar páginas web más dinámicas e interactivas, por tanto, a continuación analizaremos las posibilidades que brindan esas diferentes tecnologías de scripting disponibles en la actualidad, con el objetivo de realizar consideraciones como desarrollar contenidos de Internet interactivos y dinámicos, desplegaremos las tecnologías de scripting en dos áreas, las diseñadas para ejecutarse del lado del cliente y las diseñadas para ejecutarse del lado del servidor, en las dos áreas se mencionaran los lenguajes y técnicas más comunes en la actualidad; como se ha mencionado anteriormente se mostrara un poco sobre cada uno de los lenguajes y técnicas de cada área, pero sin llegar a profundizar de lleno en cada uno de ellos, ya que este no es el objetivo de este proyecto, pero dándole la oportunidad al lector de brindarle páginas y direcciones web, de cada uno de los lenguajes descritos, para que obtengan toda la información relacionada.

2.7.1 Programación Del Lado Del Cliente.

Una de las áreas de la tecnología de scripting o lenguajes de programación de script es la programación del lado del cliente, esta tecnología presenta la ventaja que la ejecución de la aplicación se delega al cliente, con lo cual se evita recargar al servidor de trabajo, en general el servidor entregara código HTML con código script embebido, y el cliente lo reconocerá cuando este procesando el documento, realizando la ejecución del código. En este sentido es importante aclarar que esta práctica lleva a que el código quede publicado abiertamente, por lo que podemos decir que la programación del lado del cliente es intrínsecamente open source.

A continuación, se analizara el lenguaje JavaScript como lenguaje de programación del lado del cliente y la tecnología DOM (*DocumentObjectModel)*, aunque existen otros lenguajes de programación del lado del cliente se enfocara únicamente en JavaScript por ser un lenguaje utilizado por todos los navegadores.

2.7.1.1 JavaScript.

A principios de los años 90, la mayoría de usuarios que se conectaban a Internet lo hacían con módems a una velocidad máxima de 28.8 kbps. En esa época, empezaban a desarrollarse las primeras aplicaciones web y por tanto, las páginas web comenzaban a incluir formularios complejos.

Con unas aplicaciones web cada vez más complejas y una velocidad de navegación tan lenta, surgió la necesidad de un lenguaje de programación que se ejecutara en el navegador del usuario. De esta forma, si el usuario no rellenaba correctamente un formulario, no se le hacía esperar mucho tiempo hasta que el servidor volviera a mostrar el formulario indicando los errores existentes.

BrendanEich, un programador que trabajaba en Netscape, pensó que podría solucionar este problema adaptando otras tecnologías existentes (como *ScriptEase*) al navegador Netscape Navigator 2.0, que iba a lanzarse en 1995. Inicialmente, Eich denominó a su lenguaje *LiveScript*.

Posteriormente, Netscape firmó una alianza con Sun Microsystems para el desarrollo del nuevo lenguaje de programación. Además, justo antes del lanzamiento Netscape decidió cambiar el nombre por el de JavaScript.

¿Qué es y cómo funciona JavaScript?

JavaScript es un lenguaje de programación basado en objetos del lado del cliente, con el cual se pueden desarrollar página web dinámicas, técnicamente, JavaScript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con JavaScript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

¿Cómo identificar código JavaScript?

El código JavaScript se puede encontrar embebido en la página web dentro de las etiquetas <*body*></*body*> ó <*head*></*head*>, en donde el código se escribe dentro de las etiquetas<script></script>; otra forma de encontrar el código JavaScriptes utilizando ficheros externos usando <*script type="text/javascript" src="micodigo.js"*></*script*> dentro de la etiqueta <*head*></*head*>.

CARACTERÍSTICAS.

- Es interpretado por el cliente.
- Está basado en objetos
- Su código se integra en las páginas HTML, incluido en las propias páginas.
- No es necesario declarar los tipos de variables que van a utilizarse.
- Las referencias a objetos se comprueban en tiempo de ejecución, por lo tanto no se compila.

- No puede escribir automáticamente al disco duro.
- Admite programación estructurada.
- Multiplataforma.
- No se necesita ningún kit o entorno de desarrollo.

Para entender un poco más del uso del JavaScript, se seguirá con el ejemplo desarrollado anteriormente –la página de inicio de la red social-, y que hasta el momento se ha desarrollado la estructura y contenido en HTML aplicando diseño con CSS y ahora se le aplicara algunos Scripts para volverlo más dinámico.

```
//validando que el password contenga más de 3 caracteres y menor de 8
functioncheckPassword(whatYouTyped) {
	varfieldset = whatYouTyped.parentNode;
	var txt = whatYouTyped.value;
	if (txt.length> 3 &&txt.length< 8) {
		fieldset.className = "kindagood";
	} else {
		if (txt.length> 7) {
			fieldset.className = "welldone";
		} else {
			if(txt.length> 0){
				fieldset.className = "error";
			} else {
			fieldset.className = "";
			}
		}
	}
}
```

Vinculando la hoja de script a la estructura HTML y adicionando el evento al input que lo va a generar.

```
<head>
   <metahttp-equiv="Content-Type"content="text/html; charset=utf-8" />
   <link rel="stylesheet" type="text/css" href="style.css" media="screen"/>
   <script type="text/javascript" src="scriptIndex.js"></script>
</head>
```

```
<input
type="password"class="inputForm"name="pass"onkeyup="checkPassword(this);"id="inputPass"ma
xlength="8"/>
```

Vista de la pagina web con el script incorporado y ejecutándose en un navegador IExplorer.

Ilustración 8: Secuencia de la validación realizada con JavaScript

2.7.1.2 Document Object Model "DOM" y Dynamic HTML.

DOM o *DocumentObjectModel* es un conjunto de utilidades específicamente diseñadas para manipular documentos XML, pero también dada su extensión se puede utilizar para manipular documentos XHTML y HTML, técnicamente, DOM es una API de funciones que se pueden utilizar para manipular las páginas XHTML de forma rápida y eficiente.

En si DOM especifica cómo los objetos (texto, imágenes, enlaces, etc.) en una página web son representados, definiendo atributos asociados a cada objeto y

cómo estos objetos y sus atributos pueden ser manipulados. Por otro lado el Dynamic HTML (DHTML por sus siglas en inglés) se basa en el DOM para cambiar dinámicamente la apariencia de las páginas web después de que han sido descargadas por un navegador.

Antiguamente los dos navegadores líderes (Netscape e Internet Explorer) usaban diferentes DOM. Esto contribuía a que las implementaciones para el HTML dinámico en ambos navegadores fuesen totalmente diferentes, dado este inconveniente ambas compañías presentaron sus DOM a la W3C para estandarizarlas, en donde finalmente la W3C decidió crear un modelo único de DOM para que las compañías se adapten a este.

Aunque en ocasiones DOM se asocia con la programación web y con JavaScript, la API de DOM es independiente de cualquier lenguaje de programación. De hecho, DOM está disponible en la mayoría de lenguajes de programación comúnmente empleados.

CÓMO FUNCIONA EL DOM EN LA PROGRAMACIÓN WEB.

Una vez cargada la página en el navegador, el API DOM construye la estructura original en una estructura más fácil de manejar formada por una jerarquía de nodos. De esta forma, DOM transforma la estructura ya sea XML, HTML o XHTM en una serie de nodos interconectados en forma de *árbol*.

El siguiente ejemplo muestra como a partir de estructura en XHTML se construye una estructura de nodos.

```
<! DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<htmlxmlns="http://www.w3.org/1999/xhtml">

<Head>

<metahttp-equiv="Content-Type"content="text/html; charset=iso-8859-1" />

<Title>Páginasencilla</title>

</head>

<Body>

<p>Esta página es <strong>muy sencilla</strong></p>

</body>

</html>
```

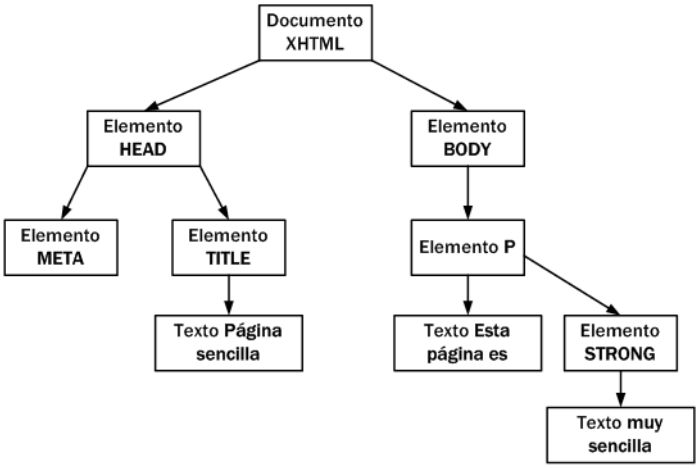


Ilustración 9: Árbol de la estructura DOM de una página web

Una vez creada la estructura DOM, se puede acceder a las propiedades (Atributos) de cada nodo, ya sea modificando su contenido o creando un nuevo contenido.

Si desean encontrar toda la información relacionada con el tema se pueden referir a la sección de anexos (Anexo I).

2.7.2 Programación Del Lado Del Servidor.

En el tema anterior se mencionó la programación del lado del cliente, en este tema se especificara que es y cómo funciona la programación del lado del servidor, adicionalmente se mencionara sobre los cuatro lenguajes de programación del lado del servidor más comunes como son: PHP, JSP, ASP (incluyendo la versión ASP.Net) y Ruby onRails, como se ha mencionado a lo largo de todo este proyecto, no se entraran en detalle sobre ninguno de los lenguajes, si no que se hará una pequeña introducción de que es y cómo funcionan, mostrando algunos ejemplos de estos, pero brindándole al lector los links de las páginas web oficiales de cada uno de ellos, en donde, si el lector lo desea, puede entrar más en detalle de cada uno de los lenguajes expuestos.

2.7.2.1 ¿Qué es la programación del lado servidor?

La programación del lado del servidor es una tecnología que consiste en el procesamiento de una petición de un usuario mediante la interpretación de un script en el servidor web para generar páginas HTML dinámicamente como respuesta.

2.7.2.2 ¿Cómo funciona la programación del lado del servidor?

Básicamente, un cliente realiza una petición de una página web enviando una URL (y quizás algo de información extra, luego el servidor ejecuta un script de servidor, adicionalmente este script puede interactuar con una base de datos u otra fuente de datos externa como ser un archivo XML, ya sea poder guardar los datos, u obtener datos adicionales, finalmente el script genera código HTML y este es enviado al cliente.

2.7.2.3 PHP.

PHP (HipertextPreprocesor) es un lenguaje de programación, con una sintaxis similar a los lenguajes C y Perl, que se interpreta por un servidor web Apache generando código HTML dinámico, lo cual permite crear programas que se puedan ejecutar en el servidor desde un navegador y dar respuestas en función de los datos introducidos por el usuario.

FUNCIONES DE PHP.

PHP dispone de funciones para realizar las operaciones habituales de los lenguajes de programación, teniendo como una característica importante la facilidad que presenta para consultar bases de datos y el generar paginas en función de los resultados obtenidos en la correspondiente consulta.

VENTAJAS DE PHP.

PHP presenta múltiples ventajas frente a otros lenguajes de programación lo que hace que este lenguaje se imponga como una alternativa para el desarrollo de todo tipo de aplicaciones web.

Interfaz. PHP no necesita de tener que instalar ningún software adicional para poder utilizarlo, ya que permite que el código se embebido dentro de la estructura HTML, o bien, utilizado desde un archivo aparte.

Protección del código. Al tener el código ejecutable albergado en el servidor, este se encuentra protegido (invisible) tanto de la manipulación de los usuarios como de la presencia de virus.

Programación orientada a objetos. Al igual que lenguajes de programación como Java, PHP permite la programación orientada objetos, definiendo clases que disponen de datos y funciones propias de esta.

ESTRUCTURA DE PHP.

Como se mencionó anteriormente, el lenguaje PHP puede ser embebido o no dentro de la estructura HTML, indicando el código PHP dentro de las etiquetas <? - código php –?> ó <?php –codigophp -?>.

Por otra parte al tener una sintaxis similar a C, se encuentran estructuras condicionales como ***if – else***, estructuras cíclicas como ***for – while***, entre otras.

Para entender mejor la estructura del lenguaje PHP, se seguirá con el ejemplo de la página de inicio de la red social, en donde se creara un login que tendrá acceso a la base de datos para validar los datos del usuario.

Archivocontrol.php

```
<?
```

```
    //inclusion de clasesession_start();
    Include ("class_socialNeworkSystem.php");
    Include ("class_systemDirectory.php");

    $inf = newsystemDirectory ();
    $login = newsocialNetworkSystem ();

    //variables pasadapor POST
    $email = $_POST ["inputEmail"];
    $pass = $_POST ["inputPass"];

    //variables propias
    $valido = false;
    $valido = $login->login($email,$pass);

    if ($valido){
        //defino una sesión y guardo datos
        session_register("autentificado");
        $autentificado = "SI";
        $usercod = $inf->getUserCod ($email);
        Header ("Location: aplicacionPrueba.php?user=$usercod");
    } Else {
        //si no existe le mando otra vez a la portadaheader ("Location: login.php?errorusuario=si");
    }
?>
```

Incluyendo el archivo PHP en la estructura HTML de la página de inicio.

```
<form action="control.php"method="post"name="frm">
```

Si desean encontrar toda la información relacionada con el tema se pueden referir a la sección de anexos (Anexo I).

2.7.2.4 JSP.

JSPJavaServerPages es una tecnología para el desarrollo de páginas web que incluyan contenido dinámico. A diferencia de una página HTML normal, que contiene contenido estático que es siempre el mismo, una página JSP puede cambiar su contenido sobre la base de cualquier número de elementos variables,

incluyendo la identidad del usuario, el tipo de navegador del usuario, la información proporcionada por el usuario, y selecciones hechas por el usuario (bibliografía). Esta funcionalidad es clave para las aplicaciones web, tales como compras en línea, directorios de empleados, entre otras aplicaciones más.

Al igual que en una página web PHP, un página JSP contiene los elementos estándar de la estructura HTML, al igual que una página web normal. Sin embargo, una página JSP también contiene elementos especiales de JSP que permiten al servidor para insertar contenido dinámico en la página.

Los scripts JSP se pueden utilizar para una variedad de propósitos, tales como recuperar información desde una base de datos o al registrar las preferencias del usuario. Cuando un usuario pide una página JSP, el servidor ejecuta los scripts JSP, combina los resultados con las partes estáticas de la página y envía la página dinámica compuesta de vuelta al navegador, como se muestra en la figura siguiente.

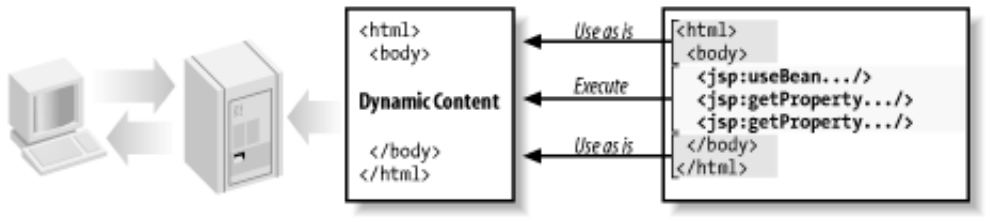


Ilustración 10: Proceso realizado por el lenguaje JSP para mostrar una página web

FUNCIONES Y ESTRUCTURA DE JSP.

A diferencia del lenguaje de programación PHP que solo utiliza scripts del mismo lenguaje, JSP permite incluir código Java mediante de tres formas: Expresiones, *scriptlets* y declaraciones.

- **Expresiones:** Son fragmentos de código Java, con la forma <%= expresión %> que se evalúan y se muestran en la salida del navegador. En general, dentro de una expresión se puede usar cualquier código que se usaría dentro de un *System.out.print (expr);*

```
<%= "Tamaño de cadena: "+cadena.length() %>
<%= new java.util.Date () %>
<%= Math.PI*2 %>
```

- **Scriptlets:** Son fragmentos de código Java con la forma <% código %>, en general, se puede insertar cualquier código que se utilizaría dentro de una función Java. Para acceder a la salida del navegador, usamos el objeto implícito out.

```
<%
    out.println ("<table>");
    For (inti=0;i<10;i++){
        out.println ("<tr><td>"+i+"</td></tr>");
        out.println ("</table>");
    }
%>
```

- **Declaraciones:** Contienen declaraciones de variables o métodos, con la forma <%! declaración %>. Estas variables o métodos serán accesibles desde cualquier lugar de la página JSP. Hay que tener en cuenta que el servidor transforma la página JSP en un *servlet*, y éste es usado por múltiples peticiones, lo que provoca que las variables conserven su valor entre sucesivas ejecuciones.

```
<%!intnumeroAccesos=0; %>
<Html>
    <Body>
        <%="La página ha sido accedida "+(++numeroAccesos)+ " veces desde el arranque del servidor"%>
     </Body>
</Html>
```

Otra funcionalidad del lenguaje JSP es la integración con los servlets, de este modo, las páginas JSP se utilizan para la capa de presentación únicamente, y los servlets para tareas de procesamiento, o para la lógica del negocio. Los Servlets actúan como el controlador responsable del procesamiento de las solicitudes y la creación de los Beans necesarios para la página JSP, este también es responsable de decidir a qué página JSP se transmitirá la solicitud, la cual recupera los objetos creados por losservlets y extrae el contenido dinámico para su inserción dentro de una plantilla.

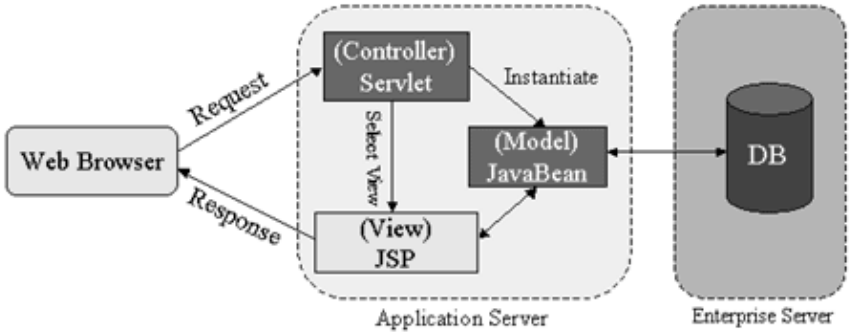


Ilustración 11: Estructura del modelo vista controlador MVC propuesto por JSP

Este modelo promueve el uso de la arquitectura MVC *(Model View Controller)*, el cual se describirá más adelante.

VENTAJAS DE JSP.

Integración con clases Java. La principal ventaja de JSP frente a otros lenguajes es que permite integrarse con clases Java (.class) lo que permite separar en niveles las aplicaciones web, almacenando en clases java las partes

que consumen más recursos así como las que requieren más seguridad, y dejando la parte encargada de formatear el documento HTML en el archivo JSP.

Paginas dinámicas más robustas. El utilizar clases Java en los servlets de JSP, permite crear páginas web más dinámicas, al abrir la posibilidad de utilizar un lenguaje robusto, con muy pocas limitaciones.

Si desean encontrar toda la información relacionada con el tema se pueden referir a la sección de anexos (Anexo I).

2.7.2.5 ASP.Net.

ASP.net (Active Server Pages .net Framework), es la tecnología de scripting desarrollado por Microsoft para servidores IIS (Internet InformationService), para la programación dinámica de sitios web, introducido en julio del 2000 conocido como ASP+ en la versión 1.0 de .Net Framework.

Al igual que los demás programas de programación web, su código se es embebido dentro de la estructura HTML de la página, en el cual se puede utilizar cualquier lenguaje de programación del .Net Framework gracias a la tecnologia CLR (CommonLanguageRuntime), lo que brinda la gran ventaja a los programadores de desarrollar aplicaciones web ASP con saber cualquier lenguaje del .Net Framework.

Ejemplo de una página ASP utilizando el lenguaje C# de .Net Framework.

```
<%@ Page Language="C#" %>
```

```
<! DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<script runat="server">
Protected void Page_Load (object sender, EventArgs e)
   {
      Label1.Text = DateTime.Now.ToLongDateString ();
   }
</script>
<html xmlns="http://www.w3.org/1999/xhtml" >
<head runat="server">
<Title>Página de Ejemplo</title>
</head>
<Body>
<form id="form1" runat="server">
<Div>
<asp: Labelrunat="server" id="Label1" />
</div>
</form>

</body>
</html>
```

ESTRUCTURA ASP.NET.

Las páginas de ASP.NET, conocidas oficialmente como "web forms" (formularios web), son el principal medio de construcción para el desarrollo de aplicaciones

web. Los formularios web están contenidos en archivos con una extensión ASPX; en jerga de programación, estos archivos típicamente contienen etiquetas HTML o XHTML estático, y también etiquetas definiendo Controles Web que se procesan del lado del servidor y Controles de Usuario donde los desarrolladores colocan todo el código estático y dinámico requerido por la página web.

Adicionalmente, el código dinámico que se ejecuta en el servidor puede ser colocado en una página dentro de un bloque <% -- código ASP.Net -- %> que es muy similar a otras tecnologías de desarrollo como PHP y JSP, pero esta práctica es, generalmente, desaconsejada excepto para propósitos de enlace de datos pues requiere más llamadas cuando se genera la página; para evitar el uso de embeber el script en la estructura HTML, Microsoft recomienda el uso del modelo *code-behind*, que coloca el código en un archivo separado o en una etiqueta de script especialmente diseñada.

VENTAJAS DE ASP.NET.

Poder y Flexibilidad. Dado que ASP.Net está basado en el Entorno Común de Ejecución de Lenguajes (CommonLanguageRuntime, CLR), el poder y la flexibilidad de la plataforma .Net está disponible para los desarrolladores, permitiendo el uso de una gran variedad de lenguajes de programación y, por tanto, el desarrollador puede escoger el mejor lenguaje para la aplicación, o peticionarla en varios lenguajes.

Simplicidad. ASP.Net hace fácil el ejecutar tareas comunes, desde el simple envío de un formulario o la autenticación de un cliente, hasta el despliegue y la configuración de un Web. Por ejemplo, el entorno de paginado de ASP.Net permite construir interfaces de usuario que separan limpiamente la lógica de la

aplicación del código de la presentación, manejando eventos con un modelo sencillo de procesamiento de formularios al estilo de Visual Basic.

Gerenciabilidad. ASP.Net usa un sistema jerárquico de configuración, basado en archivos de texto, que simplifica la aplicación de parámetros de configuración al servidor y sus aplicaciones. Porque la información de configuración es almacenada como texto, nuevos parámetros pueden ser configurados sin recurrir a herramientas de administración locales. Esta filosofía de "cero administración local" también se extiende al despliegue de aplicaciones de ASP.Net. Una aplicación de ASP.Net se despliega a un servidor simplemente copiando los archivos necesarios al servidor. No hay que reiniciar el servidor, ni siquiera para reemplazar código compilado que ya está en servicio.

Escalabilidad y Disponibilidad. ASP.Net ha sido diseñado para la escalabilidad con características específicamente dirigidas a mejorar el funcionamiento de servidores racimados (clustered) y de servidores con procesadores múltiples. Los procesos del servidor son vigilados y gerenciados por el entorno del ambiente de ejecución de ASP.NET, así que si algún proceso se entorpece o se detiene, un nuevo proceso puede ser creado para reemplazarlo, lo cual ayuda a mantener la disponibilidad de su aplicación para manejar solicitudes de servicio.

Si desean encontrar toda la información relacionada con el tema se pueden referir a la sección de anexos (Anexo I).

2.7.2.6 Ruby onRails.

Ruby onRails, también conocido como RoR o Rails es un framework de aplicaciones web de código abierto escrito en el lenguaje de programación Ruby, siguiendo el paradigma de la arquitectura Modelo Vista Controlador (MVC). Trata de combinar la simplicidad con la posibilidad de desarrollar aplicaciones del

mundo real escribiendo menos código que con otros frameworks y con un mínimo de configuración. El lenguaje de programación Ruby permite la metaprogramación, de la cual Rails hace uso, lo que resulta en una sintaxis que muchos de sus usuarios encuentran muy legible. Rails se distribuye a través de RubyGems, que es el formato oficial de paquete y canal de distribución de bibliotecas y aplicaciones Ruby.

ESTRUCTURA DE RUBY ON RAILS.

La estructura de Rails reside en dos principios en cuales se basa: menos software y convención sobre configuración.

Menos software quiere decir que se escriben menos líneas de código para implementar la aplicación. Si el código es pequeño quiere decir que el desarrollo es más rápido y con menos errores, lo que hará que el código sea fácil de entender, mantener y mejorar.

Convención sobre configuración significa que se elimina la configuración en los archivos XML. En lugar de archivos de configuración, una aplicación Rails utiliza una serie de convenciones simples que le permiten averiguarle todo mediante reflexión y descubrimiento.

VENTAJAS DE RUBY ON RAILS.

Poderoso. Ruby onRails es completamente orientado a objetos.
Código más legible. Conduce a una mejor práctica de nomenclaturas

Conciso. Menos líneas de código, alta productividad.

Si desean encontrar toda la información relacionada con el tema se pueden referir a la sección de anexos (Anexo I).

2.8 APLICACIONES WEB ENRIQUECIDAS.

La tendencia actual de la programación web es generar contenidos e interfaces cada vez más dinámicas para el usuario, permitiendo la creación de sitios que puede hacer mucho más que presentar páginas y conectarlas vía hipervínculos; lo que hace necesario evitar que una página se recargue al requerir información del servidor y de esta forma acercarse mas a la rapidez y sensación que genera una aplicación de escritorio, para esto la vista de una página debe ser generada y manipulada en el navegador del cliente, utilizando la conexión cliente-servidor solo para intercambio de datos.

Existen diversas aproximaciones técnicas para esto, como utilizar Java Applers, Flash, y Ajax, en donde el servidor es el responsable de manejar los datos y la generación del entrono gráfico, mientras que el cliente muestra dicho entorno y maneja las acciones de los usuarios.

Para entender un poco más sobre las aplicaciones web enriquecidas, a continuación se mencionara la técnica Ajax, detallando exclusivamente el que es, como funciona, mostrando un ejemplo de la página de inicio de la red social, ahora utilizando esta técnica.

2.8.1 Ajax.

AJAX (Asynchronous JavaScript + XML) no es una tecnología en sí mismo, si no, que se trata de varias tecnologías independientes que se unen de formas nuevas y sorprendentes, tecnologías como:

- XHTML y CSS, para crear una presentación basada en estándares.
- DOM, para la interacción y manipulación dinámica de la presentación.
- XML, XSLT y JSON, para el intercambio y la manipulación de información.
- XMLHttpRequest, para el intercambio asíncrono de información.
- JavaScript, para unir todas las demás tecnologías.

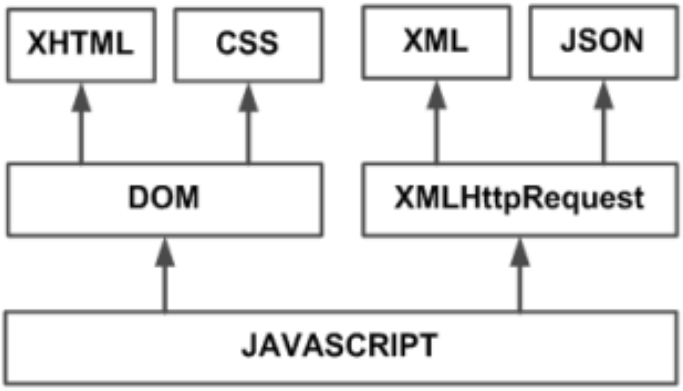


Ilustración 12: Tecnologías del AJAX

COMO FUNCIONA AJAX.

En las aplicaciones web tradicionales, las acciones del usuario en la página desencadenan llamadas al servidor, una vez procesada la petición del usuario, el

servidor devuelve una nueva página HTML al navegador del usuario, lo cual hace que al realizar peticiones continuas al servidor, el usuario debe esperar a que se recargue la página con los cambios solicitados.

AJAX permite mejorar completamente la interacción del usuario con la aplicación, evitando las recargas constantes de la página, ya que el intercambio de información con el servidor se produce en un segundo plano.

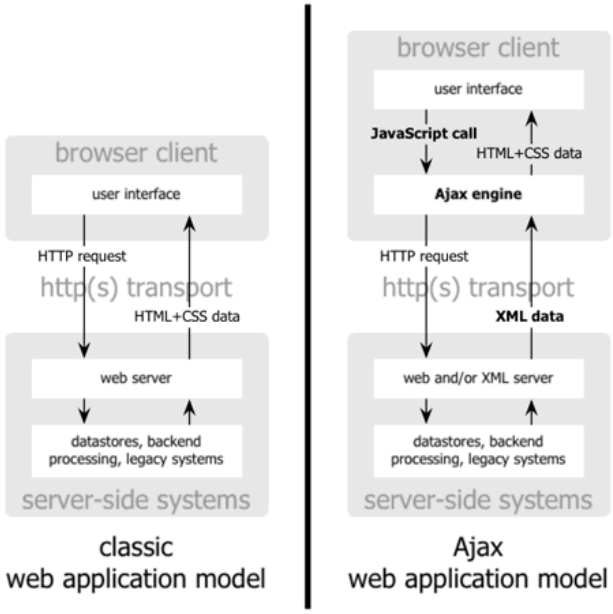


Ilustración 13: Comparación del proceso normal de una aplicación web con el proceso realizado por AJAX

A continuación se muestra la diferencia más importante entre una aplicación web tradicional y una aplicación web creada con AJAX.

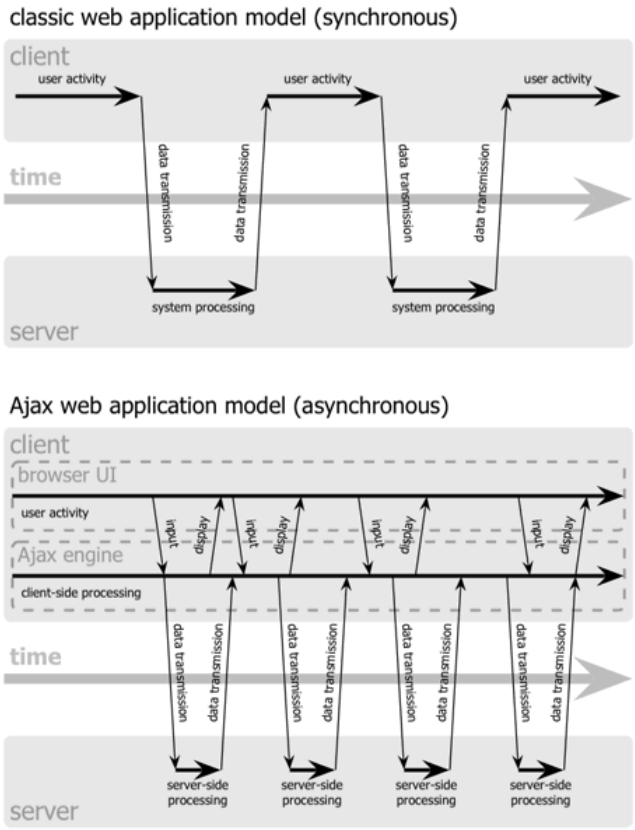


Ilustración 14: Proceso síncrono y asíncrono en una aplicación web

Para entender un poco más sobre Ajax, se continuara con el ejemplo de la página de inicio de la red social, cambiando la estructura utilizada en la programación tradicional con PHP, al uso de Ajax y PHP.

Modificando el script de JavaScript

```
FunctionObjectXmlHttp (){
    Varxmlhttp= false;
    Try {
        // No IE
        Xmlhttp= new ActiveXObject ("Msxml2.XMLHTTP");
    }
    Catch (e){
        Try {
            // IE
            Xmlhttp= new ActiveXObject ("Microsoft. XMLHTTP");
        } catch (E) {
            Xmlhttp= false;
        }
    }
    if (!xmlhttp&&typeofXMLHttpRequest!="undefined") {
        Xmlhttp= new XMLHttpRequest();
    } return xmlhttp;
}

Functionlogin(){
    var email = frm.inputEmail.value;
    var pass = frm.inputPass.value;
    if (email!=""&&email!="email"&&pass!=""&&pass!="password"){
        Varajax=ObjectXmlHttp ();
        ajax.open ("POST", "control.php", true); ajax.setRequestHeader("Content-Type",
          "application/x-www-form-urlencoded"); ajax.send("email="+email+"&pass="+pass);
        ajax.onreadystatechange= function (){
            If (ajax.readyState==4){
                var respuesta=ajax.responseText;
                if(respuesta!=error){
                    window.location=”aplicacionPrueba.php?user=”+respuesta;
                }Else{
                    If(respuesta==error){
                        window.location=”login.php?errorusuario=si”;
                    }
                }
            }
        }
    } Else {
        Alert ("Debe digitar la infor.....")
    }
}
```

Modificando la estructura del HTML.

```
<div id="formLogin">
    <form name="frm">
        <div id="divLogin">
            <input type="text" id="loginEmail" name="inputEmail" value="email"/>
            <input type="password" id="loginPass" name="inputPass" value="password"/>
            <input type="button"onclick="login()" id="loginSubmit" value="Ingresar"/>
        </div>
    </form>
</div
```

Modificando el archivo control.php.

```
<?
    //inclusion de clasesession_start();
    include ("class_socialNeworkSystem.php");
    include ("class_systemDirectory.php");

    $inf = newsystemDirectory();
    $login = newsocialNetworkSystem();

    //variables pasadapor POST
    $email = $_POST["inputEmail"];
    $pass = $_POST["inputPass"];

    //variables propias
    $valido = false;
    $valido = $login->login($email,$pass);

    if ($valido){
        //defino una sesion y guardo datos
        session_register("autentificado");
        $autentificado = "SI";
        $usercod = $inf->getUserCod($email);
        echo $usercod;
    } else {
        echo “error”;
    }
?>
```

2.9 FRAMEWORK WEB.

El termino Framework, se refiere a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un Framework se puede considerar como una aplicación genérica incompleta y configurable a la que se le puede añadir las últimas piezas para construir una aplicación concreta.

Los objetivos principales que persigue un Framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones.

Ahora entendido que es un Framework, se definirá en sí que es un Framework web.

Un Framework web es un Frameworks diseñado para el desarrollo de sitios web dinámicos, aplicaciones web y servicios web, disminuyendo la sobrecarga asociada a las actividades comunes que se realizan en el desarrollo Web. Por ejemplo, brindar bibliotecas para el acceso a la base de datos, la gestión de sesiones de usuario, y promueven a menudo la reutilización de código.

2.9.1 Tipos de Frameworks Web.

Existen varios tipos de Frameworks Web: orientados a la interfaz de usuario, como Java Server Faces, orientados a aplicaciones de publicación de documentos, como Coocon, orientados a la parte de control de eventos, como Struts y algunos que incluyen varios elementos como Tapestry.

CARACTERÍSTICAS.

A continuación se enunciaran una serie de características comunes en prácticamente todos los Frameworks existentes.

Abstracción de URLs y sesiones. No es necesario manipular directamente las URLs ni las sesiones, el Framework se encarga de hacerlo.

Acceso a datos. Incluyen las herramientas e interfaces necesarias para integrarse con herramientas de acceso a datos.

Controladores. La mayoría de Frameworks implementa una serie de controladores para gestionar eventos, como una introducción de datos mediante un formulario o el acceso a una página. Estos controladores suelen ser fácilmente adaptables a las necesidades de un proyecto concreto.

Autentificación y control. Incluyen mecanismos para la identificación de acceso a usuarios, mediante login y password, permitiendo restringir el acceso a determinas páginas a determinados usuarios.

2.10 W3C.

El Consorcio World Wide Web (W3C por sus siglas en inglés) es una comunidad internacional donde las organizaciones Miembro, personal a tiempo completo y el público en general trabajan conjuntamente para desarrollar estándares Web. Liderado por el inventor de la Web Tim Berners-Lee y el Director Ejecutivo Jeffrey Jaffe, la misión del W3C es guiar la Web hacia su máximo potencial.

OBJETIVOS DEL W3C.

El objetivo del W3C es guiar la Web hacia su máximo potencial a través del desarrollo de protocolos y pautas que aseguren el crecimiento futuro de la Web.

2.11 EL PATRÓN MVC.

El patrón Modelo-Vista-Controlador (MVC) es una guía para el diseño de arquitecturas de aplicaciones que ofrecen una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema.

Como se mencionó arriba el patrón MVC separa la lógica de negocio y la presentación por lo que se consigue un mantenimiento más sencillo de las aplicaciones, siendo así, si una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original.

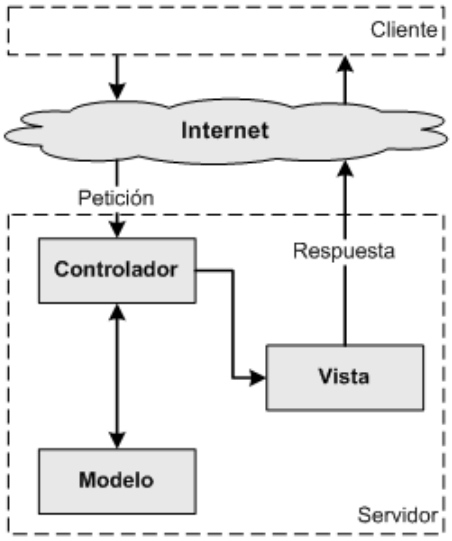


Ilustración 15: Estructura del patrón Modelo Vista Controlador

2.11.1 Las Capas De La Arquitectura MVC.

- *El modelo* representa la información con la que trabaja la aplicación, es decir, su lógica de negocio.

- *La vista* transforma el modelo en una página web que permite al usuario interactuar con ella.

- *El controlador* se encarga de procesar las interacciones del usuario y realiza los cambios apropiados en el modelo o en la vista.

2.11.2 FrameworksQue Utilizan Este Modelo.

- **Para PHP:** Sitellite, SolarPHP, sQeletor, Studs, struts4php, Symfony, TaniPHP.

- **Para Ruby:** RubyonRails.
- **Para Java:** Turbine, Expresso, Struts.
- **Para .net:** MonoRail, Asp.Net MVC 3.

2.12 CONCLUSIÓN.

Con el desarrollo efectuado en el marco teórico, se hace posible considerar que se ha comprendido los siguientes objetivos específicos.

- Analizar y entender cómo funciona la estrategia AIEPI para poder determinar cuál es la mejor forma de sistematización a aplicar.
- Estudiar y comprender tecnologías y estándares que forman la base para el desarrollo de proyectos Web.

El desarrollo del marco teórico ha sido un proceso largo pero sumamente interesante, pero, con el cual se aplicara todo el conocimiento obtenido para desarrollar el modelo de analisis y diseño del proyecto.

3. MODELO DE ANALISIS Y DISEÑO.

3.1 INTRODUCCIÓN.

En base al desarrollo del marco teórico se ha logrado comprender los elementos que componen la programación Web, entendiendo sus formas y tecnologías. Ya llegado a esta etapa del proyecto se buscara generar un modelo de análisis y diseño, en el cual se desarrollara las bases para la implementación y concreción del S.I buscando así, lograr cumplir con el objetivo general del proyecto y poder darle solución al problema planteado.

En esta sección se buscar el poder cumplimentar los siguientes objetivos específicos:

- Analizar detalladamente las necesidades de los actores de negocio involucrados para poder definir todos los requerimientos necesarios que ayuden en la construcción de SI.
- Modelar dicho sitio y sus interfaces, en base a la aplicación del patrón de arquitectura de software MVC, considerando las tecnologías y estándares estudiados.

Para poder cumplimentar con los objetivos mencionados anteriormente, se empezara con definir la planificación del proyecto, identificando las etapas y actividades que se necesitaran para el desarrollo del proyecto, siendo esta la guía en el tiempo que se usara para poder cumplir a cabalidad con el estudio del mismo.

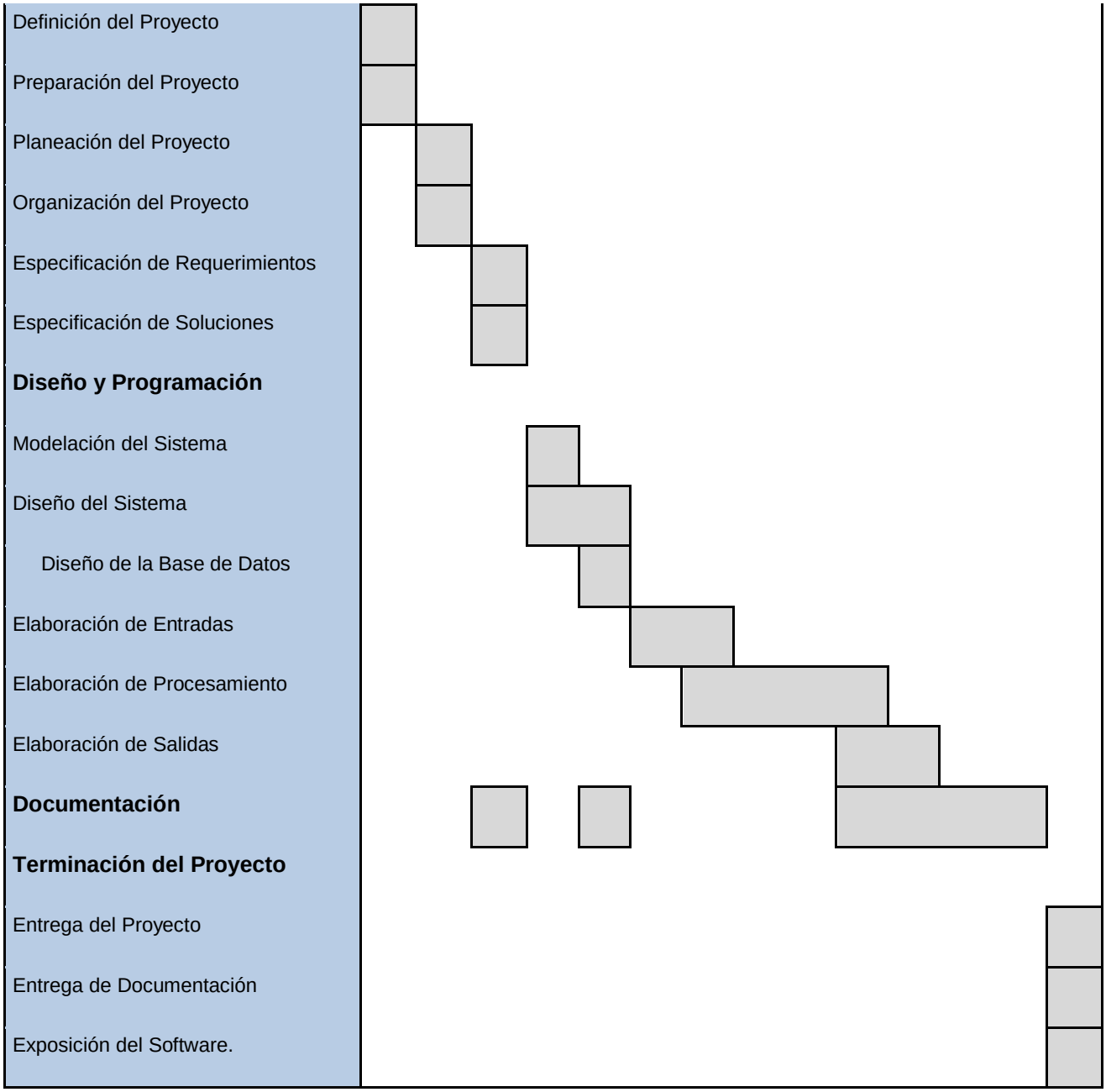
3.2 PLANIFICACIÓN.

3.2.1 Etapas, Actividades Y Duración.

Las actividades consideradas a continuación serán realizadas por el autor del trabajo, por tanto no se hace necesario la división de tareas, estas actividades están comprendidas durante un periodo de tiempo empezando el 15 de diciembre de 2011 y finalizando el 16 de marzo de 2012.

Tabla 2: Cronograma de actividades a realizar en este proyecto

Tiempo	**Diciembre**			**Enero**					**Febrero**				**Marzo**	
Actividades	**S51**	**S52**	**S53**	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**	**S7**	**S8**	**S9**	**S10**	**S11**
Análisis y Planeación														



3.3 REQUERIMIENTOS.

3.3.1 Definición De Actores De Negocio.

Como bien indica el nombre de este proyecto, se seguirá la misma estructura de negocio con que se ha desarrollado el sistema de información SAIM, identificando dos actores de negocio, Administrador, siendo este el personal de soporte del departamento de sistemas y los usuarios siendo estos médicos.

En donde estos actores podrán encontrar funciones únicas y compartidas dependiendo de la necesidad de cada caso.

3.3.2 Necesidades Genéricas Detectadas En Los Actores De Negocio.

Usuario:
Realizar la encuesta de la estrategia AIEPI de una manera fácil y rápida.
Poder que el sistema arroje las impresiones diagnosticas dependiendo de las repuestas suministradas por el paciente ha rea lizarle la encuesta.
Tener la posibilidad de observar en cualquier momento los formatos realizados.
Mostrar el formato de encuesta apropiado al paciente según la edad del mismo.

Administrador:
Tener la posibilidad de crear nuevos formatos y actualizar y/o modificar los formatos existentes.
Poder realizar reportes estadísticos con base a la información de las encuestas realizadas.

3.3.3 RequerimientosDel Sistema.

En base a las necesidades genéricas detectadas a los actores de negocio y estudiando cada una de ellas a fondo, se ha realizado una tabla de requerimientos tanto funcionales y no funcionales donde se reflejan en si todas esas necesidades detectadas.

Tabla 3: Requerimientos del sistema para el proyecto propuesto

Cód.	Descripción	Tipo	Fuente	Priori	Referencia

				dad	
RE01	El sistema permitirá la creación y modificación de áreas, preguntas, respuestas, diagnósticos y condición de diagnósticos que van a conformar cada uno de los formatos de la estrategia AIEPI que son (RECIEN NACIDOS, NIÑOS DE 0 A 2 MESES Y NIÑOS DE 2 MESES A 5 AÑOS)	F	Cliente	ALTA	NINGUNA
RE02	El sistema deberá ser capaz de mostrar al usuario los diagnósticos semaforizados dependiendo de la configuración que pudiera tener en su creación.	F	Cliente	ALTA	RE01
RE03	El sistema deberá ser capaz tener la opción de activar/desactivar, las áreas, preguntas y diagnósticos que conforman cada uno de los formatos	F	Cliente	ALTA	RE01
RE04	El sistema no podrá eliminar las áreas, preguntas y diagnósticos de cada uno de los formatos.	F	Cliente	ALTA	RE01
RE05	El sistema permitirá la realización de cada uno de los formatos de la estrategia AIEPI, identificando al paciente en el sistema dado su tipo y número de identificación, mostrando el formato adecuado según la edad del paciente.	F	Cliente	ALTA	RE01
RE06	El sistema debe permitirá la consulta de cada uno de los formatos que se le hayan realizado al paciente, identificando al paciente dado su tipo y número de identificación	F	Cliente	ALTA	RE05
RE07	El sistema no debe eliminar los formatos realizados a los pacientes de la base de datos.	F	Cliente	ALTA	RE05
RE08	El sistema realizar reportes estadísticos con los datos de los formatos realizados a los pacientes, el tipo de reporte será escogido por el usuario	F	Cliente	ALTA	RE05

RE09	Los reportes generados del sistema deberán ser en formato PDF	NF	Cliente	ALTA	NINGUNA
RE10	El sistema deberá acoplarse al sistema general la clínica SAIM, como una opción más dentro del menú de Consulta Externa	NF	Cliente	ALTA	NINGUNA
RE11	La Interfaz de Usuario debe ser atractiva, amigable y fácil de usar.	NF	Desarrollador	MEDIA	ESTANDAR
RE12	El sistema se desarrollara utilizando el lenguaje, estructura y diseño, con el que se ha venido desarrollado SAIM	NF	Cliente	ALTA	NINGUNA
RE13	El sistema se debe desarrollar bajo la arquitectura de MVC.	NF	Cliente	ALTA	NINGUNA
RE14	El sistema debe soportar que un mismo programa sea usado por dos o más usuarios distintos.	NF	Cliente	ALTA	NINGUNA
RE15	Debe ser manejo sencillo y de rápido aprendizaje	NF	Desarrollador	MEDIA	ESTANDAR
RE16	Los acceso deben ser fácilmente detectables	NF	Desarrollador	MEDIA	ESTANDAR

3.4 ANÁLISIS Y DISEÑO.

En base a los estudios de las necesidades de los actores y estableciendo los requerimientos del sistema, se desarrollará el análisis y diseño de sistemas en

donde se empezara a reflejar la construcción del sistema, no se entrara en detalle de la programación, siendo a que no es este el foco principal del proyecto.

Para lograr una comprensión del análisis y diseño del sistema, utilizaremos las siguientes herramientas:

Análisis.

Listado de casos de uso.
Descripción de los casos de uso.
Diagrama de casos de uso.

Diseño.

Diagrama E-R: Esquema base de datos
Diagramas de secuencia
Diagramas de Flujo de Procesos
Decisiones de diseño de sistema.

3.4.1 AnálisisDel Sistema.

3.4.1.1 Listado de casos de uso del sistema.

En base a lo analizado anteriormente en las necesidades de los actores y en los requerimientos obtenidos, se listara los casos de uso del sistema en donde más adelante se describirán los más relevantes.

Creación Formato de AIEPI
Creación de áreas
Creación de preguntas
Creación de diagnósticos
Creación condición de diagnósticos
Modificación Formato de AIEPI
RealizaciónEncuestaAIEPI
ReportesEncuestaAIEPI
Reportes Estadísticos AIEPI

3.4.1.2 Descripción de los casos de uso del sistema.

Anteriormente se han listado todos los casos de uso que comprende el sistema, en este etapa, se describirán aquellos casos de uso más relevantes.

Tabla 4: Descripción caso de uso creación de formato

Nombre	**CreacionFormatoDeAIEPI**	**Código**	**CU-01**
Versión	1.0	**Frecuencia**	Alta
Fecha Ult.Act.	15-ene-12	**Prioridad**	Alta
Actor(es) Primario(s)	Administrador.		
Actor(es) Secundario(s)			
Requisitos Funcionales	RE01		
Alcance	Se mostrara las diferentes opciones para la creación de los formatos (crear área, crear preguntas, crear respuestas, crear diagnósticos, crear condición diagnósticos)		
Nivel	Componente.		
Objetivos	Crear y configurar los formatos de encuesta del AIEPI		
Descripción	Creación de los formatos del AIEPI		
Pre condiciones	Realizar Login. Tener asignado el permiso		
Reglas de negocio			
Trigger	Administrador.		

Flujo normal o básico			
#	**Actor**	**#**	**Sistema**
1.1	Ingresar Login.	1.2	Validación. Mostrar permisos asignados

2.1	Opción creación	2.2	Muestra vista creación formato AIEPI
3.1	Selecciona opción (crear área, crear preguntas, crear respuestas, crear diagnósticos, crear condición diagnósticos)	3.2	Muestra la vista e información requerida para cada una de las opciones seleccionadas, así como de la información creada para el formato perteneciente a cada una de las opciones.
4.1	Asignar información (asignar área, asignar preguntas, asignar respuestas, asignar diagnósticos, asignar condición diagnósticos)	4.2	Validación. Asignación de la información al formato.
Flujo Excepcional o Alternativo			
#	**Actor**	**#**	**Sistema**
Post condiciones	Asignación realizada correctamente		
Observaciones			

Nombre	**ModificacionFormatoDeAIEPI**	**Código**	**CU-02**
Versión	1.0	**Frecuencia**	Alta
Fecha Ult.Act.	15-ene-12	**Prioridad**	Alta
Actor(es) Primario(s)	Administrador.		
Actor(es) Secundario(s)			
Requisitos Funcionales	RE01,RE03,RE04.		
Alcance	En cada una de las diferentes opciones (crear área, crear preguntas, crear respuestas, crear diagnósticos, crear condición diagnósticos) se mostrara la información perteneciente del formato para cada una de ellas.		

Nivel	Componente.
Objetivos	Modificar las diferentes características que conforman los formatos.
Descripción	Modificación de los formatos del AIEPI
Pre condiciones	Realizar Login. Tener asignado el permiso
Reglas de negocio	No se pueden eliminar las características de los formatos (área, pregunta, diagnostico), solo se podrán activar o desactivar.
Trigger	Administrador.

Flujo normal o básico			
#	**Actor**	**#**	**Sistema**
1.1	Ingresar Login.	1.2	Validación. Mostrar permisos asignados
2.1	Opción creación	2.2	Muestra vista creación formato AIEPI
3.1	Selecciona opción (crear área, crear preguntas, crear respuestas, crear diagnósticos, crear condición diagnósticos)	3.2	Muestra la vista e información requerida para cada una de las opciones seleccionadas, así como de la información creada para el formato perteneciente a cada una de las opciones.
4.1	Asignar/Desactivar característica dentro de cada una de las opciones seleccionadas.	4.2	Activa o Desactiva la característica seleccionada.

Flujo Excepcional o Alternativo			
#	**Actor**	**#**	**Sistema**

Post condiciones	Cambio visual del estado de la característica
Observaciones	

Tabla 5: Descripción caso de uso realización de la encuesta

Nombre	**RealizacionEncuestaAIEPI**	**Código**	**CU-03**
Versión	1.0	**Frecuencia**	Alta
Fecha Ult. Act.	13-ene-12	**Prioridad**	Alta
Actor(es) Primario(s)	Usuario		
Actor(es) Secundario(s)			
Requisitos Funcionales	RE05		
Alcance	Se mostrara un formato de AIEPI (Recién Nacidos, Niños de 0 a 2 Meses, Niños de 2 meses a 5 Años), según la edad del paciente		
Nivel	Componente.		
Objetivos	Aplicar la estrategia AIEPI al paciente.		
Descripción	Encuesta estrategia AIEPI		
Pre condiciones	Realizar Login. Tener asignado el permiso		
Reglas de negocio	Debe mostrarse el formato correspondiente a la edad del paciente, y desplegar las preguntas y respuestas dependiendo la edad del paciente.		
Trigger	Usuario.		

Flujo normal o básico			
#	**Actor**	**#**	**Sistema**
1.1	Ingresar Login.	1.2	Validación. Mostrar permisos asignados
2.1	Opción Encuesta	2.2	Muestra la vista Encuesta, solicitando el tipo y numero de documento de identificación del paciente.

3.1	Ingresar el tipo y numero de documento de identificación del paciente	3.2	Valida si el usuario tiene una admisión activa, Valida la edad del paciente, muestra el formato correspondiente a la edad del paciente.
4.1	Realizar encuesta por área, enviar resultados.	4.2	Valida las condiciones de los diagnósticos pertenecientes a cada área, con las preguntas y respuestas enviadas por el usuario. Visualiza el/los diagnostico según la condición cumplida
5.1	Finalizar encuesta	5.2	Muestra subreporte de la encuesta realizada, para ingresar observaciones por parte del usuario, Puede finalizar o Imprimir reporte
6.1	Finalizar/Imprimir reporte	6.2	Guarda las observaciones del usuario y muestra o no un reporte en formato **pdf** de la encuesta realizada al paciente
Flujo Excepcional o Alternativo			
#	**Actor**	**#**	**Sistema**
Post condiciones	Visualiza o no el formato de la encuesta realizada.		
Observaciones			

Tabla 6: Descripción caso de uso reportes de la encuesta

Nombre	**ReportesEncuestaAIEPI**	**Código**	**CU-04**
Versión	1.0	**Frecuencia**	Alta
Fecha Ult. Act.	13-ene-12	**Prioridad**	Alta
Actor(es) Primario(s)	Usuario		

Actor(es) Secundario(s)	Administrador.
Requisitos Funcionales	RE06
Alcance	Se mostrara listado con todos los formatos realizados al paciente, indicando fecha, tipo formato realizado, usuario que realizo el formato.
Nivel	Componente.
Objetivos	Mostrar los formatos de todas las estrategias AIEPI realizadas al paciente
Descripción	Reportes estrategia AIEPI
Pre condiciones	Realizar Login. Tener asignado el permiso
Reglas de negocio	Mostrarse todos los formatos realizados al paciente, los formatos no pueden ser eliminados.
Trigger	Usuario, Administrador.

Flujo normal o básico

#	Actor	#	Sistema
1.1	Ingresar Login.	1.2	Validación. Mostrar permisos asignados
2.1	Opción Reportes	2.2	Muestra la vista Reportes, solicitando el tipo y numero de documento de identificación del paciente.
3.1	Ingresar el tipo y numero de documento de identificación del paciente	3.2	Muestra en pantalla todos los formatos realizados al paciente, indicando: fecha de realización, usuario quien realizo el formato, tipo de formato aplicado, y opción para poder ver el formato realizado.
4.1	Ver cada uno de los formatos realizados.	4.2	Despliega el formato realizado en un archivo de tipo **pdf**.

Flujo Excepcional o Alternativo

#	Actor	#	Sistema
Post condiciones	Visualiza el formato de la encuesta realizada.		
Observaciones			

Tabla 7: Descripción caso de uso reportes estadisticos

Nombre	**ReportesEstadisticosAIEPI**	**Código**	**CU-05**
Versión	1.0	**Frecuencia**	Alta
Fecha Ult. Act.		**Prioridad**	Alta
Actor(es) Primario(s)	Administrador.		
Actor(es) Secundario(s)			
Requisitos Funcionales	RE08		
Alcance	Se mostrara datos estadísticos (diagnósticos más prevalentes, diagnósticos más prevalentes por tipo de estrategia aplicada) dentro de un periodo de tiempo.		
Nivel	Componente.		
Objetivos	Mostrar datos estadísticos en base a los formatos aplicados durante un periodo de tiempo.		
Descripción	Estadística estrategia AIEPI		
Pre condiciones	Realizar Login. Tener asignado el permiso		
Reglas de negocio			
Trigger	Administrador.		
Flujo normal o básico			
#	**Actor**	**#**	**Sistema**

1.1	Ingresar Login.	1.2	Validación. Mostrar permisos asignados
2.1	Opción Estadística	2.2	Muestra la vista Estadística solicitando, tipo de reporte y periodo de tiempo.
3.1	Mostrar reporte.	3.2	Muestra un reporte en un archivo **pdf**, según el tipo de reporte seleccionado y el periodo de tiempo.
Flujo Excepcional o Alternativo			
#	**Actor**	**#**	**Sistema**
Post condiciones	Visualiza los datos estadísticos.		
Observaciones			

3.4.1.3 Diagrama de casos de uso del sistema.

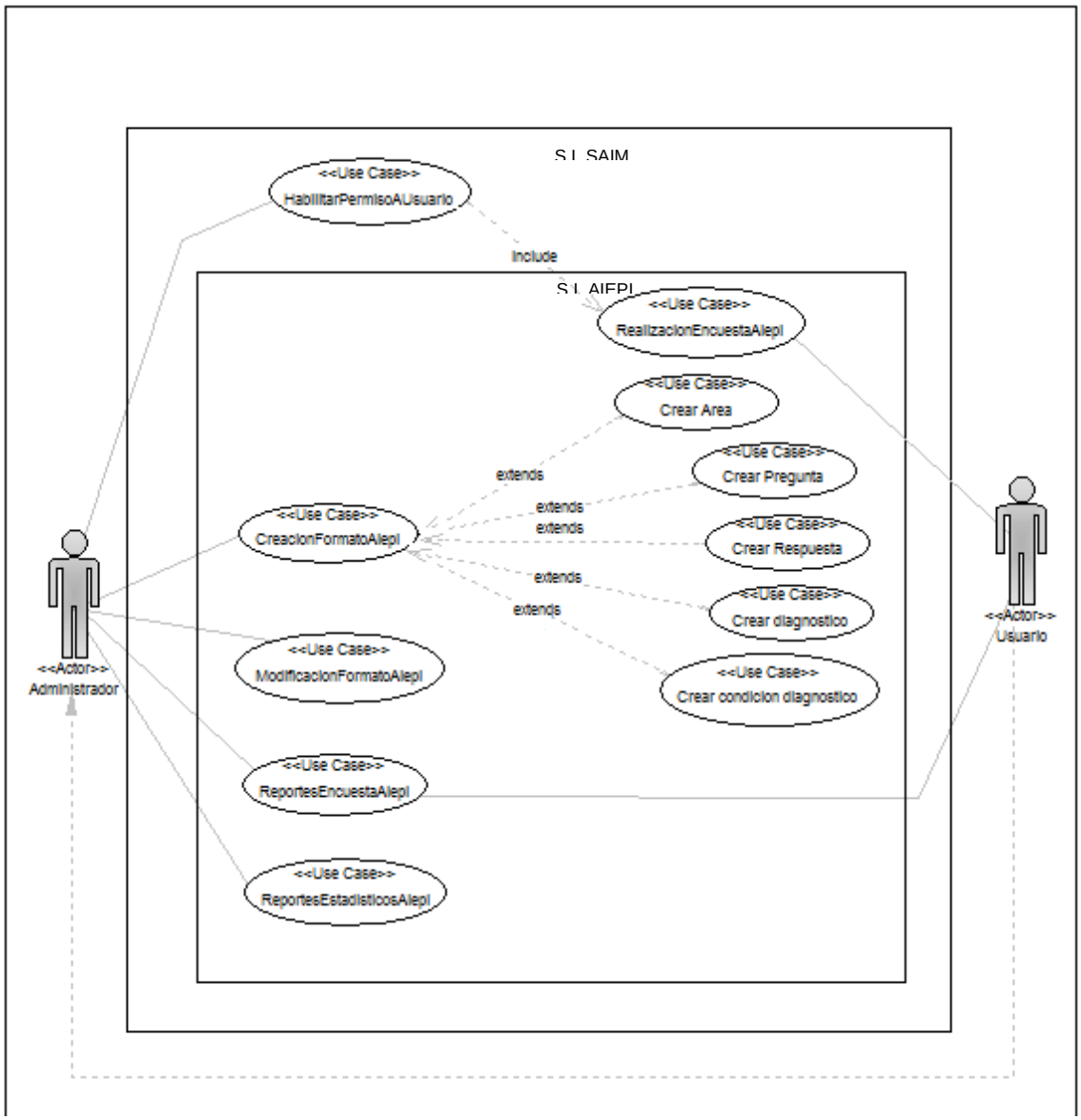


Ilustración 16: Diagrama de caso de usos del S:I desarrollado en este proyecto

3.4.1.4 Diagrama relacional.

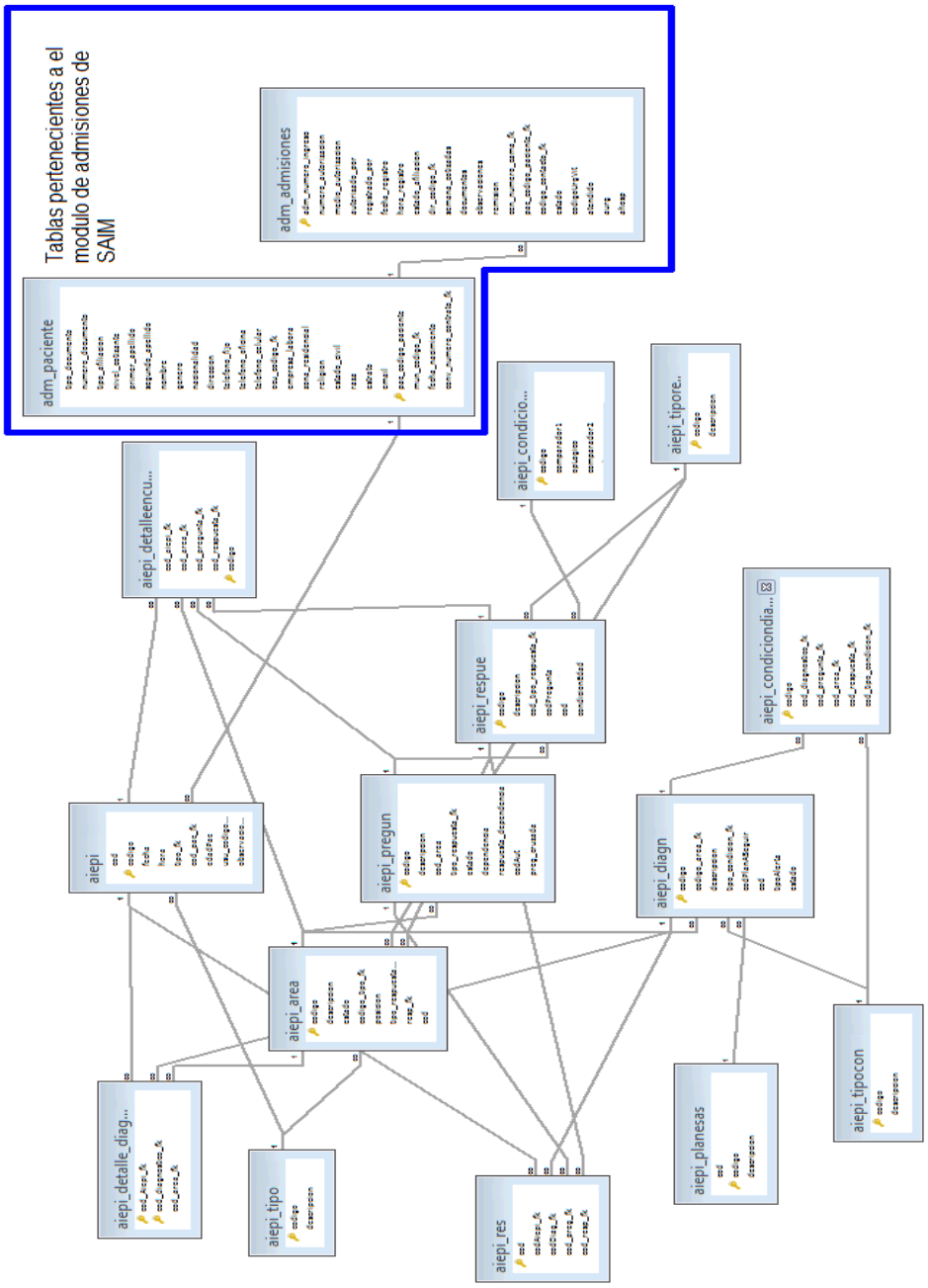


Ilustración 17: Diagrama relacional

3.4.1.5 Diagramas de Secuencia.

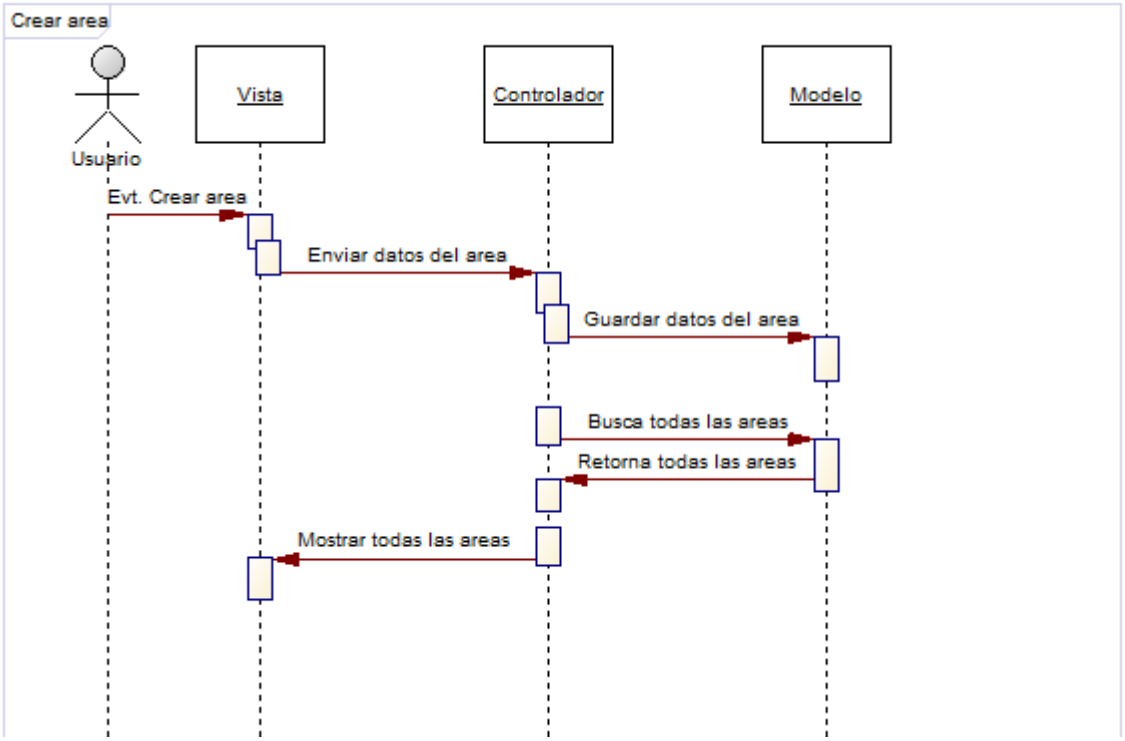
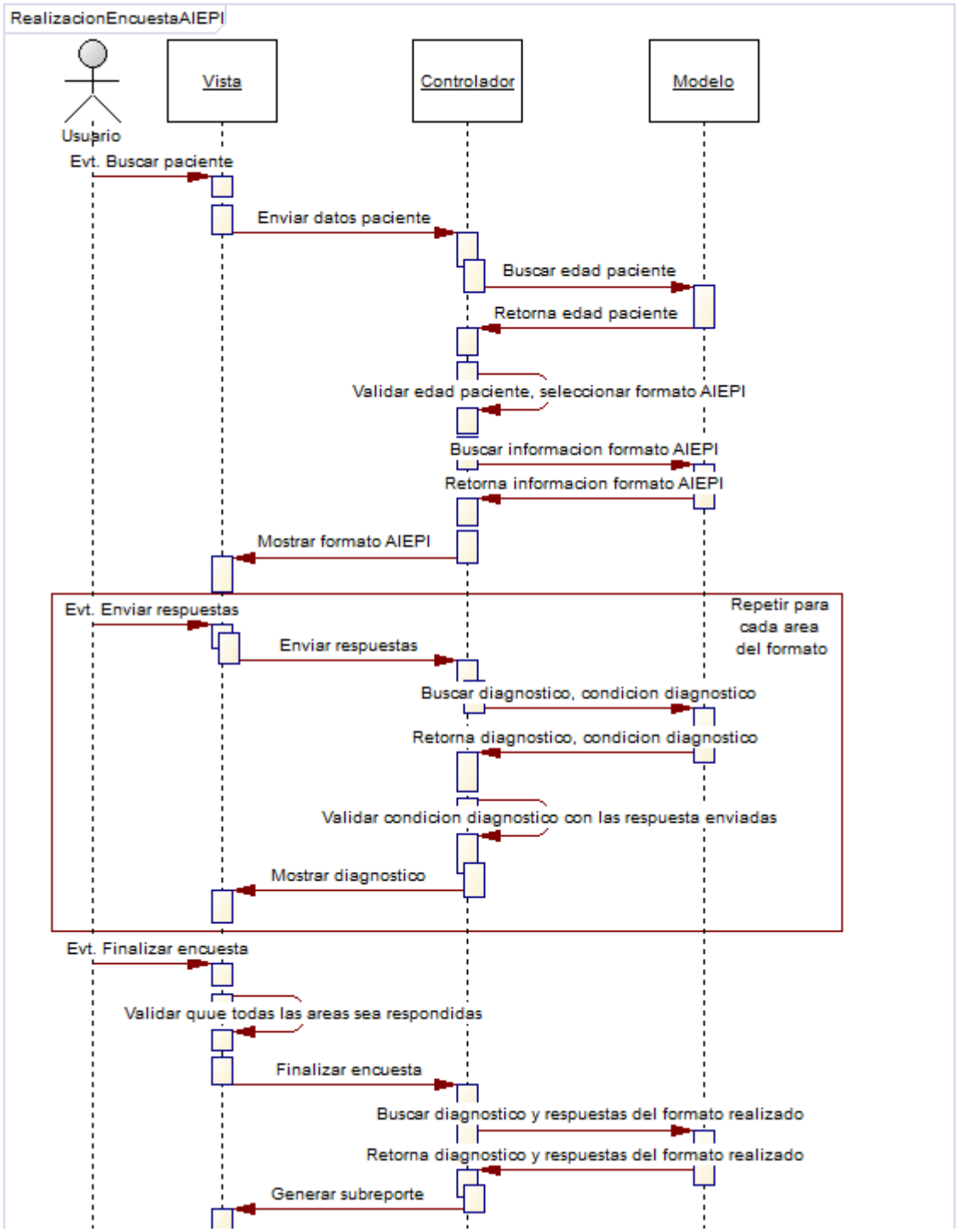


Ilustración 18: Diagrama de secuencia creacion de un area


RealizacionEncuestaAIEPI
Usuario
Vista
Controlador
Modelo
Evt. Buscar paciente
Enviar datos paciente
Buscar edad paciente
Retorna edad paciente
Validar edad paciente, seleccionar formato AIEPI
Buscar informacion formato AIEPI
Retorna informacion formato AIEPI
Mostrar formato AIEPI
Evt. Enviar respuestas
Repetir para cada area del formato
Enviar respuestas
Buscar diagnostico, condicion diagnostico
Retorna diagnostico, condicion diagnostico
Validar condicion diagnostico con las respuesta enviadas
Mostrar diagnostico
Evt. Finalizar encuesta
Validar quue todas las areas sea respondidas
Finalizar encuesta
Buscar diagnostico y respuestas del formato realizado
Retorna diagnostico y respuestas del formato realizado
Generar subreporte

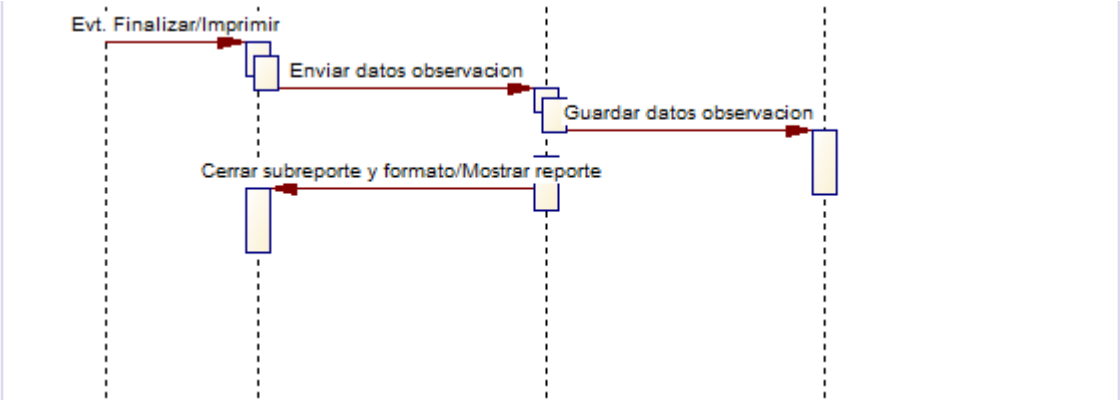


Ilustración 19: Diagrama de secuencia ralizacion de la encuesta

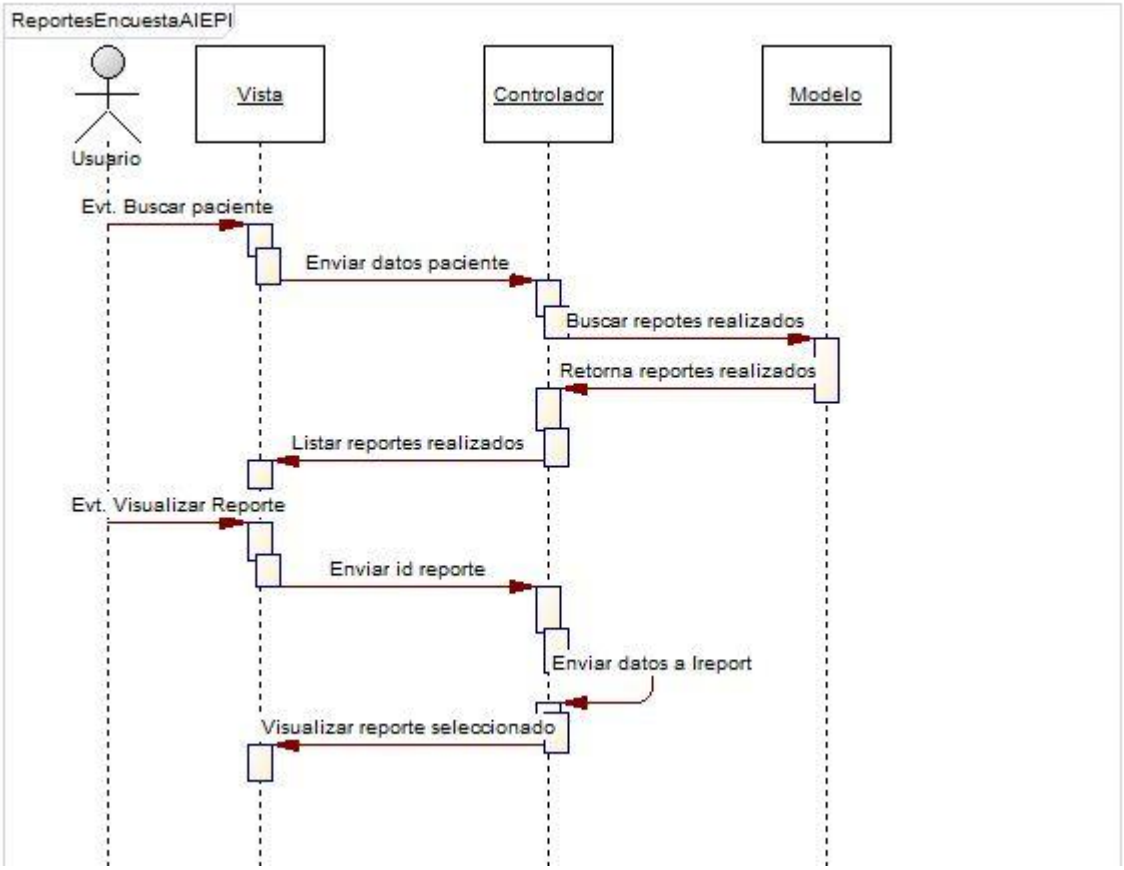


Ilustración 20: Diagrama de secuencia reportes de encuestas

3.4.2 Decisiones De Diseño Del Sistema.

Para complementar lo desarrollado en el análisis y diseño del sistema, se añadirá las siguientes consideraciones de diseño.

Modelo: Se elige el modelo cliente servidor ya que el sistema debe ser necesariamente un sitio web de Internet. Sera necesario considerar los análisis realizados en el marco teórico sobre tecnologías de scripting especificas para cada componente del modelo.

Plataforma base del sistema: Se utilizara como plataforma base los lenguajes de programación JSP (Java Server Page) y Java, ya que como se ha mencionado anteriormente este nuevo modulo pertenecerá a el S.I. Saín, el cual está hecho bajo estos mismos lenguajes y por la ventaja de poder tener por separado los módulos que se encargan de hacer los procesos de datos de los que se encargan de presentar visualmente dichos datos.

Arquitectura utilizada: Para el desarrollo de S.I. se utilizara la arquitectura de software MVC (Modelo Vista Controlador), mencionada en el marco teórico. Se utilizara esta arquitectura ya que permite mantener por separado el modelo del negocio y la vista del usuario proporcionando más seguridad a el S.I.

Base de Datos: Se elige trabajar sobre una base de datos MySQL por su probada eficiencia y eficacia en numerosos proyectos existentes en Internet.

Aplicaciones de internet enriquecidas: Se utilizar la técnica de desarrollo AJAX, para poder brindarle a el usuario unas páginas web más dinámicas y rápidas al momento de actualizar la vista, ya que como se ha mencionado en el marco teórico esta tecnología utilizando su propiedad de asíncrona brinda la ventaja de

poder ir actualizando la pagina sin tener la necesidad de recargarla completamente.

Desarrollo de interfaces: Se realizaran interfaces DHTML y DOM con hojas de estilo CSS buscando la menor cantidad posible de gráficos para lograr interfaces agradables, simples y rápidas; teniendo así paginas mas ligeras, mas dinámicas y mas rápidas, sin tener necesario la comunicación completamente con el servidor para la actualización de la vista.

Limitación de navegador: Se utilizara IExplorer y sus versiones como navegador único para el S.I. por ser este el navegador de más conocido por el usuario común.

3.5 CONCLUSIÓN.

Se puede considerar que se ha logrado cumplir los dos objetivos planteados para esta etapa del proyecto:

Analizar detalladamente las necesidades de los actores de negocio involucrados para poder definir todos los requerimientos necesarios que ayuden en la construcción de SI.

Modelar dicho sitio y sus interfaces, en base a la aplicación del patrón de arquitectura de software MVC, considerando las tecnologías y estándares estudiados.

El análisis teórico realizado anteriormente y el estudio de la situación problemática han brindado requerimientos para el desarrollo de una aplicación de software que se ajuste al objetivo principal del trabajo.

Posteriormente el uso de UWE UML ha brindado la capacidad de afianzar el diseño del sistema desarrollado con el UML tradicional.

La conclusión de esta etapa del proyecto nos brinda toda la información necesaria que ayudara a la concreción del proyecto, -paso siguiente a seguir en la elaboración-, en el cual se utilizara todo el diseño realizado en el modelo de analisis y diseño para la implementación y concreción final de este proyecto.

4. IMPLEMENTACIÓN Y PUESTA EN MARCHA.

4.1 INTRODUCCIÓN.

Se ha llegado a la etapa final de este proyecto, en la cual se buscar con el cumplimiento del objetivo general concretando la digitalización y sistematización de la AIEPI, la cual se convertirá en un nuevo módulo para el sistema de información SAIM, desarrollado por la clínica de la costa.

En esta etapa se buscar dar cumplimiento a los siguientes objetivos específicos:

- Implementación y puesta en marcha del sistema digital del AIEPI
- Fusionar el nuevo sistema con el sistema existente en la clínica
- Capacitar al personal de la clínica para el uso del sistema

En busca de cumplir con estos objetivos se analizaran inicialmente Lenguajes, tecnologías y servicios de base datos elegidos para la base del desarrollo, realizando una descripción del porque el uso de estas.

Posteriormente se mostrara parte de la solución de software implementada, en donde se podrán observar interfaces actuales de la versión funcional del proyecto.

Por último, se analizará la puesta en marcha considerando las necesidades técnicas del módulo, así como también la capacitación de los usuarios y la incorporación de este nuevo módulo al sistema de información SAIM.

4.2 IMPLEMENTACIÓN.

4.2.1 Lenguajes, Tecnologías Y Servicios De Base Datos Utilizados.

En parte los lenguajes, tecnologías, patrones y servicio de base de datos utilizados en el desarrollo de la sistematización y digitalización de la AIEPI, fueron escogidos por ser lo utilizados en el sistema de información SAIM -estos se utilizaron sin la ayuda de ningún framework, manejando la versión pura de cada uno de ellos- realizando cambios que a la vez fueron adoptados para la programación futura de nuevos módulos para el sistema; por otra parte viendo la necesidad de liberar un poco el uso del servidor, se optó por utilizar una tecnología usada en la programación del lado del cliente para evitar sobrecargas innecesarias a el servidor.

Si bien se ha definido en el marco teórico, los lenguajes y técnicas mencionadas a continuación, no se entrara de nuevo en las definiciones de cada una de estas, si no que se mencionara el porqué, o cuales fueron las ventajas de estas técnicasque hicieron que fuesen escogidas para el desarrollo tanto del proyecto como para el sistema de información SAIM.

Adicionalmente se mostrara parte del código en el que se refleja la utilización de cada una de los lenguajes y técnicas.

JSP.

JSP brinda la oportunidad de integrar lenguaje JAVA por medio de Servlets, lo cual permite separar en niveles las aplicaciones web, ofreciendo más seguridad en el desarrollo de estas aplicaciones, por otra parte, al utilizar lenguaje JAVA junto

con JSP se maximiza la posibilidad de crear aplicaciones más robustas y con un mínimo de limitaciones para el desarrollador.

Dada la posibilidad de crear sitios web más seguros y el tener separado en niveles la aplicación web, ha sido la decisión por la cual se ha elegido, JSP como el lenguaje de programación web del lado del servidor elegido para desarrollar el sistema de información SAIM como del módulo de sistematización y digitalización de la AIEPI.

El siguiente código es la verificación de la sesión del usuario logueado, código desarrollado en JSP.

```
<% if(codigou.equals("")){%>
<script language="javascript">alert("Usted No Tiene Permiso Para ver esta Pagina....");
window.location.href="Seg_login.jsp";</script>
<%}else{ %>
<%Conexion con=new Conexion();
ResultSet rs2;
Statement st2 = null;
String idu="";
st2 = con.conn.createStatement();
rs2=st2.executeQuery("select usuario from seg_usuario  where
usu_codigo="+codigou+"");
%>
<%while(rs2.next()){
idu=rs2.getString(1);
}
rs2.getStatement().close();
con.cerrar();
%>
```

Lo siguiente es el código del controlador de diagnostico función guardar condición del diagnostico realizado con JAVA Servlets.

```
public class ControlConfiguracion extends HttpServlet {
private static final long serialVersionUID = 1L;

protected void doPost(HttpServletRequestreq, HttpServletResponse res)
throwsServletException, IOException {
res.setContentType("text/html;charset=UTF-8");
```

```
String opcion = req.getParameter("op");
PrintWriter out = res.getWriter();
if(opcion.equals("guardarCondicionDiagnostico")){
String codDiag = req.getParameter("codDiag");
String condiciones = req.getParameter("condiciones");
String[] vecCondiciones = condiciones.split(",");
for (int i=0;i<vecCondiciones.length;i++){
String[] detCondiciones = vecCondiciones[i].split("-");
Stringcondicion = detCondiciones[0];
String pregunta = detCondiciones[1];
String respuesta = detCondiciones[2];
String[] detPregunta = pregunta.split("_");
Stringarea = detPregunta[1];
mc.guardarCondicion(codDiag, pregunta, area, respuesta, condicion);
}
}
….
}
```

Ajax.

La ventaja de tener una comunicación asíncrona entre el servidor y el cliente, hace de Ajax una de las tecnologías más usadas para el desarrollo de aplicaciones web, brindándole al usuario una vista continua de la página sin tener que realizar una recarga de esta para mostrar una actualización de contenido de la misma.

Dado que tanto el sistema de información SAIM como el módulo de sistematización y digitalización de la AIEPI, necesitan tener una comunicación constante con el servidor, ya sea para realizar cambios de contenidos o realizar querys a la base de datos, seria engorroso el tener que para cada una de las acciones o eventos lanzados por el usuario a el servidor la página tuviera que recargarse, es por esto que se hizo uso de la tecnología Ajax, para poder brindarle al usuario una vista continua de esta, en donde el usuario fuese transparente a los cambios.

A continuación se mostrara la forma en cómo se utilizo AJAX en el S.I.

Creación del Objeto XML según el navegador

```
functiongetXMLObject(){
varxmlhttp;
if (typeof (XMLHttpRequest) != 'undefined') {
try { xmlhttp = newXMLHttpRequest();
} catch (e) { }
} else {
try {
xmlhttp = newActiveXObject('Microsoft.XMLHTTP');
} catch (e) {
xmlhttp = newActiveXObject('Msxml2.XMLHTTP');
}
  }
returnxmlhttp;
}
```

Funcion AJAX, para buscar las preguntas del area

```
varxmlhttp = getXMLObject();
xmlhttp.onreadystatechange = function (){
if (xmlhttp.readyState == 4) {
if (xmlhttp.status==200){
divArea.innerHTML=xmlhttp.responseText;
}
});
xmlhttp.open("POST","ControlConfiguracion",true);
xmlhttp.setRequestHeader("Content-Type","application/x-www-form-urlencoded");
xmlhttp.send("op=2&codArea="+codArea+"&anio="+anio+"&meses="+meses+"&dias="+dias);
```

Esta es la forma básica de la implementación del AJAX en donde se necesita un objeto XML del navegador para luego hacer la comunicación de este con el servidor.

DOM.

El tener una estructura en árbol de la página web, y el poder manipular cada uno de los nodos pertenecientes a el árbol mediante JavaScript, hace de DOM una de las tecnologías más utilizadas en la programación web, ya que permite un uso apropiado de la comunicación del cliente con el servidor, evitando llamadas innecesarias a este, para entender un poco más acerca de lo que se puede hacer con DOM sin necesidad de estar llamando innecesariamente al servidor, se explicara una ejemplo de un funcionamiento de eliminar artículos de una tabla.

> *Se tiene una tabla de artículos con un link para eliminar, en donde el usuario puede eliminar cada uno de los artículos de esa tabla, el link mediante Ajax se comunica con el servidor para borrar el elemento seleccionado de la base de datos, una vez borrado el artículo, el servidor lee nuevamente todos los artículos de la base de datos y por medio de Ajax vuelve a actualizar la página.*

Si bien, este sería un proceso normal que haría todo programador en las aplicaciones web, hay dos procesos ejecutados por el servidor que se podrían omitir liberando uso del servidor por medio de DOM. Ahora bien, el mismo ejemplo de la eliminación del elemento esta vez utilizando DOM.

> *Se tiene una tabla de artículos con un link para eliminar, en donde el usuario puede eliminar cada uno de los artículos de esa tabla, el link mediante Ajax se comunica con el servidor para borrar el elemento seleccionado de la base de datos,*

una vez borrado el artículo, mediante de DOM se elimina la línea de este, el cual es un proceso que se hace del lado del cliente sin tener que llamar al servidor para que muestre la información de los artículos.

Este fue una de las ventajas por la que se ha optado DOM como una tecnología a usar en el proyecto y la cual ha sido adoptada para el sistema de información SAIM, para la modificación de módulos existentes y la creación de nuevos módulos para este.

El código a continuación es parte del código utilizado en la programación del S.I propuesto en la tesis y en donde se ve reflejado la utilización de la tecnología DOM y el cual se utilizo para la lectura de las respuestas de las preguntas de cada área.

```
varcont1 = document.getElementById('DIV'+area).childNodes;
varcant = cont1[1].childNodes.length;
varvect = cont1[1].childNodes;
vardivDiag = cont1[2];
varVRb = newArray;
vark=0; vars=0;
varVSb = newArray;
varl=0; varVTb = newArray;
for (vari = 0;i<cant;i++){
varid = vect[i].getAttribute('id');
id = id.substring(0,4);
if (id=="preg"){
varvect2 = vect[i].childNodes;
for(varj = 0; j<vect2.length;j++){
```

```
if(vect2[j].nodeName=="DIV"){
if (vect2[j].childNodes.length==1){
        if (vect2[j].childNodes[0].type=="radio"){
        VRb[k]=vect2[j].childNodes[0]; k++; }
        if (vect2[j].childNodes[0].nodeName=="SELECT"){
        VSb[s]=vect2[j].childNodes[0]; s++; }
        if (vect2[j].childNodes[0].type=="text"){
        VTb[l]=vect2[j].childNodes[0]; l++;
        }
                }else{
        varvectTemp = vect2[j].childNodes;
        for (varu=0;u<vect2[j].childNodes.length;u++){
        id = vectTemp[u].getAttribute('id');
        id = id.substring(0,4);
        if (id=="preg"){
        varvect2Temp = vectTemp[u].childNodes;
        for(varw = 0; w<vect2Temp.length;w++){
        if(vect2Temp[w].nodeName=="DIV"){
        if (vect2Temp[w].childNodes[0].type=="radio"){
        VRb[k]=vect2Temp[w].childNodes[0]; k++;
        }
        if (vect2Temp[w].childNodes[0].nodeName=="SELECT"){
                        VSb[s]=vect2Temp[w].childNodes[0]; s++;
        }
```

```
if (vect2Temp[w].childNodes[0].type=="text"){
VTb[l]=vect2Temp[w].childNodes[0]; l++;
}
                }
               }
              }
             }
            }
           }
          }
         }
        }
```

MySql.

MySql es un Gestor de base de datos Open Source, el cual se ha elegido por su probada eficiencia y eficacia en numerosos proyectos existentes en Internet.

Patrón MVC.

Como bien se mencionó en el marco teórico de este proyecto, el patrón MVC permite la separación en capas de las aplicaciones web, separándola en vista, controlador, y modelo, el cual hace que el desarrollo de aplicaciones web se mas legible, al tener separado el diseño del modelo de negocio, permitiendo comunicarse el uno al otro mediante el controlador.

Si bien se ha elegido el lenguaje de programación JSP, por su seguridad y por la integración de Java mediante Servlets, hace que este sea el patrón a seguir con este tipo de lenguaje, convirtiéndose los Servlets en controlador y el desarrollar clases Java para el modelo, siendo este la comunicación con el servidor y dejando la vista a las estructuras HTML y JavaScript.

4.3 MODELO DE DESPLIEGUE.

Si bien se sabe, el proyecto a realizar es una aplicación web se hace uso del modelo cliente-servidor, el cual se ha analizado previamente en el marco teórico, teniendo detalles suficiente para el desarrollo del proyecto al considerar tecnologías de scripting en ambas partes.

En el nodo servidor Web se encontrara la aplicación conteniendo archivos JSP, los Servlets de Java y la base de datos, dicho nodo se conectara mediante un browser al nodo cliente en el cual se ejecutaran las hojas de estilos CSS y el script del lado del cliente JavaScript.

Las especificaciones técnicas de ambas partes serán comentados posteriormente en este trabajo al desarrollar la puesta en marcha.

4.4 INTERFAZ DE USUARIO GUI.

Para acercar esta documentación al desarrollo obtenido, se incluyen algunas capturas de pantalla de las interfaces principales diseñadas en el modelo de analisis y diseño, sin llegar a cubrir toda la implementación actual.

Realización de la encuesta (*vista del formato de AIEPI PARA NIÑOS DE 2 MESES A 5 AÑOS*)

Ilustración 21: Interfaz encuesta estrategia AIEPI

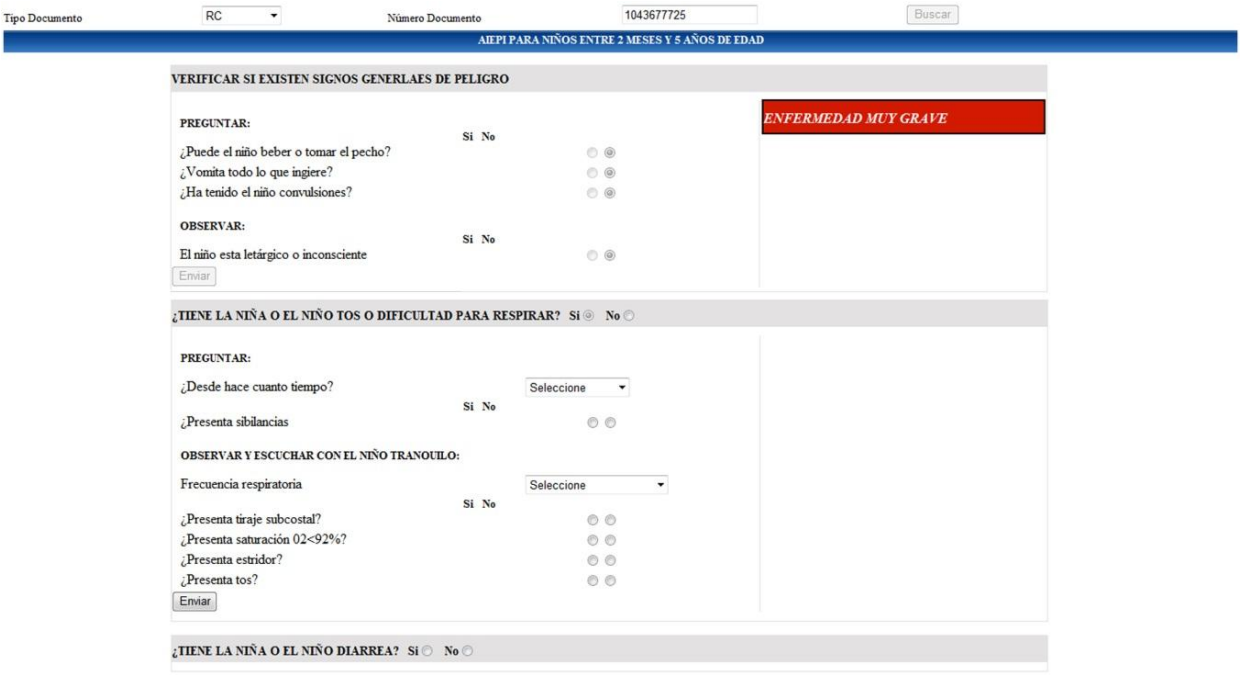


Ilustración 22: Interfaz encuesta estrategia AIEPI, desplegando area

CREACIÓN DE PREGUNTAS.

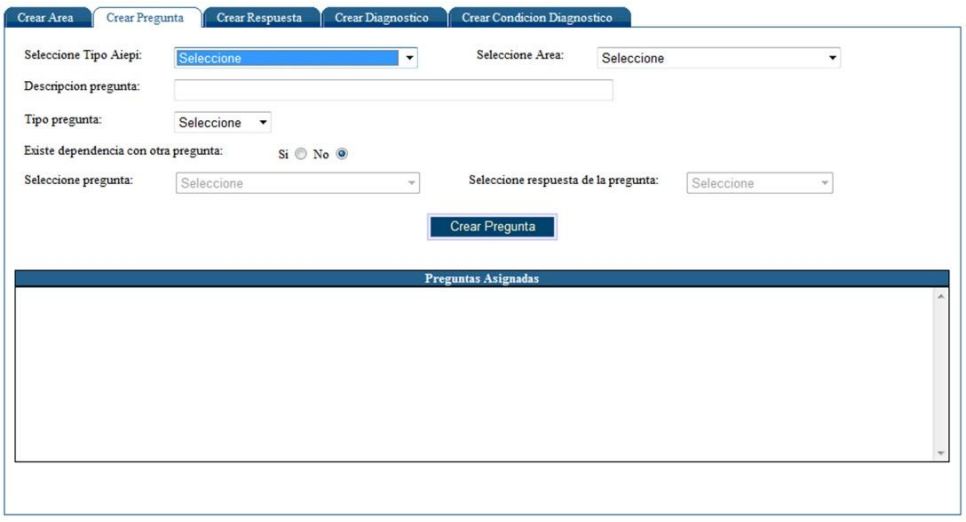


Ilustración 23: Interfaz creación formato AIEPI, creación de preguntas

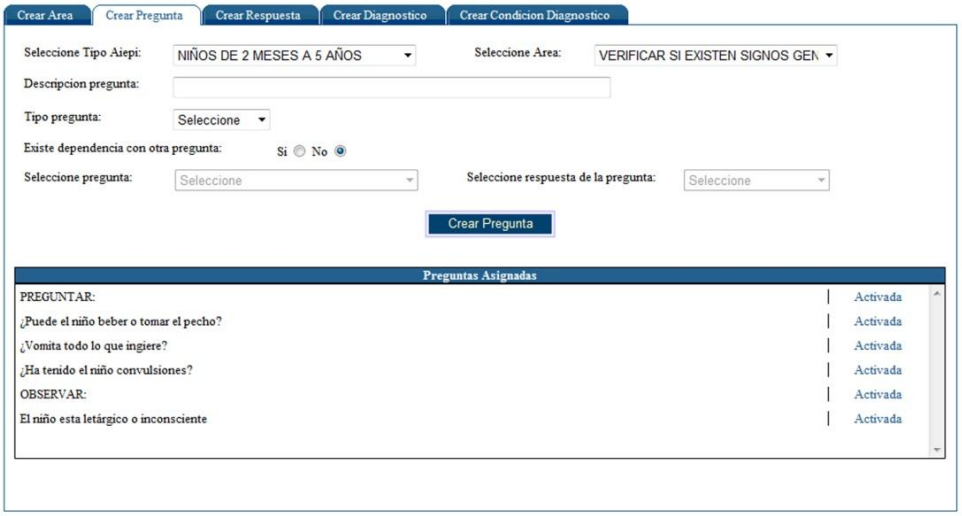


Ilustración 24: Interfazcreación formato AIEPI, creación de preguntas: desplegando preguntas realizadas

4.5 PUESTA EN MARCHA.

A continuación se especificaran las especificaciones técnicas requeridas para el funcionamiento del proyecto.

4.5.1 Tecnología Y Requerimientos Del Nodo Cliente.

Como bien se mencionó en el modelo de analisis y diseño, el módulo de sistematización y digitalización de la AIEPI al igual que el sistema de información SAIM, está diseñado exclusivamente para el uso del navegador IExplorer, para las versiones 7 en adelante, lo cual hace que este sistema de información y ahora el nuevo módulo desarrollado en este proyecto funcione bajo un sistema operativo Windows, preferiblemente en sus versiones de XP hacia adelante.

4.5.2 Tecnología Y Características Del Nodo Servidor.

Al ser desarrollado esta aplicación web bajo el lenguaje de programación JSP, hace que se necesite un servidor con las siguientes características.

- Apache Tomcat V6.0.
- Gestor de Base de datos MySql Server 5.0.
- Conexión de 2Mb dedicado.

4.5.3 Fusión Del Modulo De Digitalización Y Sistematización De La Estrategia AIEPI Al S.I. SAIM.

Se hicieron entrega de los archivos y clases correspondiente al ingeniero encargado del S.I. de la clínica para que los agregara al S.I. SAIM. Los archivos que se entregaron fueron los siguientes:

- Controlador: aiepi_Controlador -> ControlConfiguracion.java.
- Método: aiepi_metodo -> MetodoConfiguracion.java.
- Vista: aiepi.jsp, aiepi_creacion.jsp, aiepi_report.jsp, aiepi_reportes.jsp.
- Scripts: aiepi_configuracion.js, aiepi_encuesta.js.
- BD: BackupAIEPI.sql.

Con lo cual se lleva con éxito la fusión del modulo de *Digitalización y Sistematización de la estrategia para las enfermedades prevalentes de la infancia AIEPI.*

4.5.4 Capacitación De Los Usuarios.

Para la capacitación de los usuarios se ha de tener en cuenta los dos actores que interactuaran con el sistema, teniendo en cuenta que estos actores tienen unas funciones en común y otras que son propios de cada uno.

Para la cual se definió periodos de capacitación para funciones en común y periodos de capacitación para tareas individuales de los actores.

El proceso de capacitación esta desarrollado para culminarse en un máximo de 4 horas laborales, gracias a la interfaz amigable y fácil con la cual se ha desarrollado el S.I.

Las horas comprendidas quedan así:

- 1 hora para capacitación de funciones en común (Administrador, Usuario), *Reportes de encuestas AIEPI.*
- 1 hora para capacitación de las funciones individuales (Usuarios), *Realización encuesta AIEPI.*
- 2 horas para capacitación de las funciones individuales (Administrador), *Creación formato AIEPI, Reportes estadísticos AIEPI.*

4.6 CONCLUSIÓN.

Al llegar al final de la etapa de implementación y puesta en marcha, es posible considerar que se ha alcanzado el objetivo específico planteado:

- Implementación y puesta en marcha del sistema digital del AIEPI
- Fusionar el nuevo sistema con el sistema existente en la clínica
- Capacitar al personal de la clínica para el uso del sistema

Este son los últimos objetivos específicos que esta tesis persigue, por lo cual podemos dar por finalizado el desarrollo de las etapas de documentación del proyecto y concluyendo la implementación de software que debería ser la solución planteada al problema analizado en un primer momento.

5. CONCLUSIÓN.

A lo largo del desarrollo de esta tesis se demostró la utilidad de las técnicas y tecnologías de la programación web en el desarrollo de sistemas de información y en especial su uso en el desarrollo de HIS, por su gran facilidad de adaptación y facilidad de implementación. Para ello fue necesario hacer previamente una introducción detallada sobre cada una de las tecnologías y técnicas de manera tal que se pudiera construir una base solida de conocimientos sobre su estructura, alcance y limitaciones. De igual manera se hizo necesario estudiar y analizar como estaba conformada y en que consistía la estrategia AIEPI.

Posteriormente, se han considerado todas estas bases de conocimientos para llevar a cabo el desarrollo de una solución al problema planteado, construyendo así un modelo de analisis y diseño que será la base para la concreción del proyecto y poder obtener a cabalidad el cumplimiento del objetivo general del proyecto, logrando así la implementación del mismo.

Finalmente se logró cumplir con el objetivo general del proyecto desarrollando una aplicación en base a todos los requerimientos del sistema exigidos e implementando esta nueva aplicación como un modulo al S.I SAIM, como era uno de los requerimientos exigidos.

Por otra parte con el desarrollo de este proyecto y la realización de la aplicación se ha demostrado que la programación web es una de las mejores alternativas para el desarrollo de aplicaciones por su gran versatilidad e implementación y sobre todo por la gran ventaja de poder hacer uso de la aplicación en cualquier equipo sin tener la necesidad de instalarlo solo con tener internet.

BIBLIOGRAFÍA.

MUÑOS RODRIGUEZ, Pedro. Mantenimiento de portales de información. Madrid D.C.: Vision Libros, 2010. 214p. ISBN 978-84-9886-666-7.

LUJÁN MORA, Sergio. Programación de aplicaciones web: Historia, principios y clientes web. San Vicente. Editorial club universitario, 2008. 349p. ISBN 978-84-8454-206-3.

RÍOS AGUILAR, Sergio. Lenguajes HTML, JAVA y CGI: el diseño de páginas Web para Internet a su alcance. 7ed. Madrid D.C.: Abeto, 1996. 174p. ISBN 978-84-8955-427-6

LYNCH Patrick J., HORTON Sarah. Principios de diseño básicos para la creación de sitios web. México D.F.: Gustavo Gili, 2001. 164p. ISBN 978-96-8887-384-7

BATES, Chris. Web programming: building Internet applications. 7ed. California.Wiley, 2006.654p.ISBN 978-04-7001-775-3.

POWELL,Thomas A. Ajax: The Complete Reference. California. McGraw-Hill, 2008.654p. ISBN 978-00-7149-216-4.

PAHO, Organización panamericana de la salud. Enfermedades prevalentes de la infancia en las américas. Washington D.C.: OPS, 2008. 226p. ISBN 978-92-7532-901-6

PAHO, Organización Panamericana de la Salud, Ministerio de la Protección Social. 2ed. Libro clínico atención integradaa las enfermedadesprevalentes de la infancia. Bogotá D.C.: OPS, 2010. 572p. ISBN 978-95-8847-213-3.

	NORMAS PARA LA ENTREGA DE TESIS Y TRABAJOS DE GRADO A LA UNIDAD DE INFORMACION	VERSION: 01
		FECHA: Febrero 2011
		CODIGO:DOC-VACRE-NETGUDI

ANEXO 1
CARTA DE ENTREGA Y AUTORIZACIÓN DE LOS AUTORES PARA LA CONSULTA, LA REPRODUCCIÓN PARCIAL O TOTAL, Y PUBLICACIÓN ELECTRÓNICA DEL TEXTO COMPLETO DE TESIS Y TRABAJOS DE GRADO

Barranquilla, Fecha

Marque con una X
Tesis [X] **Trabajo de Grado** [ ]

Yo Harold Pangel Charris, identificado con C.C. No. 72.357.278, actuando en nombre propio y como autor de la tesis y/o trabajo de grado titulado Modulo de digitalizacion y sistematizacion de la atencion integrada a las enfermedades prevalentes de la infancia AIEPI presentado y aprobado en el año 2012 como requisito para optar al título de Ingeniero de Sistemas;
hago entrega del ejemplar respectivo y de sus anexos de ser el caso, en formato digital o electrónico (DVD) y autorizo a la CORPORACIÓN UNIVERSITARIA DE LA COSTA, para que en los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia, utilice y use en todas sus formas, los derechos patrimoniales de reproducción, comunicación pública, transformación y distribución (alquiler, préstamo público e importación) que me corresponden como creador de la obra objeto del presente documento.

Y autorizo a la Unidad de información, para que con fines académicos, muestre al mundo la producción intelectual de la Corporación Universitaria de la Costa, a través de la visibilidad de su contenido de la siguiente manera:

Los usuarios puedan consultar el contenido de este trabajo de grado en la página Web de la Facultad, de la Unidad de información, en el repositorio institucional y en las redes de información del país y del exterior, con las cuales tenga convenio la institución y Permita la consulta, la reproducción, a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato DVD o digital desde Internet, Intranet, etc., y en general para cualquier formato conocido o por conocer.

El AUTOR - ESTUDIANTES, manifiesta que la obra objeto de la presente autorización es original y la realizó sin violar o usurpar derechos de autor de terceros, por lo tanto la obra es de su exclusiva autoría y detenta la titularidad ante la misma. PARÁGRAFO: En caso de presentarse cualquier reclamación o acción por parte de un tercero en cuanto a los derechos de autor sobre la obra en cuestión, EL ESTUDIANTE - AUTOR, asumirá toda la responsabilidad, y saldrá en defensa de los derechos aquí autorizados; para todos los efectos, la Universidad actúa como un tercero de buena fe.

Para constancia se firma el presente documento en dos (02) ejemplares del mismo valor y tenor, en Barranquilla D.E.I.P., a los 17 días del mes de marzo de Dos Mil 200

EL AUTOR - ESTUDIANTE

FIRMA 72.357.278

ANEXO 2
F ORMULARIO DE LA DESCRIPCIÓN DE LA TESIS O DEL TRABAJO DE GRADO

TÍTULO COMPLETO DE LA TESIS O TRABAJO DE GRADO:

Modulo de digitalización y sistematización de la atención integrada a las enfermedades prevalentes de la infancia AIEPI

SUBTÍTULO, SI LO TIENE:

AUTOR AUTORES

Apellidos Completos	Nombres Completos
Rangel Charris	Harold

DIRECTOR (ES)

Apellidos Completos	Nombres Completos
Ariza Colpas	Paola

JURADO (S)

Apellidos Completos	Nombres Completos
De la Hoz Manotas	Alexis
Sepulveda Ojeda	Jorge

ASESOR (ES) O CODIRECTOR

Apellidos Completos	Nombres Completos
Hernandez Mercado	Jorge

TRABAJO PARA OPTAR AL TÍTULO DE: Ingeniero de Sistemas

FACULTAD: Ingeniería

PROGRAMA: Pregrado X Especialización ___

NOMBRE DEL PROGRAMA Ingeniería de Sistemas

	NORMAS PARA LA ENTREGA DE TESIS Y TRABAJOS DE GRADO A LA UNIDAD DE INFORMACION	VERSION: 01 FECHA: Febrero 2011 CODIGO:DOC-VACRE-NETGUDI

CIUDAD: Barranquilla **AÑO DE PRESENTACIÓN DEL TRABAJO DE GRADO:** 2012

NÚMERO DE PÁGINAS _____

TIPO DE ILUSTRACIONES:

- [x] Ilustraciones
- [ ] Láminas
- [ ] Retratos
- [x] Tablas, gráficos y diagramas
- [ ] Planos
- [ ] Mapas
- [ ] Fotografías

MATERIAL ANEXO (Vídeo, audio, multimedia o producción electrónica):

Duración del audiovisual: _____ minutos.

Número de casetes de vídeo: _____ Formato: VHS ___ Beta Max ___ ¾ ___ Beta Cam ____

Mini DV ____ DV Cam ____ DVC Pro ____ Vídeo 8 ____ Hi 8 ____

Otro. Cuál? _____

Sistema: Americano NTSC _____ Europeo PAL _____ SECAM _____

Número de casetes de audio: _____

Número de archivos dentro del DVD (En caso de incluirse un DVD diferente al trabajo de grado):

_____

PREMIO O DISTINCIÓN *(En caso de ser LAUREADAS o tener una mención especial)*:

_____

DESCRIPTORES O PALABRAS CLAVES EN ESPAÑOL E INGLÉS: Son los términos que definen los temas que identifican el contenido. (*En caso de duda para designar estos descriptores, se recomienda consultar con la Unidad de Procesos Técnicos de la Unidad de información en el correo biblioteca@cuc.edu.co, donde se les orientará).*

ESPAÑOL	INGLÉS
Desarrollo de Software	Software development
Aplicación web	Web app
Encuesta digital	

RESUMEN DEL CONTENIDO EN ESPAÑOL E INGLÉS:(Máximo 250 palabras-1530 caracteres):

Anexo 3:

Pagina web oficial de las diferentes tecnologías y estándares utilizados en la programación web

HTML: http://www.w3.org/community/webed/wiki/HTML

JavaScript: https://developer.mozilla.org/es/JavaScript

DOM: https://developer.mozilla.org/en/DOM

AJAX: https://developer.mozilla.org/en/AJAX

CSS: http://www.w3.org/Style/CSS/

PHP: http://www.php.net/

JSP: http://www.oracle.com/technetwork/java/javaee/jsp/index.html

ASP: http://www.asp.net/

RUBY ON RAILS:http://rubyonrails.org/

W3C: http://www.w3.org/

MySql: http://www.mysql.com/