



UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA EN COMPUTACIÓN

“Trabajo de grado previo a la obtención del Título de Ingeniero en

Computación”

TRABAJO DE GRADUACIÓN

**ANÁLISIS DE VULNERABILIDADES EN FRAMEWORKS DE CÓDIGO
ABIERTO PARA OPTIMIZAR LA GESTIÓN DE NOTIFICACIONES
ELECTRÓNICAS. CASO APLICATIVO: HISTORIAS CLÍNICAS EN EL
CONSULTORIO MÉDICO “LA DOLOROSA”**

Autor: Ligia Elena Ayala Torres

Director: Ing. Jorge Delgado

Riobamba – Ecuador

AÑO

2016

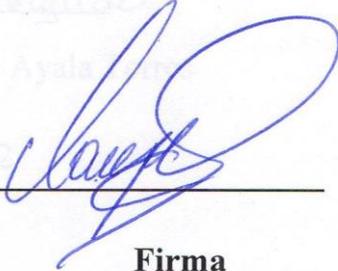
Los miembros del Tribunal de Graduación del proyecto de investigación de título: **Análisis de vulnerabilidades en frameworks de código abierto para optimizar la gestión de notificaciones electrónicas. Caso aplicativo: Historias clínicas en el consultorio médico “La Dolorosa”**, presentado por la Sra. Ligia Elena Ayala Torres y dirigido por el Sr. Ing. Jorge Delgado docente de la Universidad Nacional de Chimborazo.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

Ing. Danny Velasco

Presidente del Tribunal



Firma

Ing. Jorge Delgado

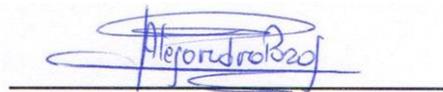
Director del Proyecto



Firma

Ing. Alejandra Pozo

Miembro del Tribunal



Firma

AUTORÍA DE LA INVESTIGACIÓN

“La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: Ligia Elena Ayala Torres autora y al Ing. Jorge Delgado Director del Proyecto; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.



Ligia Elena Ayala Torres

0603585522

AGRADECIMIENTO

Quiero agradecerles con todo mi corazón a Dios y a la Virgen María por escuchar mis oraciones y ayudarme a cumplir mi sueño.

A la Facultad de Ingeniería de la Universidad Nacional de Chimborazo de manera especial al ingeniero Jorge Delgado y a todo el personal Docente quienes a lo largo de mi carrera estudiantil me transmitieron sus sabios conocimientos y me apoyaron para poder culminar mis estudios.

A todas las personas que han formado parte de mi vida a las que me gustaría agradecer su amistad, consejos, apoyo, ánimo y compañía en los momentos más difíciles de mi carrera. De manera especial a la máster Marlene Burgos una persona luchadora, y bondadosa quien con su ejemplo me enseñó el valor de la vida, gracias por estar siempre a mi lado, por su paciencia, su apoyo, su confianza y su gran corazón.

Al Consultorio Médico La Dolorosa en la persona del doctor Fabián Arias Naranjo con sentimientos de profunda gratitud por su alto espíritu de colaboración y su calidad humana.

Al ingeniero Pablo Delgado por sus importantes aportes en la ejecución del presente proyecto.

DEDICATORIA

A Dios por darme la oportunidad de vivir y por estar conmigo en cada paso que doy, por su infinita bondad y amor. A la Virgen María por fortalecer mi corazón e iluminar mi camino.

A mi hijo Mateo, a quien amo con todo mi corazón y que es lo más hermoso que Dios me regaló.

A mi familia que de una u otra manera siempre estuvieron apoyándome cuando más los necesité y que son lo más valioso que la vida me dio.

A mis queridos amigos Mario y Adriana quienes estuvieron conmigo en los momentos más tristes de mi vida.

ÍNDICE GENERAL

AUTORÍA DE LA INVESTIGACIÓN	iii
AGRADECIMIENTO	iv
DEDICATORIA	v
ÍNDICE GENERAL.....	vi
ÍNDICE DE FIGURAS.....	x
ÍNDICE DE TABLAS	xi
RESUMEN.....	xii
SUMMARY	xiii
INTRODUCCIÓN	14
CAPÍTULO I.....	16
1. MARCO REFERENCIAL	16
1.1. Planteamiento del problema.....	16
1.2. Formulación del problema	17
1.3. Objetivos	17
1.3.1. Objetivo General	17
1.3.2. Objetivos Específicos.....	17
1.4. Justificación.....	18
CAPÍTULO II	20
2. FUNDAMENTACIÓN TEÓRICA.....	20
2.1. Antecedentes del tema.....	20
2.2. Frameworks de código abierto	21
2.2.1. Framework	21
2.2.2. Sistema de Gestión de Contenidos	21
2.2.2.1. DRUPAL.....	23
2.2.2.1.1. Características	23
2.2.2.1.2. Ventajas.....	25
2.2.2.1.3. Desventajas.....	25
2.2.2.2. JOOMLA.....	26
2.2.2.2.1. Características	27
2.2.2.2.2. Ventajas	27
2.2.2.2.3. Desventajas.....	28
2.2.2.3. WORDPRESS.....	29
2.2.2.3.1. Características	30

2.2.2.3.2.	Ventajas.....	31
2.2.2.3.3.	Desventajas.....	31
2.3.	Vulnerabilidades por exploits en frameworks de código abierto.....	32
2.3.1.	Vulnerabilidades.....	32
2.3.2.	Exploits.....	35
2.4.	Notificaciones Electrónicas.....	36
2.5.	Exploits para notificaciones electrónicas en frameworks de código abierto.....	38
2.6.	Análisis comparativo en frameworks de código abierto.....	42
2.7.	Comparación de frameworks de código abierto.....	43
CAPÍTULO III.....		45
3.	PROCESO DE INGENIERÍA DE SOFTWARE.....	45
3.1.	Ingeniería de la información.....	45
3.1.1.	Definición del Ámbito del Problema.....	45
3.1.2.	Investigar la organización.....	45
3.1.2.1.	Identificación de objetivos.....	45
3.1.2.2.	Identificación de estructura orgánico funcional.....	46
3.1.2.3.	Aplicar técnicas de investigación de campo.....	46
3.1.3.	Definición de Escenarios Problema.....	46
3.1.4.	Definición de alternativas de solución.....	47
3.2.	Estimaciones.....	47
3.3.	Planificación.....	48
3.4.	Análisis de Riesgos.....	48
3.4.1.	Identificar los Riesgos.....	49
3.4.2.	Gestión y Supervisión del Riesgo.....	49
3.5.	Factibilidad.....	52
3.5.1.	Factibilidad Operativa.....	52
3.5.2.	Factibilidad Técnica.....	52
3.5.3.	Factibilidad Económica.....	53
3.6.	Fase de Análisis.....	54
3.6.1.	Jerarquía de usuarios.....	54
3.7.	DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN.....	55
3.7.1.	Definición de la Metodología.....	55
3.8.	Descripción General del Sistema.....	55
3.9.	Definición del Prototipo.....	55
3.10.	Definición de Requisitos Generales del Sistema.....	56

3.10.1.	Requisitos funcionales relacionados con Médico	56
3.10.2.	Requisitos funcionales relacionados con Pacientes	56
3.10.3.	Requisitos funcionales relacionados con Turno médico	56
3.10.4.	Requisitos funcionales relacionados con Cita médica	57
3.11.	Requisitos de rendimiento	57
3.11.1.	Estáticos	57
3.12.	Atributos del sistema software	58
3.12.1.	Fiabilidad.....	58
3.12.2.	Disponibilidad	58
3.12.3.	Seguridad.....	58
3.12.4.	Portabilidad	58
3.13.	Requisitos no funcionales (Normas y Estándares)	58
3.14.	Requisitos no funcionales – Seguridad	59
3.15.	Requisitos no funcionales – Organización	59
3.16.	Catálogo de Usuarios	59
3.17.	Modelo de Negocio	59
3.18.	Objetivos y Alcance del Sistema.....	59
3.19.	Establecimiento de requisitos.....	60
3.19.1.	Especificación de Casos de Uso.....	60
3.20.	Casos de Uso	60
3.21.	Modelo Conceptual	68
3.22.	Secuencia de Actividades entre actores	68
3.23.	Definición de la Interfaz.....	70
3.24.	Diseño arquitectónico.....	71
3.25.	Diseño de la base de datos.....	72
3.26.	Diseño de procedimientos	73
3.26.1.	Registrar paciente	73
3.26.2.	Modificar paciente.....	79
3.26.3.	Eliminar paciente.....	82
3.27.	Pruebas de Software	85
CAPÍTULO IV		87
4.	RESULTADOS	87
4.1.	Comprobación de la Hipótesis	88
4.1.1.	Hipótesis.....	88
4.1.1.1.	Variable independiente.....	88
4.1.1.2.	Variable Dependiente	88

4.1.2.	Comprobación	88
4.1.2.1	Vulnerabilidades.....	88
4.1.2.2	Gestión notificaciones electrónicas	91
4.2.	CONCLUSIONES Y RECOMENDACIONES	93
4.2.1.	Conclusiones	93
4.2.2.	Recomendaciones.....	94
4.3.	BIBLIOGRAFÍA.....	95
4.4.	ANEXOS.....	99

ÍNDICE DE FIGURAS

Figura 1.	Diagrama de casos de uso	60
Figura 2.	Clases de la base de datos	68
Figura 3.	Secuencia de actividades entre actores y el sistema.....	69
Figura 4.	Interfaz para validación de usuarios.....	70
Figura 5.	Interfaz con menú principal.....	70
Figura 6.	Interfaz para eliminar un paciente	70
Figura 7.	Interfaz para registrar un nuevo paciente	71
Figura 8.	Interfaz para modificar un paciente.....	71
Figura 9.	Arquitectura del sistema.....	71
Figura 10.	Estructura de la base de datos	72
Figura 11.	Ejemplo de evaluación de un sitio con JoomlaScan	89
Figura 12.	Evaluación del sistema informático con JoomlaScan	90

ÍNDICE DE TABLAS

Tabla 1.	Análisis comparativo en frameworks de código abierto	42
Tabla 2.	Métricas para comparación de frameworks	43
Tabla 3.	Resumen comparación de frameworks	43
Tabla 4.	Planificación en tiempo del sistema	48
Tabla 5.	Documento de Riesgo 1	49
Tabla 6.	Documento de Riesgo 2	50
Tabla 7.	Documento de Riesgo 3	50
Tabla 8.	Documento de Riesgo 4	51
Tabla 9.	Documento de Riesgo 5	51
Tabla 10.	Caso de Uso 1.....	61
Tabla 11.	Caso de Uso 2.....	61
Tabla 12.	Caso de Uso 3.....	62
Tabla 13.	Caso de Uso 4.....	63
Tabla 14.	Caso de Uso 5.....	63
Tabla 15.	Caso de Uso 6.....	64
Tabla 16.	Caso de Uso 7.....	65
Tabla 17.	Caso de Uso 8.....	65
Tabla 18.	Caso de Uso 9.....	66
Tabla 19.	Caso de Uso 10.....	67
Tabla 20.	Caso de Uso 11.....	67
Tabla 21.	Resultados Pruebas Unitarias	85
Tabla 22.	Resultados Pruebas Funcionales	86
Tabla 23.	Resultados Vulnerabilidades JoomlaScan.....	90
Tabla 24.	Proceso para gestión notificaciones electrónicas	91
Tabla 25.	Proceso para servicios de atención médica	92

RESUMEN

En el presente proyecto, se investiga y analiza las vulnerabilidades en varios frameworks de código abierto, y la gestión de notificaciones electrónicas; mismos que se podrán percibir a través de la creación e implementación de un sistema informático para el Consultorio Médico La Dolorosa, ubicado en la ciudad de Riobamba.

Existen diferentes tipos de vulnerabilidades que ponen de manifiesto la falta de seguridad en diferentes frameworks, sin embargo muchos de éstos ya han realizado modificación a fin de contrarrestar estas deficiencias para proporcionar confiabilidad y seguridad en la información.

Para el desarrollo e implementación del sistema informático se seleccionó el framework Joomla como sistema de gestión de contenidos, Dreamweaver para el desarrollo de entorno web, PHP como lenguaje de programación, MariaDB como gestor de base de datos y Apache como servidor web.

Con la implementación de este sistema informático se automatizará el proceso para manejo de historias clínicas, así como la asignación de turnos médicos y reserva de citas médicas.

Mediante un software especializado en encontrar vulnerabilidades como es JoomlaScan se reporta vulnerabilidades potenciales, tanto del core del framework así como de plugins y componentes instalados; para asegurar que la información que se maneja en el sistema informático sea segura y confiable.

En conclusión se llegó a la obtención de un software de calidad, basado en un framework de código abierto que optimiza la gestión de notificaciones electrónicas, se cumplió con los objetivos propuestos y se cubrieron las expectativas y requerimientos de los usuarios que van a utilizar este sistema informático.



UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CENTRO DE IDIOMAS



M.Sc. Hernán Romero

29 de Febrero de 2016

SUMMARY

This project researches and analyzes the vulnerabilities in several open code frameworks, and management of electronic notifications which may be awarded through the creation and implementation of a computer system for “*La Dolorosa*” Doctor's Office, located in the city of Riobamba.

There are different types of vulnerabilities that highlight the lack of security in different frameworks, but many of these have already made modifications to counteract these deficiencies to provide reliability and information security.

For the development and implementation of the computer system the Joomla framework was selected as content management system, Dreamweaver for web development environment, PHP as programming language, MariaDB as manager of database and Apache as web server.

With the implementation of this computer system the process for managing medical records, as well as the allocation of medical shifts and booking of medical appointments.

Using a specialized software to find vulnerabilities as JoomlaScan it is reported potential vulnerabilities, both the core of the framework as well as plugins and installed components; in order to ensure that the information that is used in the computer system is safe and reliable.

In conclusion it was reached obtaining quality software, based on an open code framework optimizing the management of electronic notifications, the proposed objectives were achieved and the users' expectations and requirements were covered, users are going to use this computer system.



INTRODUCCIÓN

El mundo globalizado de hoy hace imprescindible el uso de la tecnología para el manejo de la información, sin embargo no se puede afirmar que esta información sea infranqueable, ya que todo software por más seguro que se diga que es, presenta vulnerabilidades en la codificación del mismo.

Sistemas operativos muy grandes y supuestamente seguros como Google ya han sufrido ataques por exploits, mediante un mensaje de texto han podido tener acceso a todo un dispositivo y obviamente a la información que éste contiene. Aplicaciones de mensajería también son afectadas por vulnerabilidades que a través de mensajes de texto cometen el fraude. La mensajería electrónica tampoco está a salvo, ya que a través de correo electrónico ficticio obtienen información para manejar listados de suscriptores, filtrado de contenidos e incluso archivo de mensajes.

El Consultorio Médico La Dolorosa, no dispone de un sistema informático para manejo de historias clínicas, es por eso que se ha visto la necesidad de implementar el Sistema para Historias Clínicas con la finalidad de automatizar la información de los pacientes que acuden a este centro de salud.

El presente trabajo investigativo consta de los siguientes capítulos:

En el Capítulo I el Marco Referencial describe la problemática que existe actualmente en el Consultorio Médico La Dolorosa para la gestión de historias clínicas, la delimitación, objetivos y justificación del proyecto de investigación.

En el Capítulo II se hace referencia al Marco Teórico, se analiza frameworks de código abierto, vulnerabilidades para la gestión de notificaciones electrónicas, servidor web, motor de base de datos, lenguaje de programación; necesarios para el desarrollo e implementación del sistema informático.

En el Capítulo III se realiza el proceso de ingeniería del software para determinar requerimientos, describir detalladamente el software, determinar el funcionamiento general junto con el diseño de los componentes del sistema.

En el Capítulo IV se analiza los resultados de los beneficios del sistema informático automatizado para historias clínicas, finaliza con las conclusiones y recomendaciones.

CAPÍTULO I

1. MARCO REFERENCIAL

1.1. Planteamiento del problema

Desde hace un tiempo que la integración de la sociedad con las nuevas tecnologías es completa y decir “Internet” no es ninguna palabra nueva e innovadora; ya no es un fenómeno nuevo, pero sin duda ha revolucionado el mundo tal y como se conocía hace 30 años. Es un fenómeno global, vinculado estrechamente con la comunicación, pero que influye en gran medida sobre casi todos los ámbitos de la sociedad.

Cada vez es más habitual el uso de aplicaciones y tecnologías web en empresas o Gobiernos. Su fácil implementación y uso han hecho que sean prácticamente omnipresentes y esenciales en el comercio electrónico y aplicaciones intranet o extranet; esto implica que información sensible y crítica es manejada a través de aplicaciones web.

La comunicación de hoy en día se orienta al uso casi necesario de dispositivos que permitan tener un canal de comunicación en el que casi todo el tiempo se debe estar informado de lo que pasa en ese preciso instante; ahora se puede ser partícipe de estos avances y recibir por ejemplo facturas de compras en los correos electrónicos, o recibir un mensaje de texto con un recordatorio para hacer algún trámite o para acudir a una cita médica, o también recibir una notificación en redes sociales como facebook e incluso información a través de mensajería instantánea como whatsapp; estos servicios están disponibles en internet a través de diferentes APIs (Interfaz de Programación de Aplicaciones) que son usadas para este propósito. Las notificaciones electrónicas necesitan una plataforma similar a la de una aplicación web, constituida por un servidor web, puede ser de igual manera un servidor de correos electrónicos y obviamente una infraestructura de red interna o abierta.

A pesar de los avances tecnológicos no siempre se puede garantizar en su totalidad la seguridad e integridad de datos que viajan a través de la red; son muchos los esfuerzos que se han realizado para evitar que terceras personas logren acceder a datos para sustraerse la información o simplemente para propagar software o código malicioso; lo más peligroso resulta cuando hackers logran a través de código aprovecharse de un agujero de seguridad (vulnerabilidad) en una aplicación o sistema, (exploits) para acceder a información almacenada en una base de datos y utilizan dichos datos para suplantar identidades, este es el ya conocido phishing; es por esto que resulta de gran importancia tener en cuenta un mecanismo con medidas necesarias para hacer que la información sea mantenida bajo un buen nivel de seguridad e integridad.

1.2. Formulación del problema

¿Cómo las vulnerabilidades en frameworks de código abierto inciden en la optimización para la gestión de notificaciones electrónicas. *Caso aplicativo:* historias clínicas en el consultorio médico “La Dolorosa”?

1.3. Objetivos

1.3.1. Objetivo General

Analizar las vulnerabilidades en frameworks de código abierto para optimizar la gestión de notificaciones electrónicas. *Caso aplicativo:* historias clínicas en el consultorio médico “La Dolorosa”.

1.3.2. Objetivos Específicos

- Analizar las vulnerabilidades por exploits en frameworks de código abierto.
- Realizar un estudio comparativo de exploits para notificaciones electrónicas en frameworks de código abierto.

- Desarrollar una aplicación web para historias clínicas en el Consultorio Médico La Dolorosa.

1.4. Justificación

Las TICs han hecho posible que se pueda hacer uso de las ventajas tecnológicas disponibles en la actualidad a través de internet; hoy en día se puede realizar compras en línea, inscripciones para eventos, tutorías o clases en línea a cualquier hora y desde cualquier lugar.

La sociedad actual se caracteriza por ser altamente tecnológica, es por eso que disponer de una aplicación web en internet se constituiría en una herramienta de mucha ayuda para agilizar el manejo de historias clínicas, para que los médicos puedan organizar su tiempo y dar una atención de calidad a sus pacientes.

Sin embargo no se puede pasar por desapercibido el tema de vulnerabilidades, mismas que se constituyen en las características o condiciones que presenta un sistema informático, conocido también como puntos débiles, que lo hacen idóneo a una acción que puede provocar un daño o afectar la información en lo referente a confidencialidad e integridad. Ante este escenario se hace necesario determinar estas debilidades para identificar riesgos y consecuentemente tomar las medidas necesarias para corregir estos puntos débiles.

Luego de un estudio minucioso se puede concluir que:

- El 75% de los ataques ocurren en las aplicaciones web.
- 3 de 4 servidores son vulnerables a los ataques web.
- En cada 1500 líneas de código hay una vulnerabilidad.
- Cada vez los datos personales tienen más valor, y hay más interesados en ellos.
- De 4396 vulnerabilidades, el 45% corresponde a aplicaciones web, y el 55% restante a otras vulnerabilidades.

- Entre las vulnerabilidades más explotadas se mencionan: SQL injection, Cross Site Scripting (XSS) y Cross Site REquest Forgery (CSRF).
- Por tipo de vulnerabilidad obtuvieron los siguientes resultados: 68% corresponde a Cross Site Scripting, 14% Information Leakage, 13% SQL injection, 3% Http Response Spliting, 2% Otros, 1% SSI injection.
- Entre las vulnerabilidades más comunes 85.57% corresponde a Cross Site Scripting, 15.70% Information Leakage, 26.38% SQL injection, 9.76% Http Response Spliting, 4.30% Otros.”¹

Al usar frameworks como plataforma para una aplicación web, se debe analizar y llegar a determinar el más idóneo para utilizarlo, en base a las características que presentan cada framework, especialmente en el tema de seguridad e integridad, ya que habrá un flujo de información valiosa que debe mantener los datos desde el origen hasta el momento mismo que se almacena en una base de datos.

¹ MARTORELLA CHRISTIAN, Principales vulnerabilidades en aplicaciones Web, extraído el 01 de Octubre del 2015 a las 08:00, disponible en la web <http://es.slideshare.net/Laramies/edge-security-rediris2008>

CAPÍTULO II

2. FUNDAMENTACIÓN TEÓRICA

2.1. Antecedentes del tema

En el Ecuador se utiliza en gran manera las notificaciones electrónicas, así por ejemplo se puede mencionar a la Dirección Nacional de Registro de Datos Públicos que maneja una herramienta tecnológica que hace posible que las entidades públicas que tienen que ver con aspectos como acciones coactivas y judiciales puedan enviar las notificaciones a las diferentes entidades para solicitar prohibiciones de vender o transferir por ejemplo. De igual manera se puede referir al Servicio de Rentas Internas, donde los usuarios reciben notificaciones electrónicas así como normas tributarias y aspectos relacionados con las declaraciones que deben realizar.

Existen varios proyectos enfocados al manejo de historias clínicas como: Registros médicos disponible en www.registrosmedicos.com, Historias clínicas online disponible en programahistoriasclinicas.com, o Sistemas clínicos disponible en www.sistemasclinicos.com; ninguno de estos proyectos trabaja con notificaciones electrónicas, sino únicamente se enfocan al manejo de datos de historias clínicas.

Los 5 principales tipos de vulnerabilidades en aplicaciones web son:

- Ejecución remota de código.
- SQL.
- Vulnerabilidades en formato de cadenas.
- Cross Site Scripting (XSS).
- Problemas atribuidos a los usuarios.²

² UNIVERSIDAD AUTONOMA DE MEXICO; Políticas y buenas prácticas de seguridad en servidores web del CDMIT, extraído el 01 de Octubre del 2015 a las 10:00, disponible en la web <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/915/A5.pdf?sequence=5>

Otro sitio (www.vsantivirus.com) publica que las vulnerabilidades más comunes son:

- SQL Injection.
- Cross Site Scripting (XSS).

El sitio del Departamento de Seguridad en Cómputo de la UNAM menciona en su lista de vulnerabilidades más comunes a las siguientes:

- Cross Site Scripting (XSS).
- SQL Injection.
- Buffer Overflow.³

2.2. Frameworks de código abierto

2.2.1. Framework

“En el desarrollo de software, un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto”. (Martínez C. , 2014)

2.2.2. Sistema de Gestión de Contenidos

CMS son las siglas de Content Management System, que se traduce directamente al español como Sistema Gestor de Contenidos. Como su propio nombre indica, es un sistema para gestionar contenidos. En general, un CMS permitiría administrar contenidos en un medio digital y para el caso particular que se necesita, un CMS permitiría gestionar los contenidos de una web.

³ DEPARTAMENTO DE SEGURIDAD EN CÓMPUTO DE LA UNAM; extraído el 01 de Octubre del 2015 a las 10:20, disponible en la web <http://www.seguridad.unam.mx/vulnerabilidadesDB/>

Dicho de otra forma, un CMS es una herramienta que permite a un editor crear, clasificar y publicar cualquier tipo de información en una página web. Generalmente los CMS trabajan contra una base de datos, de modo que el editor simplemente actualiza la base de datos, incluyendo nueva información o editando la existente.

En un periódico o cualquier otra página medianamente compleja. Principalmente aquellas que tienen que ser actualizadas diariamente o varias veces por día, donde además, las personas que editan la información no tienen conocimientos de informática. A estos redactores se les tiene que facilitar el trabajo mediante una herramienta que les permita subir información a la web y clasificarla para que aparezca en el lugar correcto. Por supuesto que estas personas no deben preocuparse por el código de la página ni las particularidades de programación de la plataforma donde esté alojada la web. Ellos sólo deben concentrarse en escribir las noticias, o cualquier tipo de contenidos y luego subirlas a la página con un sistema intuitivo y rápido. Una vez publicadas y clasificadas, las informaciones deben aparecer en la página web automáticamente, en los lugares donde haya decidido el editor.

Una herramienta CMS generalmente contendrá una interfaz basada en formularios, a los que habitualmente se accede con el navegador, donde se pueden dar de alta los contenidos fácilmente. Esos contenidos luego aparecerán en la página en los lugares donde se ha indicado al darlos de alta. Por lo tanto, un CMS estará compuesto de dos partes, un back y un front, siendo el back la parte donde los administradores publican las informaciones y el front la parte donde los visitantes visualizan las mismas.⁴

⁴ ALVAREZ MIGUEL; Qué es un CMS, extraído el 03 de octubre del 2015 a las 16:00 disponible en: <http://www.desarrolloweb.com/articulos/que-es-un-cms.html>

2.2.2.1. DRUPAL

Drupal es un CMS o sistema de gestión de contenidos que se utiliza para crear sitios web dinámicos y con gran variedad de funcionalidades.

Drupal es un software libre, escrito en PHP, que cuenta con una amplia y activa comunidad de usuarios y desarrolladores que colaboran conjuntamente en su mejora y ampliación. (Buitrago, 2011)

Esta ampliación es posible gracias a que se trata de un sistema modular con una arquitectura muy consistente, que permite que los módulos creados por cualquier desarrollador puedan interactuar con el núcleo del sistema y con los módulos creados por otros miembros de la comunidad. Con Drupal es posible implementar una gran variedad de sitios web: un blog personal o profesional, un portal corporativo, una tienda virtual, una red social o comunidad virtual, etc.

Es un programa de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web, y un énfasis especial en la usabilidad y consistencia de todo el sistema.

El diseño de Drupal es especialmente idóneo para construir y gestionar comunidades en Internet. No obstante, su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar muchos tipos diferentes de sitios web.⁵

2.2.2.1.1. Características

Entre las características más importantes se menciona algunas como:

- Ayuda on-line. Un robusto sistema de ayuda online y páginas de ayuda para los módulos del 'núcleo', tanto para usuarios como para administradores.

⁵ DRUPAL HISPANO; Sobre Drupal, extraído el 03 de octubre del 2015 a las 16:30 disponible en: <http://drupal.org/es/drupal>

- **Búsqueda.** Todo el contenido en Drupal es totalmente indexado en tiempo real y se puede consultar en cualquier momento.
- **Código abierto.** El código fuente de Drupal está libremente disponible bajo los términos de la licencia GNU/GPL. Al contrario que otros sistemas de 'blogs' o de gestión de contenido propietarios, es posible extender o adaptar Drupal según las necesidades.
- **Módulos.** La comunidad de Drupal ha contribuido muchos módulos que proporcionan funcionalidades como 'página de categorías', autenticación mediante jabber, mensajes privados, bookmarks, etc.
- **Personalización.** Un robusto entorno de personalización está implementado en el núcleo de Drupal. Tanto el contenido como la presentación pueden ser individualizados de acuerdo a las preferencias definidas por el usuario.
- **URLs amigables.** Drupal usa el mod_rewrite de Apache para crear URLs que son manejables por los usuarios y los motores de búsqueda.
- **Independencia de la base de datos.** Aunque la mayor parte de las instalaciones de Drupal utilizan MySQL, existen otras opciones. Drupal incorpora una 'capa de abstracción de base de datos' que actualmente está implementada y mantenida para MySQL y PostgreSQL, aunque permite incorporar fácilmente soporte para otras bases de datos.
- **Multiplataforma.** Drupal ha sido diseñado desde el principio para ser multiplataforma. Puede funcionar con Apache o Microsoft IIS como servidor web y en sistemas como Linux, BSD, Solaris, Windows y Mac OS X. Por otro lado, al estar implementado en PHP, es totalmente portable.
- **Múltiples idiomas y Localización.** Drupal está pensado para una audiencia internacional y proporciona opciones para crear un portal multilingüe. Todo el texto puede ser fácilmente traducido a través de una interfaz web, que importa traducciones existentes o integra otras herramientas de traducción como GNU ettext. (Gersbach, 2005).

2.2.2.1.2. Ventajas

También se puede mencionar que Drupal presenta varias ventajas como:

- Creación de sub-apartados en los foros.
- Método avanzado de clasificación de artículos.
- Acceso a la web es muy rápido.
- Compatibilidad con Jabber, Yahoo.
- Permite a los usuarios configurar su propio perfil en dependencia de los permisos otorgados por el administrador. (Pérez, 2007)

De igual manera, se presentan las siguientes características:

- Tiene una gran capacidad para almacenar, organizar y gestionar grandes volúmenes de contenido.
- Como los otros CMS de su competencia cuenta con módulos y añadidos con los que agregar cientos de funciones para hacer más completo este gestor y cumplir con las funciones que una web necesita.
- Tiene unas grandes posibilidades de desarrollo por lo que se puede adaptar a cualquier necesidad que se pueda tener ya sea para un desarrollo web profesional o sea para uno de carácter personal.
- Igual que el desarrollo se puede adaptar a cualquier necesidad, el diseño web también se puede adaptar a la perfección a las necesidades con este sistema.
- Este gestor de contenidos está pensado para poder aguantar webs de alto tráfico con lo cual sea cual sea el volumen que abarque la web se puede contar con este gestor. (PYME Networks, 2014)

2.2.2.1.3. Desventajas

Entre las desventajas se menciona que:

- Drupal se usa, entre otros, en intranets de compañías, enseñanza en línea, comunidades de arte y administración de proyectos.
- Es obligatorio cerrar sesión antes de salir del sitio.
- Los foros son un poco desorganizados.

- A pesar de contar con plugins y añadidos para este gestor, cuenta con muchos menos que sus competidores debido a su menor uso por parte de los usuarios finales.
- El panel de control de Drupal es muy complejo para un usuario de nivel bajo o medio con lo cual requiere una mayor dedicación para poder dominar el sistema respecto a cualquiera de sus competidores.
- Como cualquier otro desarrollo web de código libre, cualquier persona puede descargarlo y por lo tanto estudiar sus puntos débiles y hacerlos públicos para que cualquiera pueda atacar a una web con este sistema instalado.
- Aunque se puede adaptar cualquier diseño web al gestor, se debe disponer de un profesional con experiencia para minimizar los problemas de adaptación.
- Al tener una menor extensión de uso, la comunidad de usuarios y soporte para ayudar no es tan grande como en otros sistemas, además de ser una gran mayoría de habla inglesa y no de habla hispana lo cual puede ser un "handicap" para muchos usuarios no relacionados con esta lengua. (García, 2014)

2.2.2.2. JOOMLA

Es la herramienta líder en la creación de webs, es el Gestor de Contenidos (CMS en inglés) más premiado a nivel mundial, existen más de 30 millones de páginas web creadas con Joomla y presenta más de 10.000 componentes que permiten ampliar las funcionalidades de la web con nuevas opciones como pueden ser: tienda virtual, envío de boletines, foros, galerías de imágenes y un sinnúmero de posibilidades que no paran de crecer.

Permite gestionar con mucha facilidad toda una web, crear un nuevo apartado, modificar los actuales, añadir nuevas imágenes, crear nuevas opciones de menú y casi cualquier cosa que pueda necesitar el usuario, todo esto se puede hacer rápidamente y sin tener conocimientos técnicos, solamente conociendo Word o algún editor de textos se podrá manejar la web.

Es software libre (Open Source), se puede usar gratuitamente y el 95% de los componentes para Joomla también son gratuitos.⁶

2.2.2.2.1. Características

- Publicación de contenidos que se puede organizar por Categorías y Secciones, permite crear tantas páginas como se necesite y editarlas con un editor como si se trabajara en un procesador de texto.
- Se puede definir tantos menús y submenús como se necesite.
- Administración de imágenes y ficheros, permite subir las imágenes y ficheros que sean necesarios.
- Administración de usuarios para crear contenidos específicos u accesos a determinadas partes de la web solo para usuarios registrados.
- Encuestas que se pueden definir y mostrar en la web para que los usuarios voten y se tenga estadísticas.
- Diseño basado en plantillas que se puede modificar si se lo necesita para adaptarlas a las necesidades.
- Creación de módulos adicionales para poder colocar contenidos en determinadas partes de la web.
- Sindicación de noticias para poder publicar contenidos RSS automáticamente.
- Gestión de Banners que permitirán tener publicidad en la web.
- Instalación de Componentes adicionales que permitirán hacer crecer Joomla según las necesidades del usuario.⁷

2.2.2.2.2. Ventajas

- Cuenta con un sistema de Administración bastante organizado y potente.
- Tiene una gran biblioteca de extensiones.

⁶ WEBEMPRESA, ¿Qué es Joomla?, extraído el 04 de octubre del 2015 a las 08:00, disponible en: <http://www.webempresa.com/joomla.html>

⁷ WEB DESIGN, recuperado el 15 de 08 de 2015, de <http://www.webdesigncuba.net/ventajas-y-desventajas-de-joomla>

- Constantes Actualizaciones Gratuitas.
- Plantillas muy fáciles de usar sin necesidad de adaptación.
- A diferencia de otros gestores como puede ser Wordpress, Joomla no está tan limitado a ser un blog o una simple web de noticias, si no que se puede montar un portal completo con este sistema ya sea para hacer una tienda, un sitio de vídeos, un lugar de juegos online, un portal informativo etc.
- Existen multitud de plantillas gratuitas para descargar y no tener que realizar ningún diseño web.
- Al ser de código libre hay multitud de extensiones libres por Internet y totalmente gratuitas que pueden modificar el sistema para tener las características que se necesita en el desarrollo y añadir así todas las funcionalidades.
- Se puede configurar urls amigables y todo lo relacionado con SEO de manera sencilla como si de cualquier otro desarrollo web se tratara.⁸

2.2.2.2.3. Desventajas

- No tiene por defecto activado un plugin que muestre o habilite comentarios a los post o páginas dentro del sitio.
- No tiene un sistema de continuidad de los plugins, no se continúan para otras versiones.
- Se necesita programación mucho más compleja para lograr resultados satisfactorios.
- A fuerzas hay que declarar los recursos y algunos datos en el archivo xml.
- Las nuevas versiones son muy experimentales lo cual provoca conflictos con otros módulos de joomla, esto da un ejemplo de que joomla no cuenta con una organización en su comunidad.
- Como el resto de "Content Management System", al ser un sistema de código libre, cualquier persona puede descargarlo para estudiar sus vulnerabilidades y así atacar cualquier sitio con el instalado.

⁸ WEB DESIGN, recuperado el 15 de 08 de 2015, de <http://www.webdesigncuba.net/ventajas-y-desventajas-de-joomla>

- Es mucho menos intuitivo que otras plataformas como puede ser Wordpress con lo cual para un usuario "novel" puede ser un poco complejo.
- Su ventaja de distanciarse de la estética de un blog también puede ser una desventaja a la hora de tener un portal y querer tener un apartado con un blog en el mismo ya que Joomla dificultaría esta acción y se necesitaría recurrir a otros CMS externos.
- La creación y modificación de diseños es más compleja que en otros sistemas, sobre todo si no se tiene gran relación con este gestor previamente.
- La programación para este gestor también suele ser más difícil que la de cualquier otro desarrollo conocido.⁹

2.2.2.3. WORDPRESS

WordPress empezó en 2003 originalmente como una plataforma de blogging, pero con el tiempo ha ido evolucionando a un sistema de CMS (Content Management System) que funciona para crear prácticamente cualquier tipo de sitio Web.

Gracias a su flexibilidad y el hecho de que es un software de código abierto, se ha transformado en la herramienta más poderosa y fácil de utilizar para crear páginas o blogs. WordPress está disponible en su versión completa (WordPress.org) como un software descargable que se instala en un dominio con hospedaje propio. También está en una versión basada en la Web mucho más limitada. (Tagle, 2014)

WordPress es un sistema de gestión de contenidos o CMS enfocado a la creación de cualquier tipo de sitio, aunque ha alcanzado una gran relevancia en la creación de blogs (páginas web con una estructura cronológica que se actualiza regularmente). Ha sido desarrollado en el lenguaje PHP para entornos que ejecuten MySQL y Apache, bajo licencia GPL y es software libre. Su fundador es MattMullenweg. WordPress fue creado a partir del desaparecido b2/cafelog y se

⁹ WEB DESIGN, recuperado el 15 de 08 de 2015, de <http://www.webdesigncuba.net/ventajas-y-desventajas-de-joomla>

ha convertido en el CMS más popular de la blogosfera y es el más popular con respecto a cualquier otro CMS de uso general. Las causas de su enorme crecimiento son, entre otras, su licencia, su facilidad de uso y sus características como gestor de contenidos.

Otro motivo a considerar sobre su éxito y extensión es la enorme comunidad de desarrolladores y diseñadores, encargados de programarlo en su núcleo o crear complementos (llamados plugins) y plantillas (llamadas temas) para la comunidad. En febrero de 2015 era usado por el 23,4% de todos los sitios existentes en Internet.¹⁰

2.2.2.3.1. Características

- WordPress es gratis. Esta plataforma es utilizada por millones de bloggers y propietarios de sitios porque es gratis.
- Fácil de instalar y usar. Si se tiene un blog alojado en una compañía de hosting adecuada, se puede instalar WordPress con solo unos clics.
- Temas. WordPress tiene más de 10.000 temas libres, con temas Premium personalizados. Hay muchos otros sitios con temas gratuitos para WordPress que hacen que un sitio se vea muy bien, permite expresar la personalidad, y hacer que un sitio se destaque de otros sitios.
- Páginas. WordPress permite crear fácilmente ilimitadas páginas. Esto ayuda a convertir WordPress en un fantástico CMS (Content Management System). Se puede manejar fácilmente las páginas con plugins.
- Autores múltiples. Los Bloggers que les gusta que su blog esté en lugar y al día, pueden hacer un perfil de su blog y convertirse en autores invitados. Pueden publicar artículos y revisarlos para su publicación. Esto mantiene el contenido fresco e interesante.

¹⁰ CAPACITACIÓN Y FORMACIÓN. *C-FIP*. Recuperado el 15 de 08 de 2015, de http://www.c-fip.com.ar/?page_id=143

- Publicación programada. Se puede escribir entradas de blog anticipadamente y luego programarlas para ser publicadas en fechas específicas.¹¹

2.2.2.3.2. Ventajas

- Diseño.- Wordpress cuenta con miles de plantillas gratuitas y otras para comprar además de software específico para crear diseños desde cero como artisteer. En fin una amplia gama de diseño gracias a la cantidad de gente que lo usa en sus blogs.
- Plantillas.- Las plantillas son personalizables tanto en diseño como en la traducción de los textos, se puede crear una plantilla desde cero en muy pocos pasos.
- Seo.- En el tema de posicionamiento en búsquedas web es buenísimo además de contar con muchos plugins para aumentarlo aún más como puede ser el llamado All in One SEO Pack. Permite configurar el permalink para que las url sean amigables.
- Programación.- Su programación está hecha por expertos con muchas funcionalidades aunque no se pueda modificar el código fácilmente.
- Plugins.- Cuenta con miles de plugins de personalización y widgets desde un panel de inicio de sesión hasta chat para usuarios y lo mejor es que de cada uno hay varios para elegir según los gustos del usuario. (Gezal, 2015)

2.2.2.3.3. Desventajas

- Consumo de recursos.- Es un devorador de recursos por que hace muchas consultas a la base de datos y carga muchos script en cada página que se abre. En una página nueva con solo 2000 visitas o menos al mes puede consumir un ancho de banda de 5 GB.

¹¹ INTERNET MARKETING. Recuperado el 15 de 08 de 2015, de <http://michaelroca.com/2013/03/31/ventajas-y-caracteristicas-de-crear-un-blog-wordpress/>

- **Mantenimiento.-** Actualizaciones constantes de versión de software puede ser debido a fallos de seguridad o a mejoras del gestor la verdad es que nadie sabe con exactitud por qué sucede esto pero al fin y al cabo quiere decir que no está obsoleto y que hay personal activo detrás del mismo.
- **Modificar código.-** Es muy difícil modificar su código y es mejor no utilizarlo porque ya está más que programado y es casi imposible modificar sus funciones y códigos php etc. (Martínez J. , 2015)

2.3. Vulnerabilidades por exploits en frameworks de código abierto

2.3.1. Vulnerabilidades

Una vulnerabilidad informática es un elemento de un sistema informático que puede ser aprovechado por un atacante para violar la seguridad, así mismo pueden causar daños por sí mismos sin tratarse de un ataque intencionado. A las vulnerabilidades se les consideran un elemento interno del sistema, por lo que es tarea de los administradores y usuarios el detectarlos, valorarlos y reducirlos.¹²

La seguridad de la web debe ser prioritaria para aquellas organizaciones que usan internet como elemento primordial de comunicación con sus clientes o ciudadanos.

Asegurarse de una adecuada protección contra aquellos accesos no autorizados a los recursos de información, es esencial para la viabilidad de cada organización. Lo anterior debe ser incluido en la parte más alta de la lista de riesgos a los que hay que hacer frente, lo cual puede requerir de capacitación especializada para los auditores, profesionales de la seguridad y equipo de desarrollo. Lo más importante, es que estos grupos sean conscientes de todas las vulnerabilidades de

¹² UNAM, Tutorial de Seguridad e Informática, extraído el 02 de octubre de 2015 a las 10:00 disponible en: <http://redyseguridad.fi-p.unam.mx/proyectos/tsi/capi/Cap2.html>

las aplicaciones web, que incluyen la conocida y recientemente descubierta debilidad que puede ser explotada por los atacantes de internet.¹³

Hay tres reglas generales que deben ser continuamente reforzadas para minimizar los riesgos asociados a la utilización de la web para negocios y el uso privado:

1. Control de acceso a la información sensible: Desarrolladores de aplicaciones web no deben colocar información predecible o sensible en cualquier página web de acceso libre, dentro de un registro de Internet.
2. Establecer fuertes controles sobre la entrada: La regla más importante es nunca confiar en las transmisiones de datos entre el browser, el servidor web y los dispositivos de red. Siempre debe existir validación y revalidación en los controles de entrada.
3. Establecer pruebas de vulnerabilidad en el ciclo de vida del desarrollo de sistemas: La mayoría de las empresas de auditoría y consultoría de Tecnologías de la Información proveen económicas pruebas para testear la vulnerabilidad de las aplicaciones web. Estas pruebas permiten identificar debilidades en seguridad que pueden permitir el acceso a intrusos a la aplicación web y a las bases de datos. La incapacidad de identificar las vulnerabilidades en la Web a través de un testing estandarizado puede generar un impacto significativo en el proceso de solicitud del cliente.

Entre las vulnerabilidades más comunes se encuentran las siguientes:

- Cross-site scripting: Esto ocurre cuando la aplicación web toma los datos suministrados por el usuario y los envía al browser sin una validación o codificación previa del contenido. En consecuencia, las vulnerabilidades XSS* permiten a los atacantes ejecutar un programa script en el browser de la víctima. Además, este riesgo es considerado como TOP TEN dentro de las debilidades de las aplicaciones web.

¹³ DAS-CHILE, Vulnerabilidad, extraído el 02 de octubre del 2015 a las 10:20 disponible en: <http://www.bscconsultores.cl/descargas/B.2%20Vulnerabilidad.pdf>

- **Inyección SQL:** Esta vulnerabilidad ampliamente publicitada, ocurre al nivel de la base de datos de una aplicación. Es un ataque vía web, que aprovecha errores en la filtración de datos introducidos por el usuario y que permiten a un atacante tener el control de cierta aplicación.
- **Insecure Direct Object Reference:** Una referencia directa a un objeto ocurre cuando un desarrollador expone una reseña a un objeto de implementación interno como un archivo, directorio, registro de una base de datos, clave como una URL o un parámetro formal. Los atacantes pueden manipular estas referencias para acceder a otros objetos sin autorización.
- **Cross-site request forgery (Falsificación en sitios cruzados):** CSRF o también conocido como XSRF es una clase de ataque que afecta a las aplicaciones web con una estructura de invocación predecible. Existen aplicaciones que usan cookies, autenticación de navegador o certificados de cliente. La idea básica de XSRF es un simple atacante que engaña de alguna manera al usuario para que realice una acción determinada en la aplicación objetivo / vulnerable sin que el usuario tenga conocimiento de los hechos que están ocurriendo realmente.
- **Manejo incorrecto de errores:** Algunas aplicaciones pueden filtrar involuntariamente información sobre su configuración y funcionamiento. De esta manera, las aplicaciones web a menudo otorgan datos acerca de su estado interno por medio de mensajes de error detallados o de depuración. Muchas veces esta información puede ser aprovechada para poner en marcha o incluso automatizar los ataques más poderosos.
- **Error para restringir un acceso URL:** El método de ataque para explotar una vulnerabilidad puede ser muy simplista. Este incluye enlaces y el uso de técnicas de fuerza bruta para encontrar páginas desprotegidas. Vulnerabilidades específicas incluyen acceso y explotación de la información sensible, URLs ocultos y especiales, artículos y códigos de seguridad que evalúan los privilegios en el cliente. Como resultado, los atacantes pueden obtener acceso a información confidencial, de control de seguridad en el cliente para que el navegador y las aplicaciones eludan los controles integrados en el código que se envía al browser.

2.3.2. Exploits

Exploit (proviene del inglés to exploit, explotar, aprovechar) es el nombre con el que se identifica un programa informático malicioso, o parte del programa, que trata de forzar alguna deficiencia o vulnerabilidad (bug) del sistema. El fin de este puede ser la destrucción o inhabilitación del sistema atacado, aunque normalmente se trata de violar las medidas de seguridad para poder acceder al mismo de forma no autorizada y emplearlo en beneficio propio o como origen de otros ataques a terceros.¹⁴

Dado que no se puede establecer una regla general porque cada uno ataca un problema diferente es una buena costumbre leer el código fuente para saber cómo compilarlo y ejecutarlo con éxito. Además a menudo vienen con algún texto explicativo tipo léeme o similar.

nano exploit.c

pico exploit.c

more exploit.c

cat exploit.c

etc.

En un entorno unix/linux lo normal será bajar el código fuente y luego compilarlo. Para ello se sigue el procedimiento habitual de la compilación de un programa en unix/Linux. Esto es: gcc exploit.c -o exploit y luego para ejecutarlo: ./exploit parámetro1 parámetro2 ... parámetro n

Pero si el exploit estaba en perl

perl exploit.pl parámetro1 parámetro 2 ... parámetro n

Si el exploit está en shell script

./sh exploit.sh parámetro1 parámetro 2 ... parámetro n

¹⁴ SeguridadPC.net, extraído el 02 de octubre de 2015 a las 12:00 disponible en <http://www.seguridadpc.net/exploit.htm>

Si desea compilarlo en windows deberá tener la herramienta apropiada para manejarlo en windows. Si está en C Dev c/c++ o LCCWIN 32 por ejemplo. Si está en perl necesitarás un emulador de perl para windows.¹⁵

2.4. Notificaciones Electrónicas

Las notificaciones electrónicas son aquellas comunicaciones que emite la administración pública y privada mediante la utilización de medios electrónicos y telemáticos, tales como el Internet y el correo electrónico. En el campo de la Administración de Justicia, surgen como una alternativa inmediata para lograr que los procesos judiciales que utilicen este medio se desarrollen con una mayor celeridad, economía y seguridad procesal.

Las notificaciones electrónicas forman parte de la Informática Jurídica de Gestión aplicado al campo de la Administración de Justicia Pública, y se puede decir que forman parte del llamado gobierno electrónico, el cual es definido como la "realización de una serie de actividades que cumple actualmente el Estado moderno, como administración de un determinado país, valiéndose para ello de los nuevos recursos tecnológicos y específicamente, de los que ofrece la red de Internet.¹⁶

A través de las notificaciones electrónicas aplicadas al campo de la administración de justicia, los litigantes de un proceso podrán enterarse del contenido de las resoluciones judiciales, desde la comodidad de su hogar, oficina o desde una cabina pública de Internet, sin necesidad de desplazarse a las sedes o domicilios procesales y sin la obligación de comprar cédulas de notificación; es decir se ahorra tiempo y dinero.

¹⁵ ELHACKER.NET, Exploits, extraído el 02 de octubre de 2015 a las 15:00, disponible en <https://www.elhacker.net/exploits.html>

¹⁶ FERNANDEZ ITALO; "El Gobierno en la Era Digital, E-government", disponible en: <http://www.alfa-redi.org/revista/data/26-10.asp>

Las Notificaciones Electrónicas se realizan generalmente vía Internet, ya sea directamente a través de una página web o por correo electrónico.

a) Notificaciones a través de una página Web

Consisten en aquellas notificaciones realizadas poniendo a disposición de los usuarios, a través de una página web en Internet, las resoluciones que emite una determinada entidad. Sin embargo este sistema no ofrece una debida confidencialidad, pues cualquier usuario que ingresa a la página web de dicha entidad, puede enterarse del contenido de las notificaciones publicadas en dicha página web.

En Ecuador, se realiza este tipo de notificaciones por entidades tales como la Superintendencia Nacional de Administración Tributaria (SUNAT), la Superintendencia Nacional de Aduanas, ESSALUD, entre otras, amparándose en lo dispuesto en el Art. 104 inciso e, del Texto Único Ordenado del Código Tributario.

b) Notificaciones realizadas a través de Correo Electrónico

La notificación por correo electrónico es aquella comunicación dirigida a los domicilios o direcciones electrónicas de los usuarios. Estas direcciones o casillas electrónicas son las direcciones electrónicas procesales de las partes y "constituye la residencia habitual, en la red de Internet, de la persona"¹⁷.

Al respecto cabe aclarar que cuando se envía un mensaje a una dirección de correo electrónico, como por ejemplo usuario@law.com, lo que se está haciendo en realidad es enviar dicha información a un servidor, en este caso al servidor de law.com, donde el mensaje es almacenado. Los usuarios posteriormente tienen acceso, desde Internet, a la lectura de sus mensajes, sólo cuando hayan ingresado un nombre de usuario (login) y una contraseña (login) determinada.

¹⁷ NUÑEZ PONCE, Julio; "Implicancias Jurídicas de la Notificación Enviada por Medios Informáticos y el Domicilio Virtual" , artículo publicado en la Revista Electrónica de Derecho Informático, disponible en <http://www.alfaredi.org/revista/data/13-5.asp>

Para realizar las notificaciones por correo electrónico el Poder Judicial debería ampliar su servicio de correos electrónicos a los abogados y/o litigantes, otorgándoles una cuenta, dirección o casilla electrónica en sus servidores de tipo “pj.gob.pe”. Estas direcciones electrónicas actuarían como el domicilio procesal, lugar donde se les debe hacer llegar sus notificaciones a los litigantes de un proceso.”

2.5. Exploits para notificaciones electrónicas en frameworks de código abierto

Un framework proporciona toda una estructura global que sirve de base para el desarrollo e implementación de software, sin embargo se ha llegado a determinar que no existe una aplicación totalmente segura y operable y siempre va a existir muchas o pocas vulnerabilidades que hacen que el framework sea inestable y por consiguiente proporcione inseguridades en el manejo de la información.

Entre las principales vulnerabilidades encontradas en frameworks de código abierto se menciona:

- **Sitio web comprometido**

Esto significa que un sitio web realizado con un CMS puede ser víctima de un hacker convirtiéndolo en un sitio web hackeado, lo que permite a los hackers un mal uso de un dominio de confianza. La intención de estos hackers será instalar un enlace en la base de datos sin el conocimiento del propietario, así cuando alguien accede al sitio web es redireccionado a otro dominio. Un ejemplo muy común son los spammers que pueden llegar a manipular un sitio web con el fin de redirigir a los usuarios a una URL donde se venden cualquier tipo de productos que nada tienen que ver con el website. De acuerdo con una encuesta realizada por StopBadware y Commtouch, aquí se presenta un desglose de los sitios comprometidos basados en el CMS que utilizan:

- WordPress 28%
- Joomla 9%
- Drupal 1%
- Otros CMS 62%

- **SQL Injection**

En este tipo de ataque un código malicioso es insertado en el website, principalmente en campos de un formulario, y es usado como puerta de entrada para explorar las vulnerabilidades de la base de datos. Un ejemplo de esta vulneración se produce en los casos en los que un hacker introduce una sentencia SQL maliciosa, normalmente en un campo de un formulario, capaz de descifrar usuarios y contraseñas de acceso al CMS, eliminar o modificar la estructura de alguna tabla o incluso eliminar por completo la base de datos que mantiene el sistema CMS.

Algunos de los métodos de inyección de código utilizados en los ataques, son los siguientes:

1.- Inyección en un formulario: Se trata del ataque más sencillo. Consiste en inyectar código en un formulario que después al enviarlo al servidor, será incluido en el código fuente de alguna página. Una vez insertado en el código fuente, cada vez que se cargue la página se ejecutará el código insertado en ella.

2.- Inyección por medio de elementos: En este tipo de sistema de inyección de código se utiliza cualquier elemento que viaje entre el navegador y la aplicación, como pueden ser los atributos usados en las etiquetas HTML utilizadas en el diseño de la página.

3.- Inyección por medio de recursos: Aparte de los elementos en la url y los formularios, hay otras formas en las que se puede actuar como son las cabeceras

HTTP. Estas cabeceras son mensajes con los que se comunican el navegador y el servidor. Aquí entran en juego las cookies, las sesiones, campo referer.¹⁸

- **Vulnerabilidades de código**

Los sitios web definidos mediante WordPress, Drupal o Joomla son construidos utilizando los conocimientos de programación PHP. Muchas veces, si no se codifica correctamente, un hacker puede encontrar vulnerabilidades en el código. Algunos ejemplos de estas vulnerabilidades de código pueden ser:

- Envío de parámetros sin validar o sin codificar.
- Errores en el control de acceso
- Mal uso de gestión de Sesiones
- Desbordamiento de búfer
- Comandos equivocados o con un mal funcionamiento.

- **Contraseñas fáciles de descifrar**

La facilidad de descifrar las contraseñas es una vulnerabilidad de seguridad muy común. Por ejemplo, un miembro del equipo interno o un ex empleado de la empresa pueden descubrir la contraseña fácilmente, iniciar sesión en la base de datos y robar información valiosa. Por lo tanto, se aconseja cambiar las contraseñas regularmente y utilizar contraseñas complejas difíciles de hackear.

Para proteger los datos guardados en una base de datos que mantiene un website definido mediante un CMS es bueno controlar constantemente la base de datos para garantizar que no se producen actividades inusuales o que puedan corromper el sistema. Para ello los administradores de los website deberían tomar las siguientes medidas:

- Verificar que sólo los usuarios autorizados tienen acceso a la base de datos, asegurarse de que todos los usuarios que tienen acceso a la base de datos no

¹⁸ CARDENAL GARDOKI (2015), Qué es y cómo funciona un ataque Cross-Site Scripting. Recuperado desde http://pressroom.hostalia.com/wp-content/themes/hostalia_pressroom/images/cross-site-scripting-wp-hostalia.pdf

son administradores y asignar privilegios a los usuarios en función de sus roles.

- Vigilar los cambios que tienen lugar en su sitio web. La instalación de un nuevo módulo o plugin puede exponer a ciertas vulnerabilidades a la base de datos que mantiene el CMS debido a que en los sistemas de gestión de contenidos toda la información se almacena en la propia base de datos.
- Instalar un sistema de seguimiento de datos del sistema de seguridad que continuamente audite la base de datos en tiempo real y ofrezca informes sobre los cambios de la base de datos para detectar y registrar posibles inyecciones SQL.
- Elegir un proveedor de hosting consistente y seguro que ofrezca servidores optimizados para WordPress, Joomla y Drupal. Es aconsejable disponer de un servicio de alojamiento donde se pueda controlar todos los parches de seguridad, aplicaciones, actualizaciones de software, etc.

- **Cross Site Scripting (XSS)**

El XSS es una vulnerabilidad que suele aparecer por la falta de mecanismos en el filtrado de los campos de entrada que dispone la web, y permite el envío de datos e incluso la ejecución de scripts completos. El código malicioso utilizado en este tipo de ataques está compuesto por cadenas de datos: scripts completos, contenidos en enlaces o ejecutados desde formularios vulnerables.¹⁹

El XSS reflejado o indirecto es el tipo de ataque XSS más habitual y consiste en editar los valores que se pasan mediante URL, cambia el tipo de dato pasado del usuario a la web, haciendo que ese código insertado se ejecute en dicho sitio.²⁰

¹⁹ CARDENAL GARDOKI (2015). Qué es y cómo funciona un ataque Cross-Site Scripting. Recuperado desde http://pressroom.hostalia.com/wp-content/themes/hostalia_pressroom/images/cross-site-scripting-wp-hostalia.pdf

²⁰ PARRA VIOETA, Vulnerabilidad CMS, extraído el 05 de octubre de 2015 a las 16:00, disponible en <http://www.seas.es/blog/informatica/vulnerabilidad-cms/>

2.6. Análisis comparativo en frameworks de código abierto

Para desarrollar una aplicación web de calidad, se ha creído conveniente hacer énfasis en dos criterios:

- a. Desarrollo: Es preciso contar con herramientas así como un lenguaje de programación para poder desarrollar un software de calidad; es por eso que el framework debe ser compatible con herramientas de código abierto como PHP, MYSQL, PostgreSQL, MariaDB, Apache.
- b. Rendimiento: Este criterio se presta para muchas controversias aún sin tomar en cuenta los resultados que se obtienen; es muy importante que los frameworks resuelvan los problemas que presentan en cuanto a seguridades en las capas de presentación, lógica y datos.

Tabla 1. Análisis comparativo en frameworks de código abierto

Framework \ Criterio	Desarrollo	Rendimiento
Drupal	PHP, MYSQL, PostgreSQL, Apache	Presenta algunas vulnerabilidades, pocos módulos y plugins.
Joomla	PHP, MYSQL, MariaDB, PostgreSQL, Apache	Mínimas vulnerabilidades, con gran variedad de módulos, componentes y plugins, anti SPAM, protecciones de acceso a la administración, contra intrusiones y hackeos; niveles de cifrado de la contraseña, sistema de seguridad de doble factor. Consume muy pocos recursos.
Wordpress	PHP, MYSQL, Apache	Pocas vulnerabilidades, gran cantidad de módulos, plugins, sin embargo consume muchos recursos, dificultad para modificar su código.

2.7. Comparación de frameworks de código abierto.

Se proponen métricas intuitivas para determinar aspectos de seguridad y uso de recursos; para esto se necesita estipular una escala para poder escoger un framework en base a sus bondades y deficiencias como se detalla a continuación:

- Seguridad XSS: 1= Bajo, 3= Medio, 5= Alto.
- Seguridad SQL Injection: 1= Bajo, 3= Medio, 5= Alto.
- Plugins, módulos, componentes: 1= Pocos, 3= Algunos, 5= Muchos.
- Facilidad de uso: 1= Bajo, 3= Medio, 5= Alto.
- Sitio web comprometido: 1= Altamente, 3= Medianamente, 5= Poco.

Tabla 2. Métricas para comparación de frameworks

Frameworks \ Criterios	Seguridad XSS	Seguridad SQL Injection	Plugins, módulos, componentes	Facilidad de uso	Sitio web comprometido
Drupal	1	1	1	1	3
Joomla	5	5	5	3	5
Wordpress	3	3	3	3	3

Una vez realizada la evaluación y mediante las métricas mencionadas anteriormente, se procede a realizar el cálculo con los datos de la Tabla 2 para poder tener una idea clara y precisa del framework que se ajusta a los requerimientos del sistema informático.

Tabla 3. Resumen comparación de frameworks

FRAMEWORK	SUMATORIA
Drupal	7
Joomla	23
Wordpress	15

Para establecer el framework que se ajuste a los requerimientos del sistema informático planteado en cuanto a la seguridad y rendimiento, se basará en la Tabla 2 y Tabla 3.

Se observa que el framework Joomla obtuvo mayor puntaje, por lo que se puede aseverar que proporciona un nivel de seguridad alto contra vulnerabilidades por XSS, así como de SQL injection; proporciona un buen número de módulos, plugins y componentes para el desarrollo de aplicaciones web y no requiere de un conocimiento avanzado para poder manipular su código o poder acoplarse con un sistema informático para ambiente web.

Por lo anteriormente mencionado se utilizará el framework Joomla.

CAPÍTULO III

3. PROCESO DE INGENIERÍA DE SOFTWARE

3.1. Ingeniería de la información

3.1.1. Definición del Ámbito del Problema

En esta etapa se constituye en un aspecto sustancial establecer cuáles son las necesidades de los usuarios. Al finalizar esta fase se obtendrá los aspectos necesarios para determinar los requisitos.

En la actualidad en el Consultorio Médico La Dolorosa, las historias clínicas se las realiza de manera deficiente, lo que ocasiona una atención deficiente en el manejo de la información a los pacientes que acuden a este centro de salud.

3.1.2. Investigar la organización

3.1.2.1. Identificación de objetivos

Los objetivos del Consultorio Médico La Dolorosa son los siguientes:

- Ofertar servicios médicos de calidad con profesionales que consigan un alto grado de prestigio y satisfacción entre la población que confía en este centro médico.
- Utilizar tecnología de punta para el diagnóstico y tratamiento en el ámbito de la medicina primaria tanto preventiva como asistencial.
- Incrementar la atención personalizada a pacientes y organizaciones públicas y privadas.
- Contar con un equipo de profesionales de prestigio con auténtica vocación de servicio al paciente.

3.1.2.2. Identificación de estructura orgánico funcional

Es necesario tener muy claro cómo está organizado funcionalmente el Consultorio médico La Dolorosa ya que de esta manera se puede respaldar la solución software.

3.1.2.3. Aplicar técnicas de investigación de campo

Para el producto final o software a desarrollar es necesario que exista una idea diáfana de lo que éste debe realizar (tanto el cliente que en este caso es el Consultorio Médico La Dolorosa y la desarrolladora del sistema informático), en tal virtud se han efectuado reuniones con el Galeno del Consultorio Médico La Dolorosa.

3.1.3. Definición de Escenarios Problema

Resulta muy importante clasificar y categorizar los problemas del Consultorio médico La Dolorosa para poder entenderlos de manera clara para una mejor definición.

Los escenarios que se han detectado se detallan a continuación:

- Las citas médicas se las realiza de forma personal acudiendo al consultorio médico o por teléfono, se separa la cita para una fecha y hora determinada.
- Las historias clínicas se las lleva manualmente en un formato adaptado por el médico tratante.
- Las recetas son emitidas por los doctores a sus pacientes escritas a mano, ilegibles en la mayoría de veces o incluso incomprensibles para los pacientes.
- La búsqueda de información es tediosa por cuanto no está disponible en el instante que se necesita, y hay que esperar un tiempo estimado entre 5 a 10 minutos para acceder a ella; esto implica que se desperdicia tiempo y recursos. Sin embargo hay situaciones en las que no se encuentra la información requerida, por cuanto está extraviada.

3.1.4. Definición de alternativas de solución

En este punto es imprescindible detallar toda una serie de procedimientos con el propósito de dar solución a los problemas que se han detallado en el punto anterior; por lo que se desarrollará una solución informática para automatizar el proceso de historias clínicas.

- El módulo de Pacientes, permite ingresar, eliminar y realizar búsquedas de información detallada de pacientes, y reservar citas con médicos.
- El módulo de Médicos, permite manejar información de los médicos que laboran en el Consultorio Médico La Dolorosa, así como el proceso para asignar turnos médicos.
- Toda la información ingresada puede ser manejada de forma segura y en cualquier momento que sea requerida, proporcionando de esta manera menos consumo de recursos tanto humanos como materiales.

3.2. Estimaciones

Mediante las estimaciones de proyectos generales, se desarrollará el sistema informático con el modelo COCOMO como técnica para las estimaciones. Se considerará el número de instrucciones de líneas de código (KLDC) para proyectos estándares; el presente proyecto tiene 2.5 KLDC, seguidamente se deberá deducir: esfuerzo (E hombres-mes), tiempo de desarrollo de software (D meses), productividad (P KLDC / hombres-mes), cantidad de hombres (CH hombres).

$$E = 2.4 (\text{KLDC})^{1.05}$$

$$D = 2.5 (E)^{0.38}$$

$$E = 2.4 (2.5)^{1.05}$$

$$D = 2.5 (6.3)^{0.38}$$

$$E = 6.3 \text{ hombres - mes}$$

$$D = 5.03 \text{ meses} \approx 5 \text{ meses}$$

$$CH = \frac{E}{D}$$

$$CH = \frac{6.3}{5.03}$$

$$CH = 1.2 \text{ hombres} \approx 1 \text{ hombre}$$

$$P = \frac{(KLDC * 1000)}{E}$$

$$P = \frac{(2.5 * 1000)}{6.3}$$

$$P = \frac{396.8 \text{ LDC}}{\text{hom}br\text{es} - \text{mes}}$$

3.3. Planificación

A través de un plan de tareas se ha determinado que el proyecto tendrá una duración de 120 días equivalente a 4 meses, para esto han intervenido varios recursos como: Usuarios, Programador, Planificador, Analista, Entrevistador, Diseñador.

Tabla 4. Planificación en tiempo del sistema

ACTIVIDAD	DURACIÓN
Formulación	4 días
Planificación	8 días
Análisis	32 días
Modelización	66 días
Pruebas	10 días
TOTAL	120 días

3.4. Análisis de Riesgos

Se deben analizar los posibles riesgos que se presenten, así como su incidencia en el proyecto, tanto en su desarrollo como en el producto final, tal como se detalla a continuación:

- Mal cálculo en la estimación del presupuesto.
- Requerimientos que no han sido considerados al momento de presentar las necesidades por parte del representante del Consultorio Médico La Dolorosa.
- Incidencia en la implementación del proyecto por desconocimiento de herramientas de desarrollo.
- Situaciones inesperadas relacionadas con la seguridad del software.
- Cambio de políticas de gestión.

3.4.1. Identificar los Riesgos

- Riesgos del Proyecto
 - Costo de Producción
- Riesgos Técnicos
 - Imprecisión en la Especificación de Requerimientos de Software
 - Problemas de diseño
 - Problemas en Verificación & Validación
- Riesgos de Negocios
 - Riesgo de gestión

3.4.2. Gestión y Supervisión del Riesgo

Una vez determinados los riesgos es necesario buscar una alternativa para que provoquen un impacto mínimo o en el mejor de los casos no incidan en el proyecto, para esto se utilizará un formato en el que se hace constar las causas y prevenciones necesarias.

Tabla 5. Documento de Riesgo 1

DOCUMENTO RIESGO 1
1. IDENTIFICACIÓN Mal cálculo en la estimación del presupuesto
2. TIPO Riesgo del Proyecto
3. SUBTIPO Costo de producción
4. CAUSAS Incoherencias en los cálculos, no evaluar de forma satisfactoria los factores que influyen en el cálculo (archivos, funciones, etc.) o por el contrario se puede presentar una sobre estimación.
5. PLAN DE AVERSIÓN Buscar los recursos que sean necesarios para terminar el proyecto a tiempo, y no acumular actividades de tareas al personal.

Tabla 6. Documento de Riesgo 2

DOCUMENTO RIESGO 2
<p>1. IDENTIFICACIÓN</p> <p>Requerimientos que no han sido considerados al momento de presentar las necesidades por parte del representante del Consultorio Médico La Dolorosa.</p>
<p>2. TIPO</p> <p>Riesgo Técnico</p>
<p>3. SUBTIPO</p> <p>Imprecisión en los SRS (Especificación de Requerimientos de Software)</p>
<p>4. CAUSAS</p> <p>Información deficiente.</p> <p>Poca experiencia.</p> <p>Documentos no existentes.</p>
<p>5. PLAN DE AVERSIÓN</p> <p>Buena información</p> <p>Crear documentos</p>

Tabla 7. Documento de Riesgo 3

DOCUMENTO RIESGO 3
<p>1. IDENTIFICACIÓN</p> <p>Incidencia en la implementación del proyecto por desconocimiento de herramientas de desarrollo.</p>
<p>2. TIPO</p> <p>Riesgo Técnico</p>
<p>3. SUBTIPO</p> <p>Problemas de diseño</p>
<p>4. CAUSAS</p> <p>El equipo de desarrollo puede no contar con la experiencia suficiente en cuanto a utilización de las herramientas de desarrollo, implementación.</p>
<p>5. PLAN DE AVERSIÓN</p> <p>Buscar material, manuales, talleres sobre las diferentes herramientas a ser utilizadas en el proyecto.</p>

Tabla 8. Documento de Riesgo 4

DOCUMENTO RIESGO 4
<p>1. IDENTIFICACIÓN Situaciones inesperadas relacionadas con la seguridad del software.</p>
<p>2. TIPO Riesgo Técnico</p>
<p>3. SUBTIPO Problemas en Verificación & Validación</p>
<p>4. CAUSAS Falta de experiencia del personal del proyecto en las cuestiones relacionadas a Seguridad Web, control de ingreso malicioso (hackers), seguridad de acceso físico a los equipos, seguridad del software de aplicación, falta de instalación y actualizaciones de programas de seguridad.</p>
<p>5. PLAN DE AVERSIÓN Utilizar algoritmos para encriptar información.</p>

Tabla 9. Documento de Riesgo 5

DOCUMENTO RIESGO 5
<p>1. IDENTIFICACIÓN Cambio de políticas de gestión</p>
<p>2. TIPO Riesgo de Negocios</p>
<p>3. SUBTIPO Riesgo de gestión</p>
<p>4. CAUSAS El proyecto puede sufrir retrasos, cuanto más avanzado este el desarrollo del mismo más crítico será implementar los cambios, y generación de nuevos requisitos.</p>
<p>5. PLAN DE AVERSIÓN Cumplir las actividades dentro del tiempo establecido para cada una de ellas, y obtener los resultados programados en lo que se refiere al producto final.</p>

3.5. Factibilidad

Cuando ya se han determinado los requerimientos en la fase inicial, se comprueba la factibilidad del proyecto.

3.5.1. Factibilidad Operativa

El consultorio médico La Dolorosa no dispone de un sistema informático para manejar las historias clínicas de los pacientes que asisten a esta casa de salud, es por eso que se ha visto la necesidad de implementar el Sistema para manejo de historias clínicas.

3.5.2. Factibilidad Técnica

- **Recursos Humanos**

Para el desarrollo del software intervienen las siguientes personas:

- Ligia Elena Ayala – Desarrolladora
- Ing. Jorge Delgado – Asesor de Tesis
- Dr. Fabián Arias – Consultorio Médico La Dolorosa

Este es el equipo que llevará a cabo el desarrollo e implementación del sistema informático.

- **Recursos Hardware**

Para el desarrollo del sistema informático:

- Procesador: AMD 1.6 Gb
- Memoria RAM: 2048 Mb
- Tarjeta de Red: 10/100 Mb
- Sistema Operativo: Windows 7

Para la implementación:

- Procesador: Intel Core I3 2.3 Gb
- Memoria RAM: 2048 Mb
- Tarjeta de Red: 10/100 Mb
- Sistema Operativo: Windows 7

• **Recursos Software**

Varias herramientas se han utilizado para el desarrollo e implementación, como se detalla a continuación:

- Servidor Web: Apache
- Lenguaje de Programación: PHP 5.3
- Herramienta para diseño de páginas web: Dreamweaver CS6
- Herramienta para editar gráficos: Photoshop CS6
- Base de datos: MariaDB

En base a los recursos humanos, hardware y software con los que se cuenta para desarrollar el software, se puede concluir que desde el punto de vista técnico el sistema para historias clínicas del Consultorio Médico La Dolorosa si es factible de realizarlo.

3.5.3. Factibilidad Económica

La desarrolladora cubrirá los gastos que generen el desarrollo e implementación del software.

• **Análisis Costo – Beneficio**

▪ **Costo**

- Programación \$ 500
- Literatura relacionados al tema \$ 150
- Internet \$ 100
- Varios \$ 150
- COSTO TOTAL \$ 900**

- **Determinación de Beneficios**

Cuando el software sea completado se podrá ver beneficios tangibles e intangibles para los usuarios finales (médicos y pacientes del Consultorio Médico La Dolorosa).

- **Beneficios Tangibles**

- Gestión de historias clínicas del Consultorio Médico La Dolorosa.

- **Beneficios Intangibles**

- Automatizar el proceso de manejo de historias clínicas para manipular de mejor manera la información concerniente a cada uno de los pacientes que acuden a esta casa de salud.
- Manejo eficiente de la información.

3.6. Fase de Análisis

3.6.1. Jerarquía de usuarios

En la fase de especificación de requisitos de software se hizo referencia a varios usuarios que están relacionados con el funcionamiento del sistema informático como se presenta a continuación:

- Usuario Administrador: Es el encargado de ingresar información de turnos médicos, datos de los pacientes.
- Usuario Médico: Usuario que tiene como finalidad ingresar datos para la cita médica y generar información para la historia clínica.
- Usuario Paciente: Usuario que tiene acceso a la información proporcionada tanto por el usuario Administrador, así como el usuario Médico.

3.7. DISEÑO E IMPLEMENTACIÓN DE LA APLICACIÓN

En esta etapa se definirán un conjunto de técnicas y características incluyendo las ventajas y desventajas para tener una idea amplia del software que se va a desarrollar. El producto final permitirá a los médicos del Consultorio Médico La Dolorosa que manejen las historias clínicas de los pacientes de una manera eficiente.

3.7.1. Definición de la Metodología

Existen varias metodologías para el desarrollo de software, pero se trató de encontrar una metodología que se adapte al contexto y necesidades del Consultorio Médico La Dolorosa, que garantice que el análisis, implementación y documentación proporcionen un producto de excelente calidad.

3.8. Descripción General del Sistema

El sistema permitirá a los médicos del Consultorio Médico La Dolorosa manejar historias clínicas de los pacientes, y a su vez permitirá enviar notificaciones electrónicas a los pacientes confirmando la reservación de un turno médico.

El sistema informático está desarrollado con herramientas de software libre para ejecutarse sin ningún inconveniente en plataforma Windows.

3.9. Definición del Prototipo

En esta fase se instauran los requerimientos que son necesarios para el desarrollo de la aplicación por parte del cliente; es así que se desarrolló una reunión con el Doctor Fabián Arias en la que se determinaron los lineamientos que debe cubrir la aplicación web en lo que se refiere a usuario – aplicación.

3.10. Definición de Requisitos Generales del Sistema

3.10.1. Requisitos funcionales relacionados con Médico

- **Agregar médico**

El sistema ingresará datos de médicos, con información a través de un formulario.

- **Modificar médico**

El sistema permitirá realizar modificaciones en los datos de los médicos a través de un identificador (código de cada uno).

- **Eliminar médico**

El sistema podrá eliminar los datos de médicos a través de su identificador correspondiente.

3.10.2. Requisitos funcionales relacionados con Pacientes

- **Agregar Paciente**

El sistema ingresará datos de los pacientes, con información a través de un formulario.

- **Modificar Paciente**

El sistema permitirá realizar modificaciones en los datos de pacientes a través de un identificador (código de cada uno).

- **Eliminar Paciente**

El sistema podrá eliminar los datos de pacientes a través de su identificador respectivo.

3.10.3. Requisitos funcionales relacionados con Turno médico

- **Agregar Turno médico**

El sistema ingresará datos para turno médico de un médico específico, con información a través de un formulario.

- **Modificar Turno médico**

El sistema permitirá realizar modificaciones en los datos de turnos médicos mediante un identificador (código de cada uno).

- **Eliminar Turno médico**

El sistema podrá eliminar los datos de turnos médicos a través de su identificador respectivo.

3.10.4. Requisitos funcionales relacionados con Cita médica

- **Agregar Cita médica**

El sistema ingresará datos de la cita médica, con información a través de un formulario.

- **Modificar Cita médica**

El sistema permitirá realizar modificaciones en los datos de la cita médica mediante el código de cada uno como identificador.

- **Eliminar Cita médica**

El sistema podrá eliminar los datos de la cita médica a través de su identificador.

3.11. Requisitos de rendimiento

3.11.1. Estáticos

El sistema informático puede ser instalado en una estación de trabajo que tenga un servidor web, un motor de base de datos y conexión a internet o también puede ser alojado en un hosting. El acceso dependerá del perfil que posea el usuario.

3.12. Atributos del sistema software

3.12.1. Fiabilidad

En base a los requisitos que se han establecido con el Dr. Fabián Arias del Consultorio Médico La Dolorosa se puede evidenciar que el sistema informático realizará las acciones necesarias para que cubra los requerimientos descritos.

3.12.2. Disponibilidad

El sistema estará disponible para proporcionar información en cualquier momento por su diseño modular, lo cual implica que si se debe realizar alguna modificación no afectará al sistema.

3.12.3. Seguridad

Es necesario garantizar que la información que se manipula a través del sistema informático se maneja con seguridad, en tal virtud se manejan sesiones, así como se evita ataques por inyección de SQL.

3.12.4. Portabilidad

En esta fase no está contemplado ningún requerimiento de portabilidad para este sistema informático.

3.13. Requisitos no funcionales (Normas y Estándares)

- El sistema informático se desarrollará mediante la metodología RUP.
- Para un manejo intuitivo y rápido debe proporcionar interfaces claras.

3.14. Requisitos no funcionales – Seguridad

Mediante el ingreso del nombre de usuario y la contraseña respectiva y a través de sesiones se verificará la información proporcionada y se convalidará con los datos almacenados en la base de datos, y se proporcionará según el perfil del usuario las opciones habilitadas para cada uno.

3.15. Requisitos no funcionales – Organización

Para el desarrollo del sistema deben utilizarse herramientas apropiadas que faciliten su elaboración, así como conocimiento adecuado de las mismas.

3.16. Catálogo de Usuarios

- Administrador: encargado de controlar el sistema y de su administración.
- Médico: encargado de atender las citas médicas reservadas por el paciente.
- Paciente: usuario que reserva un turno médico para posteriormente ser atendido en la cita médica respectiva.

3.17. Modelo de Negocio

Mediante los casos de uso, en esta fase se hace referencia a los procesos de uno o más elementos organizacionales.

3.18. Objetivos y Alcance del Sistema

El sistema permitirá ingresar información de turnos médicos, citas médicas, y enviará notificaciones electrónicas, esta información se almacenará en una base de datos y estará disponible para ser manipulada o generar reportes.

3.19. Establecimiento de requisitos

3.19.1. Especificación de Casos de Uso

Los casos de uso permiten de manera detallada especificar los procesos que se deben realizar de manera prioritaria y la interacción que existe entre el software y los usuarios que están directamente relacionados para la manipulación de la información. La aplicación maneja tres módulos, uno para el Administrador, otro para los Médicos y otro para los Pacientes del Consultorio Médico La Dolorosa. El módulo administrador realizará la gestión de mantenimiento; mientras que el módulo de médicos y pacientes manejarán la información referente a turnos y citas médicas.

3.20. Casos de Uso

Es necesario el diagrama de casos de uso que se ilustra a continuación:

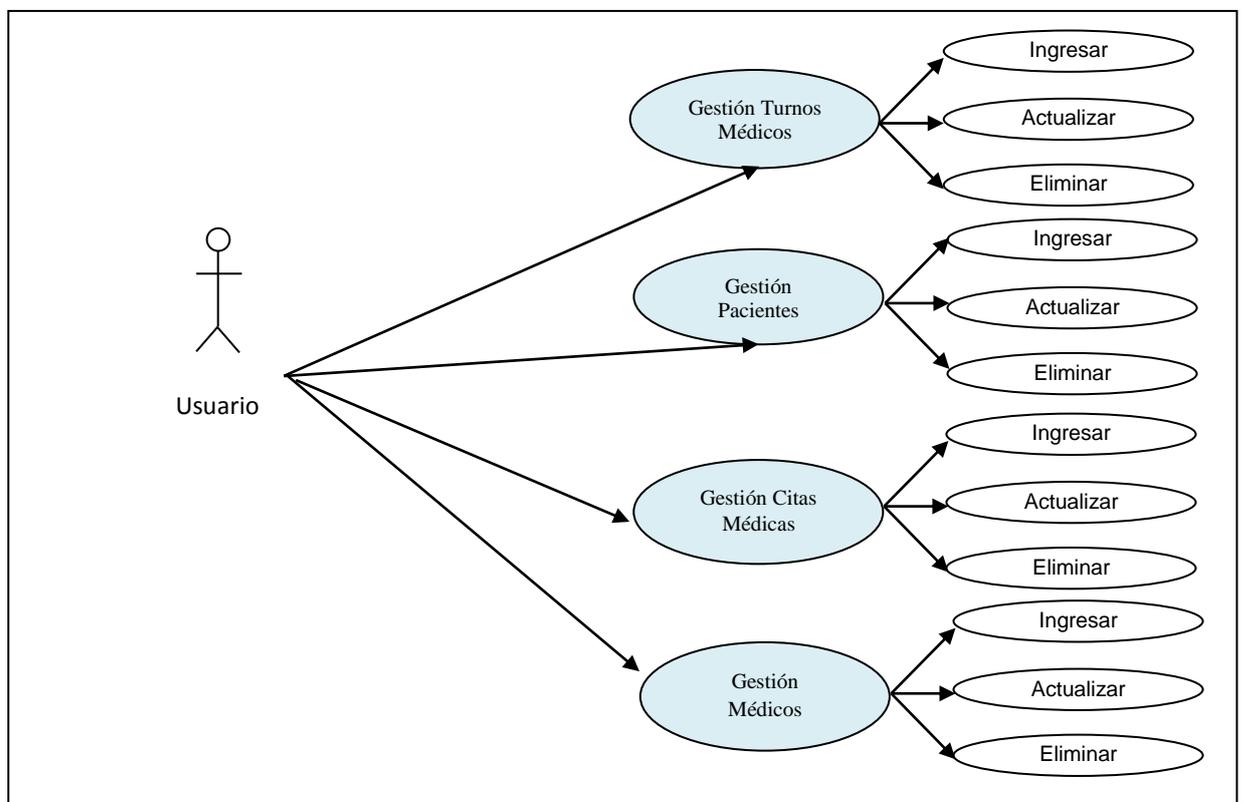


Figura 1. Diagrama de casos de uso

3.20.1. Detalle de los casos de uso

Caso de Uso (CU-1): Validación de usuario

Actores: Administrador, Instructor, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el usuario pueda ingresar su usuario y clave para ser validados con la base de datos.

Tipo: Primario.

Tabla 10. Caso de Uso 1

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario ingresa su nombre de usuario y la clave respectiva.	2. La aplicación valida los datos proporcionados con los que están almacenados en la base de datos. 2.1. Si los datos son correctos le proporciona una interfaz para que interactúe con los menús. 2.2. Si los datos son incorrectos le presenta un mensaje de error en la validación.

Caso de Uso (CU-2): Ingresar Paciente

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el usuario pueda ingresar y registrar los datos del paciente.

Tipo: Primario.

Tabla 11. Caso de Uso 2

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario da clic en el menú PACIENTES	2. La aplicación le proporciona una interfaz para que escoja la opción correspondiente a paciente.
3. El usuario da clic en la opción NUEVO PACIENTE	4. La aplicación le proporciona una interfaz para que ingrese la información.

5. El actor ingresa: nombres, apellidos, número de cédula, domicilio, celular, fecha de nacimiento, sexo, correo electrónico	6. El sistema almacena la información ingresada.
--	--

Caso de Uso (CU-3): Actualizar Paciente

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el usuario de la aplicación pueda actualizar los datos de los pacientes.

Tipo: Primario.

Tabla 12. Caso de Uso 3

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario da clic en el menú PACIENTES	2. La aplicación le proporciona una interfaz para que escoja la opción correspondiente a paciente.
3. El usuario da clic en la opción MODIFICAR PACIENTE	4. La aplicación le proporciona una interfaz con un listado de todos los pacientes ingresados en la base de datos.
5. Selecciona el paciente a modificar	6. Proporciona una interfaz con la información correspondiente al paciente seleccionado.
7. El actor modifica: nombres, apellidos, número de cédula, domicilio, celular, fecha de nacimiento, sexo, correo electrónico	8. El sistema almacena la información ingresada.

Caso de Uso (CU-4): Eliminar Paciente

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda dar de baja los pacientes.

Tipo: Primario.

Tabla 13. Caso de Uso 4

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario da clic en el menú PACIENTES	2. La aplicación le proporciona una interfaz para que escoja la opción correspondiente a paciente.
3. El usuario da clic en la opción ELIMINAR PACIENTE	4. La aplicación le proporciona una interfaz con un listado de todos los pacientes ingresados en la base de datos.
5. Selecciona el paciente a eliminar	6. Da de baja el paciente seleccionado.

Caso de Uso (CU-5): Ingresar Médico

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda registrar los datos de los médicos.

Tipo: Primario.

Tabla 14. Caso de Uso 5

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario da clic en el menú MÉDICOS	2. La aplicación le proporciona una interfaz para que escoja la opción correspondiente a médico.
3. El usuario da clic en la opción NUEVO MÉDICO	4. La aplicación le proporciona una interfaz para que ingrese la información.
5. El actor ingresa: nombres, apellidos, número de cédula, domicilio, celular, especialidad, correo electrónico	6. El sistema almacena la información ingresada.

Caso de Uso (CU-6): Actualizar Médico

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda actualizar los datos de los médicos.

Tipo: Primario.

Tabla 15. Caso de Uso 6

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario da clic en el menú MÉDICOS	2. La aplicación le proporciona una interfaz para que escoja la opción correspondiente a médico.
3. El usuario da clic en la opción MODIFICAR MÉDICO	4. La aplicación le proporciona una interfaz con un listado de todos los médicos ingresados en la base de datos.
5. Selecciona el médico a modificar	6. Proporciona una interfaz con la información correspondiente al médico seleccionado.
7. El actor modifica: nombres, apellidos, número de cédula, domicilio, celular, especialidad, correo electrónico	8. El sistema almacena la información ingresada.

Caso de Uso (CU-7): Eliminar Médico

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda dar de baja los datos de los médicos.

Tipo: Primario.

Tabla 16. Caso de Uso 7

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario da clic en el menú MÉDICOS	2. La aplicación le proporciona una interfaz para que escoja la opción correspondiente a médico.
3. El usuario da clic en la opción ELIMINAR MÉDICO	4. La aplicación le proporciona una interfaz con un listado de todos los médicos ingresados en la base de datos.
5. Selecciona el médico a eliminar	6. Da de baja el médico seleccionado.

Caso de Uso (CU-8): Ingresar Turno médico

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda registrar los turnos médicos.

Tipo: Primario.

Tabla 17. Caso de Uso 8

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Administrador ingresa a la opción MÉDICOS	2. La aplicación le proporciona una interfaz para que escoja la información correspondiente a médicos.
3. Selecciona la opción NUEVO TURNO MÉDICO	4. Presenta interfaz para que se ingrese información correspondiente al turno.
5. El actor Administrador ingresa: fecha, hora de inicio, hora de finalización, médico	6. El sistema almacena la información ingresada.

Caso de Uso (CU-9): Actualizar Turno médico

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda actualizar los datos de los turnos médicos.

Tipo: Primario.

Tabla 18. Caso de Uso 9

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario da clic en el menú MÉDICOS	2. La aplicación le proporciona una interfaz para que escoja la opción correspondiente a médico.
3. El usuario da clic en la opción MODIFICAR TURNO MÉDICO	4. La aplicación le proporciona una interfaz con un listado de todos los turnos médicos ingresados en la base de datos.
5. Selecciona el turno médico a modificar	6. Proporciona una interfaz con la información correspondiente al turno médico seleccionado.
7. El actor modifica: fecha, hora de inicio, hora de finalización, médico	8. El sistema almacena la información ingresada.

Caso de Uso (CU-10): Eliminar Turno Médico

Actores: Administrador, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el administrador de la aplicación pueda dar de baja los turnos médicos.

Tipo: Primario.

Tabla 19. Caso de Uso 10

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario da clic en el menú MÉDICOS	2. La aplicación le proporciona una interfaz para que escoja la opción correspondiente a médico.
3. El usuario da clic en la opción ELIMINAR TURNO MÉDICO	4. La aplicación le proporciona una interfaz con un listado de todos los turnos médicos ingresados en la base de datos.
5. Selecciona el turno médico a eliminar	6. Da de baja el turno médico seleccionado.

Caso de Uso (CU-11): Atiende Cita médica

Actores: Médico, aplicación.

Propósito: Proporcionar una interfaz intuitiva para que el médico pueda atender un turno médico agendado por un paciente.

Tipo: Primario.

Tabla 20. Caso de Uso 11

Acciones de los Actores	Respuestas de la Aplicación
1. El usuario Médico ingresa a la opción ATENDER TURNO MÉDICO	2. La aplicación le proporciona una interfaz para que escoja el turno médico agendado por un paciente.
3. El actor Médico ingresa: Diagnóstico, receta.	4. El sistema almacena la información ingresada.

3.21. Modelo Conceptual

Las clases identificadas son las siguientes:

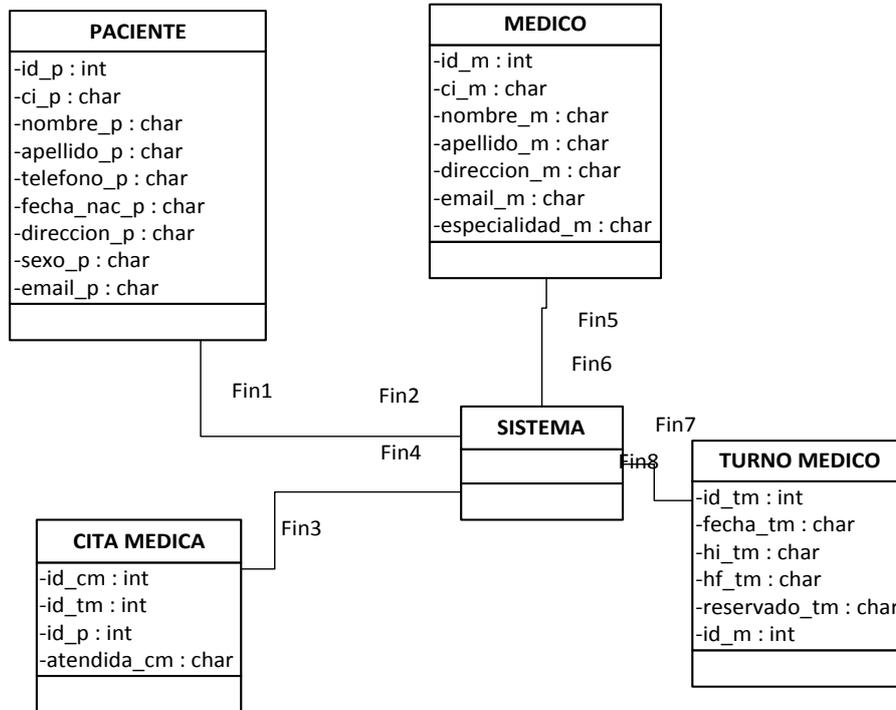


Figura 2. Clases de la base de datos

3.22. Secuencia de Actividades entre actores

El diagrama de secuencia muestra todos los eventos generados por actores externos, su orden y los eventos internos del sistema y que fluyen de los actores hacia el sistema.

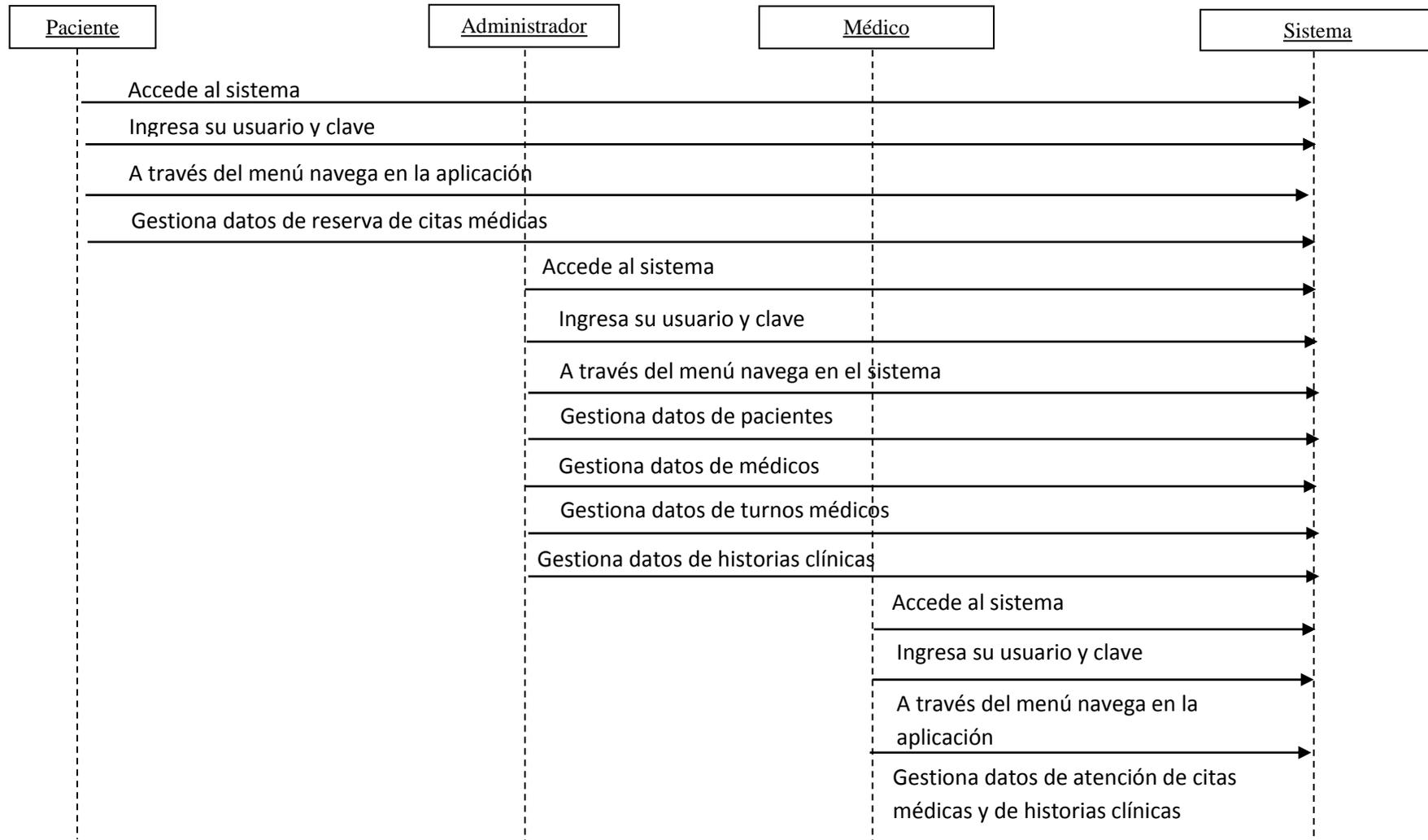


Figura 3. Secuencia de actividades entre actores y el sistema

3.23. Definición de la Interfaz

Mediante formularios y a través de un lenguaje de programación estandarizado para obtener un rendimiento óptimo en la información que interactúa entre los usuarios y el sistema, por lo que se puede aseverar que el rendimiento del sistema no dependerá de la arquitectura en la que ésta se implemente sino de los navegadores web y el sistema operativo desde donde se acceda.

Figura 4. Interfaz para validación de usuarios

The screenshot shows a web interface for user registration and login. At the top, there are links for 'Home' and 'Intranet'. The main content area is titled 'Registro Usuarios' and features a logo with a caduceus and the text 'CONSULTORIO MEDICO LA DOLOROSA'. Below the logo, there are input fields for 'Usuario:', 'Clave:', and a dropdown menu for 'Tipo:' with 'Administrador' selected. An 'INGRESAR' button is positioned below the form. The footer contains the text 'Consultorio Médico La Dolorosa'.

Figura 5. Interfaz con menú principal

The screenshot displays a main menu interface. At the top, there are links for 'Home' and 'Intranet', and a user profile section showing 'admin' with options for 'Inicio' and 'Cerrar Sesión'. Below this, there are three blue buttons with icons and text: 'PACIENTES' with an 'INGRESAR' button, 'HISTORIAS CLINICAS' with an 'INGRESAR' button, and 'MEDICO' with an 'INGRESAR' button. The footer contains the text 'Consultorio Médico La Dolorosa'.

Figura 6. Interfaz para eliminar un paciente

The screenshot shows a patient list interface. At the top, there are links for 'Home' and 'Intranet', and a user profile section showing 'admin' with options for 'Inicio' and 'Cerrar Sesión'. Below this, there is a table titled 'Listado de Pacientes' with the following data:

ID	NOMBRE	DEDULA	TELEFONO	CORREO	ELIMINAR
1.	Escobar Alexander	0111111111	0995331273	pvdq2006@hotmail.com	Eliminar

Figura 7. Interfaz para registrar un nuevo paciente

Home Intranet

admin | Inicio | Cerrar Sesión

Todos los campos con * son obligatorios.

Datos del Paciente

Nombres : *

Apellidos : *

Número de Cédula : *

Domicilio : *

Celular : *

Fecha de Nacimiento : *

Sexo : -->seleccione v * *

Correo Electrónico : *

GUARDAR

Figura 8. Interfaz para modificar un paciente

Home Intranet

admin | Inicio | Cerrar Sesión

Datos del Paciente

Nombres : Alexander *

Apellidos : Escobar *

Domicilio : fdfj sdf df ds *

Celular : 0996331273 *

Fecha de Nacimiento : 01-03-1990 *

Sexo : Hombre v *

Correo Electrónico : pvda2006@hotmail.com *

Siguiente

3.24. Diseño arquitectónico

El entorno WEB propuesto posee seguridad a nivel de arquitectura de aplicación como se puede ver en la figura a continuación:

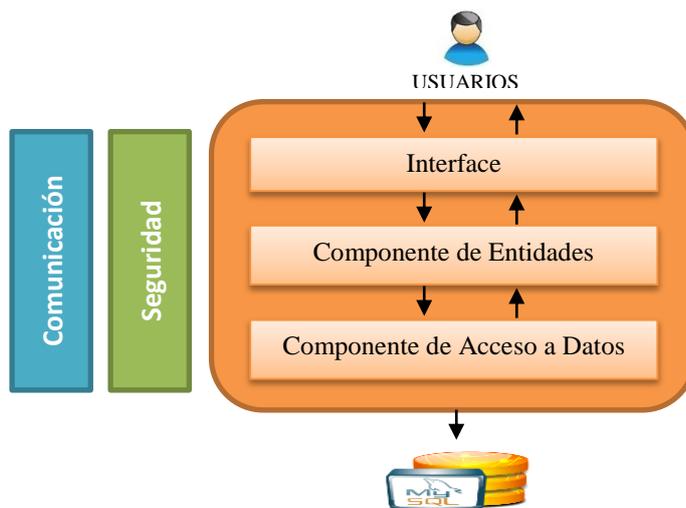


Figura 9. Arquitectura del sistema.

La comunicación entre componentes se realiza siguiendo un esquema de protección o seguridad de acceso por concepción natural de la aplicación y en cuanto se refiere a interfaz, está diseñada para facilitar la utilidad por parte del usuario pero también para evitar que se viole la integridad del sistema mediante la introducción de código malicioso sean estos exploits, SQL Injections, entre otros. El uso de sesiones también contribuye a un aumento considerable de medidas de prevención y seguridad.

3.25. Diseño de la base de datos

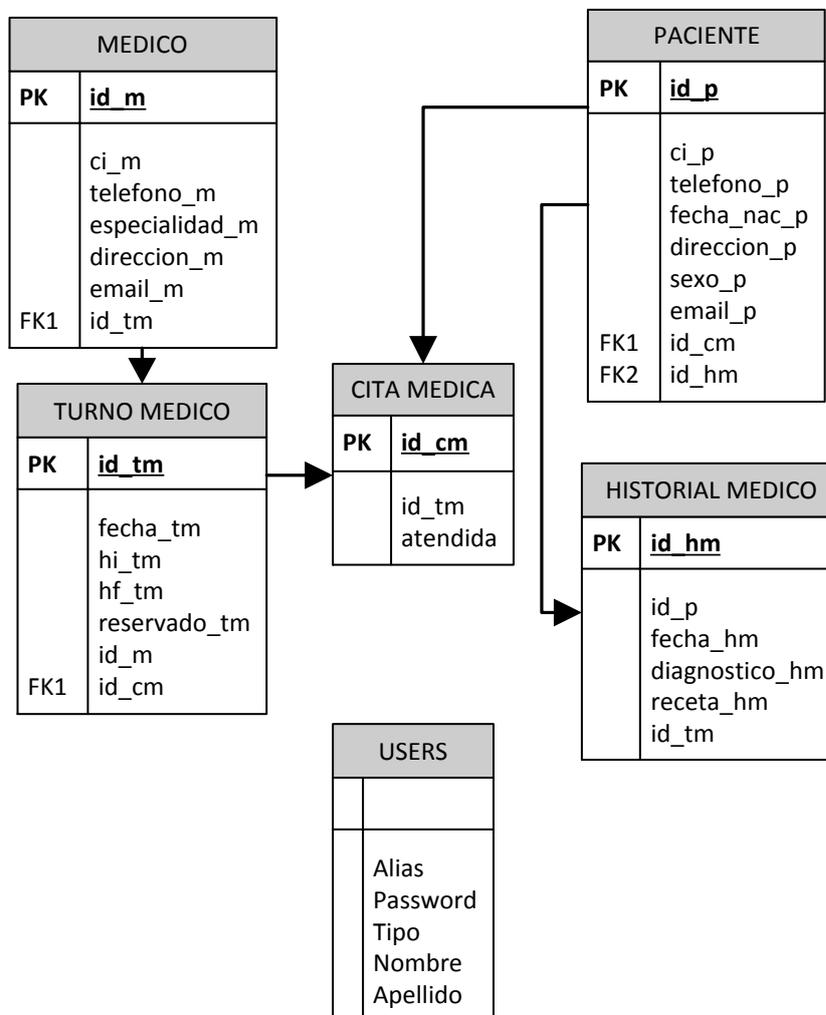


Figura 10. Estructura de la base de datos

3.26. Diseño de procedimientos

3.26.1. Registrar paciente

```
<?php require("Usuario.php");
    require("Paciente.php");
    session_start();
    require("script/general.php");
    $BD = new BaseDatos();
    if(isset($_REQUEST['id']))
    {
        $NuevoPaciente = new Paciente();
        $NuevoPaciente->ci_p = $_REQUEST['identidad'];
        $NuevoPaciente->telefono_p = $_REQUEST['celular'];
        $NuevoPaciente->fecha_nac_p = $_REQUEST['fecnac'];
        $NuevoPaciente->direccion_p = $_REQUEST['domicilio'];
        $NuevoPaciente->sexo_p = $_REQUEST['sexo'];
        $NuevoPaciente->email_p = $_REQUEST['correo'];
        $NuevoPaciente->Registrar();
        $NuevoUsuario = new Usuario();
        $NuevoUsuario->Alias = $_REQUEST['identidad'];
        $NuevoUsuario->Password = $_REQUEST['identidad'];
        $NuevoUsuario->Tipo = 'PACIENTE';
        $NuevoUsuario->Nombre = $_REQUEST['nombres'];
        $NuevoUsuario->Apellido = $_REQUEST['apellidos'];
        $NuevoUsuario->Registrar();
        echo'<script>alert("LOS DATOS HAN SIDO ALMACENADOS
SATISFACTORIAMENTE")</script>';
    }
?>
<html>
<head>
<title><?php title(1);?></title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<script language="javascript" src="script/general.js"></script>
<link href="script/estilo.css" type="text/css" rel="stylesheet">
<script language="javascript" src="script/validacadenas.js"></script>
<script language="javascript" src="script/validamail.js"></script>
<script src="calendario/popcalendar.js"></script>
<script>
function valida(form)
{
    if(form.nombres.value=="")
    {
        alert('No ha ingresado sus Nombres');
        form.nombres.className='inputrojo';
    }
}
```

```

        form.nombres.focus();
        return;
    }
    if(!IsAlpha(form.nombres.value))
    {
        alert('Hay caracteres no permitidos en el campo Nombres');
        form.nombres.className='inputrojo';
        form.nombres.focus();
        return;
    }
    if(form.apellidos.value=="")
    {
        alert('No ha ingresado sus Apellidos');
        form.apellidos.className='inputrojo';
        form.apellidos.focus();
        return;
    }
    if(!IsAlpha(form.apellidos.value))
    {
        alert('Hay caracteres no permitidos en el campo Apellidos');
        form.apellidos.className='inputrojo';
        form.apellidos.focus();
        return;
    }
    if(form.identidad.value=="")
    {
        alert('No ha ingresado su Documento de Identidad');
        form.identidad.className='inputrojo';
        form.identidad.focus();
        return;
    }
    if(!IsAlphaNum(form.identidad.value))
    {
        alert('Hay caracteres no permitidos en el campo Documento de Identidad');
        form.identidad.className='inputrojo';
        form.identidad.focus();
        return;
    }
    if(form.domicilio.value=="")
    {
        alert('No ha ingresado su Domicilio');
        form.domicilio.className='inputrojo';
        form.domicilio.focus();
        return;
    }
    if(!IsAlphaNum(form.domicilio.value))
    {
        alert('Hay caracteres no permitidos en el campo Domicilio');
        form.domicilio.className='inputrojo';
        form.domicilio.focus();
    }

```

```

        return;
    }

    if(form.celular.value=="")
    {
        alert('No ha ingresado un numero de Celular');
        form.celular.className='inputrojo';
        form.celular.focus();
        return;
    }
    if(!IsNum(form.celular.value))
    {
        alert('Hay caracteres no permitidos en el campo Celular');
        form.celular.className='inputrojo';
        form.celular.focus();
        return;
    }

    if(form.fecnac.value=="")
    {
        alert('No ha ingresado una Fecha de Nacimiento');
        form.fecnac.className='inputrojo';
        form.fecnac.focus();
        return;
    }

    if(form.sexo.value=="")
    {
        alert('No ha ingresado su Sexo');
        form.sexo.className='selectrojo';
        form.sexo.focus();
        return;
    }
    if(!IsAlpha(form.sexo.value))
    {
        alert('Hay caracteres no permitidos en el campo Sexo');
        form.sexo.className='selectrojo';
        form.sexo.focus();
        return;
    }

    if(form.correo.value=="")
    {
        alert('Debe ingresar su correo electronico');
        form.correo.className='inputrojo';
        form.correo.focus();
        return;
    }
    if(!validarEmail(form.correo.value))
    {

```



```

class="input" type="text" name="apellidos" size="40"><span>Ingrese sus
apellidos.<br></span></a><font class="rojo">&nbsp;&bull;</font></td>
</tr>
<tr>
<td align="left"><a class="faq"><input class="input" type="text" name="identidad" size="40"
width="35%" class="azul">N&uacute;mero de C&eacute;dula :</td><td align="left"><a
class="faq"><input class="input" type="text" name="identidad" size="40"
maxlength="10"><span>C&eacute;dula sin gui&oacute;n. Ej.
0604049761<br></span></a><font class="rojo">&nbsp;&bull;</font></td>
</tr>
<tr>
<td align="left"><a class="faq"><input
width="35%" class="azul">Domicilio :</td><td align="left"><a class="faq"><input
class="input" type="text" name="domicilio" size="40"><span>Ingrese su
domicilio.<br></span></a><font class="rojo">&nbsp;&bull;</font></td>
</tr>
<tr>
<td align="left"><a class="faq"><input
width="35%" class="azul">Celular :</td><td align="left"><a class="faq"><input
class="input" type="text" name="celular" size="40" maxlength="10"><span>Ingrese su
n&uacute;mero de celular.<br></span></a><font
class="rojo">&nbsp;&bull;</font></td>
</tr>
<tr>
<td align="left"><a class="faq"><input
width="35%" class="azul">Fecha de Nacimiento :</td><td align="left"><a
class="faq"><input class="input" type="text" name="fecnac" size="40" id="dateArrival"
onClick="popUpCalendar(this, frmdatospersonales.dateArrival,'dd-mm-
yyyy')"><span>Ingrese su fecha de nacimiento. aaaa-mm-dd<br></span></a><font
class="rojo">&nbsp;&bull;</font></td>
</tr>
<tr>
<td align="left"><a class="faq"><input
width="35%" class="azul">Sexo :</td><td align="left">
<select
name="sexo" class="select">
<option
value="">-->seleccione</option>
<option
value="Hombre">Hombre</option>
<option
value="Mujer">Mujer</option>
</select><font class="rojo">&nbsp;&bull;</font>
</td>
</tr>
<tr>
<td align="left"><a class="faq"><input
width="35%" class="azul">Correo Electr&oacute;nico.</td><td align="left"><a

```



```

        </tr>
    </table>
    </FORM>
    <?php
        }    ?>
    </td>
</tr>
</table>
<p align="center" class="legal"><? inferior();?></p>
</body>
</html>

```

3.26.2. Modificar paciente

```

<?php require("Usuario.php");

        require("Paciente.php");
        require("script/general.php");
        session_start();
        $BD = new BaseDatos();
        if (isset($_REQUEST['param']))
        {
                $NuevoPaciente = new Paciente();
                $NuevoPaciente-
>UpdateP($_REQUEST['id_p'],$_REQUEST['celular'],$_REQUEST['fecnac'],$_REQUEST['d
omicilio'],$_REQUEST['sexo'],$_REQUEST['correo']);

                $NuevoUsuario = new Usuario();
                $NuevoUsuario-
>UpdateNA($_REQUEST['ci_p'],$_REQUEST['nombres'],$_REQUEST['apellidos']);
                echo"<script>alert('Los cambios se han almacenado satisfactoriamente
en la base de datos');</script>";
        }
?>
<html>
<head>
<title><?php title(4);?></title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<link href="script/estilo.css" type="text/css" rel="stylesheet">
<script language="javascript" src="script/validacadenas.js"></script>
<script language="javascript" src="script/validamail.js"></script>
<script src="calendario/popcalendar.js"></script>
</head>

<body bgcolor="#FFFFFF" leftmargin="0">
<table width="100%" border="0" cellspacing="0" cellpadding="0" align="center">
    <tr>
        <td>

```

```

<?php
if(isset($_SESSION['AuthUser']))
{
    superiorp();

    $SQL = "SELECT * FROM paciente WHERE
ci_p='".$_SESSION['AuthUser']."'";
    $busca = $BD->ExecQuery($SQL);
    while($row=mysql_fetch_object($busca))
    {
        $SQL1 = "SELECT * FROM users WHERE
Alias='".$row->ci_p."'";
        $busca1 = $BD->ExecQuery($SQL1);

        while($row1=mysql_fetch_object($busca1))
        {
            ?>
            <form name="frmdatospersonales" method="post"
action="Paciente_Modificar_Datos.php">
            <br><br>
            <table width="70%" border="0" cellspacing="0" cellpadding="0"
align="center">
            <tr>
            <td>
            <fieldset style="border:1px solid #2e679c">
            <legend class="azul" style="border: #003;"><strong>&nbsp;&nbsp;&nbsp;Datos
del Paciente&nbsp;&nbsp;&nbsp;</strong></legend>
            <table width="100%" border="0" cellspacing="3"
cellpadding="0" align="center">
            <tr>
            <td width="35%" class="azul">Nombres :</td><td
align="left"><a class="faq"><input class="input" type="text" name="nombres" size="45"
value="<?php echo".$row1->Nombre."; ?>"><span>Ingrese sus
nombres.<br></span></a><font class="rojo">&nbsp;&nbsp;&nbsp;&bull;</font></td>
            </tr>
            <tr>
            <td width="35%" class="azul">Apellidos :</td><td
align="left"><a class="faq"><input class="input" type="text" name="apellidos" size="45"
value="<?php echo".$row1->Apellido."; ?>"><span>Ingrese sus
apellidos.<br></span></a><font class="rojo">&nbsp;&nbsp;&nbsp;&bull;</font></td>
            </tr>
            <tr>
            <td width="35%" class="azul">Domicilio :</td><td
align="left"><a class="faq"><input class="input" type="text" name="domicilio"
size="45" value="<?php echo".$row->direccion_p."; ?>"><span>Ingrese su
domicilio.<br></span></a><font class="rojo">&nbsp;&nbsp;&nbsp;&bull;</font></td>
            </tr>
            <tr>

```

```

        <td width="35%" class="azul">Celular :</td><td
align="left"><a class="faq"><input class="input" type="text" name="celular" size="45"
maxlength="10" value="<?php echo".$row->telefono_p."; ?>"><span>Ingrese su
n&uacute;mero de celular.<br></span></a><font
class="rojo">&nbsp;&bull;</font></td>
    </tr>
    <tr>
        <td width="35%" class="azul">Fecha de Nacimiento :</td><td
align="left"><a class="faq"><input class="input" type="text" name="fecnac" size="45"
value="<?php echo".$row->fecha_nac_p."; ?>" id="dateArrival"
onClick="popUpCalendar(this, frmdatospersonales.dateArrival,'dd-mm-
yyyy')"><span>Ingrese su fecha de nacimiento. aaaa-mm-dd<br></span></a><font
class="rojo">&nbsp;&bull;</font></td>
    </tr>
    <tr>
        <td width="35%" class="azul">Sexo :</td><td align="left">
        <select name="sexo" class="select">
            <option value="">-->seleccione</option>
            <option value="Hombre" <?php if($row-
>sexo_p=="Hombre") echo"selected"; ?>>Hombre</option>
            <option value="Mujer" <?php if($row->sexo_p=="Mujer")
echo"selected"; ?>>Mujer</option>
        </select><font class="rojo">&nbsp;&bull;</font>
    </td>
    </tr>
    <tr>
        <td width="35%" class="azul">Correo
Electr&oacute;nico.</td><td align="left"><a class="faq"><input class="input"
type="text" name="correo" size="45" value="<?php echo".$row->email_p.";
?>"><span>Correo Electr&oacute;nico.<br></span></a><font
class="rojo">&nbsp;&bull;</font></td>
    </tr>
</table>
</fieldset>
</td>
</tr>
</table><br><br><br>
<p align="center">
    <input type="hidden" name="id_p" value="<?php echo".$row-
>id_p."; ?>">
    <input type="hidden" name="id" value="<?php
echo".$_REQUEST['id']."; ?>">
    <input type="hidden" name="ci_p" value="<?php echo".$row-
>ci_p."; ?>">
    <input type="hidden" name="param" value="1">
    <input class="boton" type="submit" value="Siguiente" />
</p>
</form>
<?php

```

```

        }
    }
} //del if
else {
    echo'<P align="center"><br><br><br>NO TIENE PERMISO PARA
ACCEDER</P>';
}??>
</td>
</tr>
</table>
<p align="center" class="legal"><? inferior();?></p>
</body>
</html>

```

3.26.3. Eliminar paciente

```

<?php require("Usuario.php");

    require("Paciente.php");
    require("script/general.php");
    session_start();
    $BD = new BaseDatos();
    if (isset($_REQUEST['param']))
    {
        $NuevoPaciente = new Paciente();
        $NuevoPaciente->Delete($_REQUEST['id']);

        $NuevoUsuario = new Usuario();
        $NuevoUsuario->Delete($_REQUEST['id']);
    }
?>
<html>
<head>
<title><?php title(3);?></title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<link href="script/estilo.css" type="text/css" rel="stylesheet">
<script language="javascript" src="script/validacadenas.js"></script>
<script language="javascript" src="script/validamail.js"></script>
<script src="calendario/popcalendar.js"></script>
<script type="text/javascript">
function confirma(){
    if(confirm("¿Realmente desea eliminarlo?"))
    { alert("El registro ha sido eliminado"); }
    else { return false }
}
</script>
</head>

```

```

<body bgcolor="#FFFFFF" leftmargin="0">
<table width="100%" border="0" cellspacing="0" cellpadding="0" align="center">
<tr>
<td>
<?php
if(isset($_SESSION['AuthUser']) )
{
superior(); ?>
<table width="70%" border="0" cellspacing="0"
cellpadding="0" align="center">
<tr>
<td>
<fieldset style="border:1px solid
#ff0000">
<legend class="azul"
style="border:#004080;"><strong>&nbsp;&nbsp;&nbsp;Listado de
Pacientes&nbsp;&nbsp;&nbsp;</strong></legend>
<br><br>
<table
width="100%" border="0" cellspacing="5" cellpadding="0">
<tr bgcolor="#0099FF">
<td align="left">N&omicron;n;</td>
<td align="left">NOMBRE</td>
<td align="left">CEDULA</td>
<td align="left">TELEFONO</td>
<td align="left">CORREO</td>
<td align="left">ELIMINAR</td>
<tr>
<?php
$cont=1;
$SQL="SELECT * FROM users WHERE Tipo='PACIENTE'";
$result=$BD->ExecQuery($SQL);
while($row=mysql_fetch_object($result))
{
echo'<tr>';
$SQL1="SELECT * FROM paciente WHERE ci_p='".$row->Alias.'";
$result1=$BD->ExecQuery($SQL1);

```

```

while($row1=mysql_fetch_object($result1))
    {
        $color = (($cont % 2) == 0) ? '#EFEFEF' : '#F5F5F5';
        echo' <td>'. $cont.'.</td>
                                <td>'. $row->Apellido.'&nbsp;'. $row-
>Nombre.'.</td>
                                <td>'. $row->Alias.'.</td>
                                <td>'. $row1->telefono_p.'.</td>
                                <td>'. $row1->email_p.'.</td>
                                <td align="center"><a
href="ListadoPacientesEliminar.php?param=1&id='. $row->Alias.'" onclick="if(confirm()
== false) return false">&nbsp;Eliminar</a></td>;
                                }
        echo'</tr>'; $cont++;
    }
if($cont==1)
    {
        echo"<p align=center>No se han encontrado datos para su
b&uacute;squeda.</p>";
    }
                                ?>
                                </table>
                                </fieldset>
                                </td>
                                </tr>
                                </table><br><br>
<?php
}
else {
    echo'<P align="center"><br><br><br>NO TIENE PERMISO PARA
ACCEDER</P>';
}??
</td>
</tr>
</table>

```

<p align="center" class="legal"><? inferior();?></p>
 </body>
 </html>

3.27. Pruebas de Software

Las pruebas que se necesitan desarrollar comprenden los siguientes tipos:

- a. Pruebas Unitarias: son las realizadas sobre cada módulo en forma independiente.

Se proponen métricas para evaluar el rendimiento del sistema informático como se detalla a continuación:

- Ingreso de datos: 1= Bajo, 3= Medio, 5= Alto.
- Eliminación de datos: 1= Bajo, 3= Medio, 5= Alto.
- Modificación de datos: 1= Bajo, 3= Medio, 5= Alto.
- Reportes: 1= Bajo, 3= Medio, 5= Alto.
- Crear turno médico: 1= Bajo, 3= Medio, 5= Alto.
- Reservar turno médico: 1= Bajo, 3= Medio, 5= Alto.
- Atender turno médico: 1= Bajo, 3= Medio, 5= Alto.

Tabla 21. Resultados Pruebas Unitarias

Métricas \ Módulos	Ingreso de datos	Eliminación de datos	Modificación de datos	Reportes	Crear turno médico	Reservar turno médico	Atender turno médico
Administrador	5	5	5	3	5	-	-
Paciente	-	-	5	3	-	5	-
Médico	-	-	5	3	-	-	5

Luego de realizar la evaluación de los módulos a fin de comprobar errores, se puede determinar que el producto final alcanza los siguientes parámetros: el Módulo Administrador alcanza una valoración de 23 de 25, el Módulo Paciente 13 de 15 y el Módulo Médico 13 de 15; esto implica que el software alcanza parámetros muy buenos y está listo para su utilización.

- b. Pruebas Funcionales: son las que se basan en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el sistema informático.

Tabla 22. Resultados Pruebas Funcionales

PRUEBA	RESULTADO
Funcionales	Una vez desarrollado el sistema informático se realizaron las respectivas pruebas para denotar el nivel de cumplimiento de los requerimientos, obteniendo los resultados esperados; presenta una cooperación entre los diferentes módulos lo que supone será de gran ayuda para pacientes y médicos.

CAPÍTULO IV

4. RESULTADOS

Luego que el sistema informático ha sido desarrollado e implementado, es necesario que se evalúen cada uno de los requerimientos que se detallaron en capítulos anteriores; es así que en el Capítulo II se determinó el framework Joomla como el que se adecuaba de mejor manera para acoplarse a las necesidades del software para el Consultorio Médico La Dolorosa.

El sistema informático permite ingresar y manipular información precisa de pacientes, médicos, turnos médicos y citas médicas, tomando en cuenta que el tiempo desde que el usuario procede a realizar una petición (consulta a la base de datos) y visualizar el respectivo resultado (información proporcionada desde la base de datos) no sobrepasa los 5 segundos.

Además se puede observar aspectos como:

- El límite de tiempo para que una persona mantenga su atención en una página mientras espera es de 3 segundos.
- El tiempo para que los usuarios perciban que una página web reacciona instantáneamente es de 0.1 segundos.
- Teniendo en cuenta que una gran parte de los usuarios no siempre van a acceder desde computadores con características óptimas, el sistema informático no excede los 15 Kb lo que, para un módem básico de 500 kbps, representa 3 segundos de tiempo de descarga en el navegador.

Asimismo inciden también características hardware del equipo desde donde se navega, tales como memoria y procesador, así como el tipo de enlace para su conexión a internet.

Mediante los criterios explicados en párrafos anteriores, se estipula que el sistema informático maneja datos correctos, seguros y veraces; se ajustan a lo que el usuario requiere, y los tiempos de respuesta son mínimos (no exceden los 4 segundos).

4.1. Comprobación de la Hipótesis

4.1.1. Hipótesis

En la presente investigación se planteó la siguiente hipótesis: “Análisis de vulnerabilidades en frameworks de código abierto optimizará la gestión de notificaciones electrónicas”.

4.1.1.1. Variable independiente

Análisis de vulnerabilidades en frameworks de código abierto

4.1.1.2. Variable Dependiente

Gestión de notificaciones electrónicas. *Caso aplicativo*: historias clínicas en el consultorio médico “La Dolorosa”.

4.1.2. Comprobación

4.1.2.1 Vulnerabilidades

Para la comprobación de la hipótesis planteada en este proyecto se utiliza una herramienta especializada en encontrar vulnerabilidades en aplicaciones web realizadas con Joomla como es JoomlaScan.

JoomlaScan es un software especializado en encontrar vulnerabilidades en webs desarrolladas con Joomla. Existen otras herramientas para encontrar

vulnerabilidades en sitios web como: acunetix, w3af, nessus, que también se pueden utilizar pero lo que diferencia a JoomlaScan de los demás es que es una herramienta diseñada exclusivamente para sitios desarrollados bajo la plataforma Joomla. Las características de JoomlaScan son las siguientes:

- Realizado en lenguaje Perl.
- Detecta vulnerabilidades como: inyección SQL, LFI, RFI, XSS entre otros.
- Basado en OWASP (proyecto de código abierto dedicado a determinar y combatir las causas que hacen que el software sea inseguro).
- Recopila los avisos publicados, que hacen referencia a Joomla, de securityfocus.com para que después de la detección de la versión y los componentes instalados se muestren los posibles bugs que les puedan afectar.

En la siguiente captura de pantalla se muestra los resultados de evaluar un sitio desarrollado en Joomla, se ha identificado varios componentes instalados, así como la utilización de RSFirewall. Se detalla varios consejos de seguridad y las posibles vulnerabilidades recopiladas de securityfocus.com:

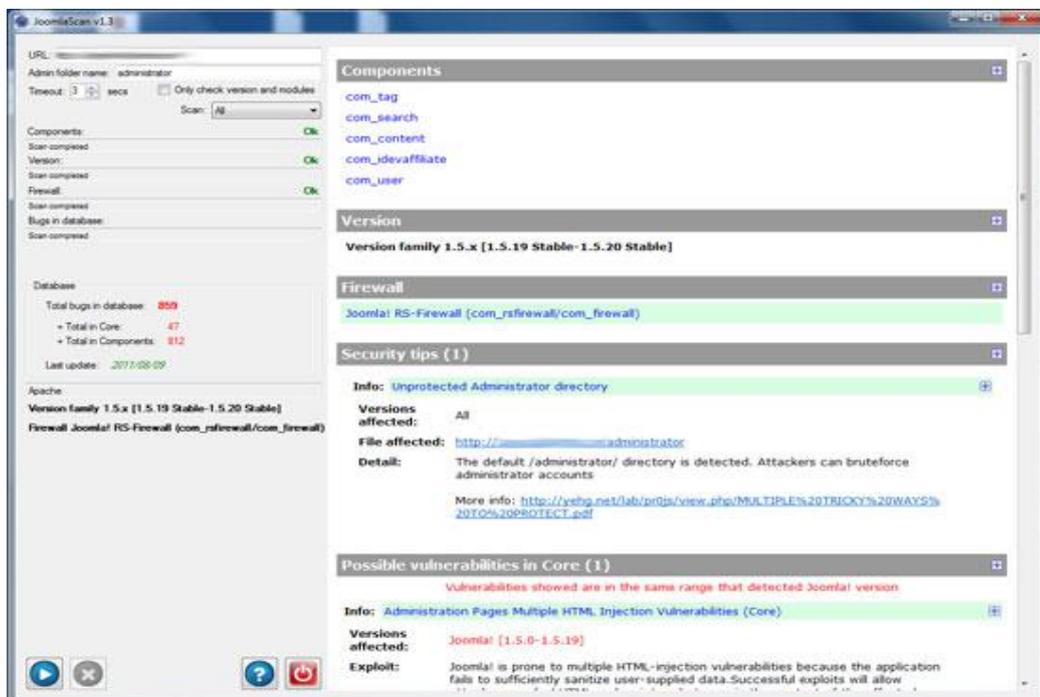


Figura 11. Ejemplo de evaluación de un sitio con JoomlaScan

Sometida la aplicación web a esta herramienta y para la comprobación de la hipótesis planteada, presenta los siguientes resultados:

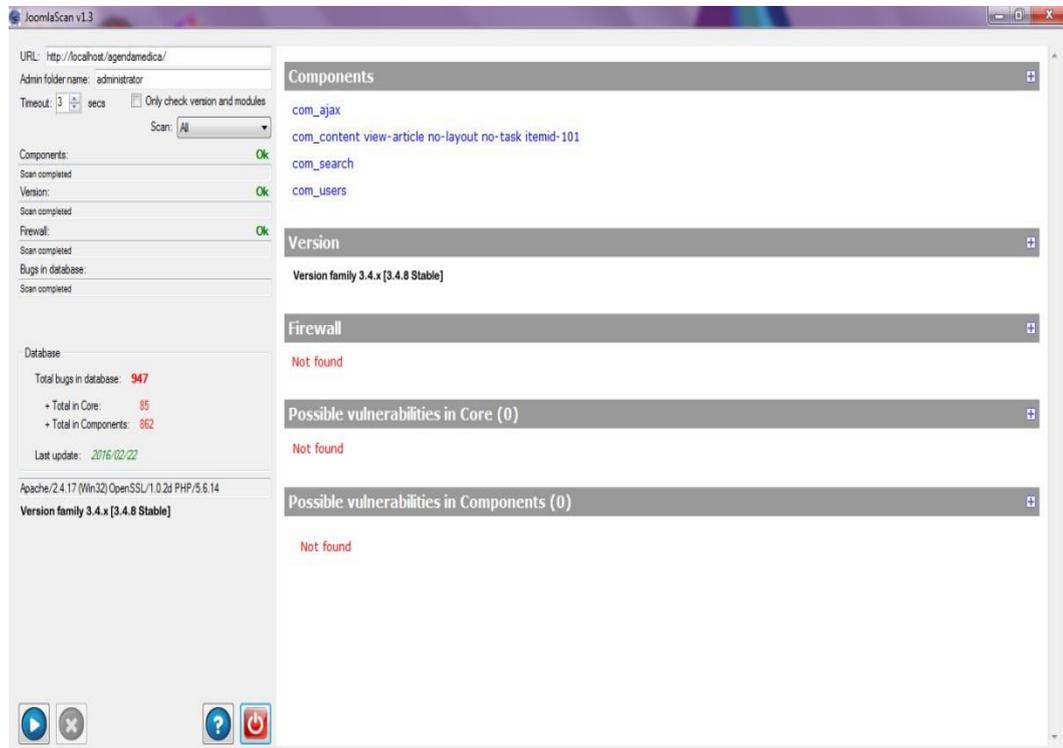


Figura 12. Evaluación del sistema informático con JoomlaScan

Tabla 23. Resultados Vulnerabilidades JoomlaScan

Nº	Vulnerabilidad	Parámetro
1	Core	Comprueba ficheros que varían de una versión a otra. Comprueba ficheros que contienen el ID de la última revisión. Envía instrucciones para que se muestren ciertos contenidos solicitados por un usuario específico.
2	Módulos	Verifica sistema de privilegios para usuarios. Verifica visibilidad y accesibilidad de contenidos. Comprueba el funcionamiento de los elementos para mostrar en la página otros contenidos además del componente, los elementos los considera una “instancia” de un tipo de módulo.

Una vez que se han analizado los resultados se puede ver que la versión de Joomla que se está ejecutando es Versión 3.4.8, los componentes son: com_ajax, com_content, com_search, com_users; todos los directorios están protegidos y no existen vulnerabilidades del tipo SQL injection, LFI, RFI, XSS, lo cual implica que la información que maneja incluyendo las notificaciones electrónicas se caracterizan por la eficiencia, eficacia y pertinencia. En tal virtud y en base a la seguridad y confiabilidad del sistema informático la hipótesis planteada está demostrada como afirmativa.

4.1.2.2 Gestión notificaciones electrónicas

Tabla 24. Proceso para gestión notificaciones electrónicas

Proceso	Manual	Sistema para Historias Clínicas
Turnos Médicos	El paciente realiza una llamada telefónica o acude personalmente al consultorio. La secretaria busca en la agenda para ver si hay algún turno disponible, si hay un turno lo separa y registra la reservación del turno en la agenda, este proceso tarda de 4 a 6 minutos.	El paciente accede con su usuario y password, en el menú selecciona reservar turno, escoge el médico y pulsa el botón reservar, seguidamente recibe un correo electrónico y un mensaje de texto al celular que confirma la reservación del turno, el mismo que contiene la siguiente información: médico, fecha y hora; este proceso se realiza en un lapso de 10 a 15 segundos.

Tabla 25. Proceso para servicios de atención médica

Servicios atención médica	Manual	Sistema para Historias Clínicas
Crear turnos médicos	La secretaria crea una hoja para un día específico para cada médico y establece turnos con una diferencia de 30 minutos cada uno.	El usuario Administrador mediante el menú accede a la opción Turno Médico, Nuevo Turno, escoge el doctor y le asigna el día, hora de inicio, hora de finalización.
Reservar Turnos Médicos	El usuario realiza una llamada telefónica o acude personalmente al consultorio; la secretaria busca en la agenda para ver si hay algún turno disponible, si hay un turno lo separa y registra la reservación del turno en la agenda.	El usuario paciente, mediante el menú selecciona reservar turno, escoge la fecha, la hora y el médico disponible.
Atender Turnos Médicos	El médico solicita a la secretaria que le pase el listado con los turnos reservados, de acuerdo a la hora hace pasar a los pacientes, y registra el diagnóstico en la ficha correspondiente para guardar la información de cada cita médica.	El usuario médico, a través del menú Atender turno médico, selecciona el paciente, tiene acceso a las historias clínicas del paciente y procede a llenar el diagnóstico y la respectiva receta.
Historias Clínicas	El médico busca en el archivador la carpeta que contiene el historial	El usuario Médico puede mediante el menú y el número de cédula acceder a

	médico del paciente.	todo el historial clínico del paciente
Receta	Una vez realizado el diagnóstico del paciente, el médico procede a escribir la receta y las indicaciones a mano. El tiempo estimado para este proceso es de 5 minutos	El usuario Médico una vez que ha ingresado los datos de la consulta médica tiene la opción para imprimir directamente la receta con las indicaciones para el paciente. El tiempo estimado para este proceso es de 20 segundos

4.2. CONCLUSIONES Y RECOMENDACIONES

4.2.1. Conclusiones

- Las vulnerabilidades en un sistema informático, por lo general permiten acceder mediante un programa o script para tener control del mismo, para minimizar la incidencia de vulnerabilidades es imprescindible tomar precauciones como limitar el control de acceso a la información, realizar los controles sobre entradas de datos y establecer pruebas de vulnerabilidad en el ciclo de vida del desarrollo del sistema informático.
- Existen varios tipos de exploits para notificaciones electrónicas en frameworks de código abierto como: SQL injection, Cross site scripting (XSS), según el estudio realizado en el Capítulo II se estableció que las principales vulnerabilidades son las siguientes: **comprometimiento del sitio web**, mediante un estudio realizado por StopBadware y Commtouch se determinó que Wordpress es el framework que presenta más probabilidad de ser hackeado con un 28%; **SQL injection** donde Joomla presenta la extensión jHackGuard para impedir el acceso por inyección de SQL los otros frameworks han realizado muy pocos esfuerzos por evitar esta amenaza y

finalmente **Cross site scripting (XSS)** presentando Joomla varios componentes como Jsecure y Jomdefender para asegurar el filtrado de los campos de entrada, Wordpress y Drupal no han creado ningún componente o plugin para ayudar a combatir esta vulnerabilidad.

- Para la implementación del sistema de historias clínicas se utilizó el framework Joomla, que permite integrar el uso de nuevas tecnologías móviles en entornos web (Cloud / nube), según el análisis realizado en el Capítulo II mediante métricas para comparación de frameworks, Joomla obtuvo una valoración de 23 puntos, por lo que se puede aseverar que proporciona un nivel de seguridad alto en relación a las vulnerabilidades por XSS, así como de SQL injection; facilita un buen número de módulos, plugins y componentes para el desarrollo de aplicaciones web, anti SPAM, protecciones de acceso a la administración, contra intrusiones y hackeos; niveles de cifrado de la contraseña, sistema de seguridad de doble factor; sin embargo durante el desarrollo del sistema para minimizar los riesgos de vulnerabilidades se utilizó Joomlscan, herramienta diseñada exclusivamente para la plataforma de Joomla que permite determinar posibles amenazas tanto a nivel de core, módulos, componentes y plugins.

4.2.2. Recomendaciones

- La arquitectura del sistema de historias clínicas permite a futuro integrar nuevos módulos como contabilidad, farmacia, laboratorio; sin embargo al tratarse de aplicaciones en la web, es necesario considerar en el desarrollo e implementación políticas de control de vulnerabilidad a través de Joomlscan.
- Para la integración del sistema de historias clínicas es necesario utilizar estándares en el formato de la historia clínica que permite la interoperabilidad con profesionales de la salud y los usuarios finales (pacientes).

4.3. BIBLIOGRAFÍA

- Figuerola, C. G. (2010). Software libre y software gratuito para la innovación docente y la creación de contenidos orientados al EEES. Salamanca: España.
- Cobo, A., Gómez, P., Pérez, D., & Rocha, R. (2005). PHP Y MySQL Tecnologías para el desarrollo de aplicaciones web. España: Díaz de Santos.
- Coronel, C., Morris, S., & Rob, P. (2011). BASE DE DATOS Diseño, implementación y administración (Novena ed.). México D.F: CENGAJE Learnig.
- Kendall, K., & Kendall, J. (2011). Análisis y diseño de sistemas. México: Pearson.
- Pavón, J. (2011). Creación de un portal con PHP y MySQL. México: Alfaomega Grupo editor S.A.
- Sánchez, J. (2007). Sistema de gestión de base de datos. Lima: RitisaGraff.
- Silberschat, A., Korth, H., & Sudarshan, S. (2002). FUNDAMENTOS DE BASES DE DATOS. Madrid: McGRAW-HILL.
- Stallman, R. M. (2004). Software libre para una sociedad libre. (J. Rowan, D. Sanz, & L. Trinidad, Trads.) Madrid: Traficantes de sueños.

WEBGRAFÍA

Desarrollo Web. (09 de 06 de 2008). Recuperado el 15 de 08 de 2015, de <http://www.desarrolloweb.com/articulos/introduccion-css3.html>

PYME Networks. (2014). Recuperado el 15 de 08 de 2015, de <http://www.pymenetworks.es/sobre-nosotros/aviso-legal>

INTERNET MARKETING. (2015). Recuperado el 15 de 08 de 2015, de <http://michaelroca.com/2013/03/31/ventajas-y-caracteristicas-de-crear-un-blog-wordpress/>

On Web Solutions. (2015). Recuperado el 15 de 08 de 2015, de http://onwebsolutions.es/blog_HTML5_CSS3_Javascript.html

Wikipedia. (01 de 09 de 2015). Recuperado el 15 de 08 de 2015, de <https://es.wikipedia.org/wiki/Joomla>

Wikipedia. (31 de 08 de 2015). Recuperado el 15 de 08 de 2015, de <https://es.wikipedia.org/wiki/JavaScript>

Wikipedia. (05 de 09 de 2015). Recuperado el 15 de 08 de 2015, de <https://es.wikipedia.org/wiki/MySQL>

Wikipedia. (25 de 06 de 2015). Recuperado el 15 de 08 de 2015, de https://es.wikipedia.org/wiki/Aplicaci%C3%B3n_web

Wikipedia. (31 de 08 de 2015). Recuperado el 15 de 08 de 2015, de <https://es.wikipedia.org/wiki/JavaScript>

Wikipedia. (25 de 08 de 2015). Recuperado el 15 de 08 de 2015, de https://es.wikipedia.org/wiki/Hoja_de_estilos_en_cascada

Bermúdez, G., Delgado, C., & Gonzáles, A. (Agosto de 2014). Google Académico. Recuperado el 26 de Septiembre de 2014, de <http://rus.ucf.edu.cu/index.php/rus/article/view/351/202>

Buitrago, J. (27 de 05 de 2011). Drupal Groups. Recuperado el 15 de 08 de 2015, de <https://groups.drupal.org/node/148379>

Cáceres, P., & Marcos, E. (s.f.). www.dlsi.ua.es/~jaime/webe/articulos/s112.pdf. Recuperado el 19 de Agosto de 2014, de <http://www.dlsi.ua.es/~jaime/webe/articulos/s112.pdf>

Capacitación y Formación, I. (2015). C-FIP. Recuperado el 15 de 08 de 2015, de http://www.c-fip.com.ar/?page_id=143

Cobo, A., Gómez, P., Pérez, D., & Rocha, R. (2005). PHP Y MySQL Tecnologías para el desarrollo de aplicaciones web. España: Díaz de Santos.

Coronel, C., Morris, S., & Rob, P. (2011). BASE DE DATOS Diseño, implementación y administración (Novena ed.). México D.F: CENGAJE Learnig.

Durán, L. (2007). El gran libro del PC interno. Barcelona: MARCOMBO, S.A .

Figuerola, C. G. (2010). Software libre y software gratuito para la innovación docente y la creación de contenidos orientados al EEES. Salamanca: España.

García, J. (2014). INTERNET. Recuperado el 15 de 08 de 2015, de <https://mjalice.files.wordpress.com/2011/05/internet1.doc>

Gersbach, M. (01 de 04 de 2005). Drupal Hispano. Recuperado el 15 de 08 de 2015, de <http://drupal.org.es/caracteristicas>

Gezal, D. (2015). Wordpress. Recuperado el 15 de 08 de 2015, de <https://prezi.com/9ftf9spi7zba/wordpress/>

HTML BASICO. (s.f.). Recuperado el 15 de 08 de 2015, de www.uprm.edu/cti/docs/seminario/html_basico.pdf

Kendall, K., & Kendall, J. (2011). Análisis y diseño de sistemas. México: Pearson.

Maigua, G., & López, E. (2012). BUENAS PRACTICAS EN LA GESTION Y DIRECCION DE PROYECTOS INFORMATICOS. Argentina: edUTecNe.

Martínez, C. (16 de 12 de 2014). ¿Qué son los Frameworks? Recuperado el 15 de 08 de 2015, de <http://www.nubelo.com/blog/que-son-los-frameworks/>

Martínez, J. (23 de 06 de 2015). Wordpress. Recuperado el 15 de 08 de 2015, de http://carpetainformaticajlm.blogspot.com/2015_06_01_archive.html

Pavón, J. (2011). Creación de un portal con PHP y MySQL. México: Alfaomega Grupo editor S.A.

Pérez, D. (19 de 12 de 2007). Maestros del Web. Recuperado el 15 de 08 de 2015, de <http://www.maestrosdelweb.com/una-serie-de-administradores-de-contenidos/>

Romeo, G. (19 de 03 de 2015). VULNERABILIDADES CMS. Obtenido de <http://www.seas.es/blog/informatica/vulnerabilidad-cms/>

Sánchez, J. (2007). Sistema de gestión de base de datos. Lima: RitisaGraff.

Silberschat, A., Korth, H., & Sudarshan, S. (2002). FUNDAMENTOS DE BASES DE DATOS. Madrid: McGRAW-HILL.

Stallman, R. M. (2004). Software libre para una sociedad libre. (J. Rowan, D. Sanz, & L. Trinidad, Trads.) Madrid: Traficantes de sueños.

Tagle, J. (10 de 09 de 2014). Wordpress Avanzado. Recuperado el 15 de 08 de 2015, de <http://wpavanzado.com/que-es-wordpress/>

Tamayo, M., Lemes, J., & Naranjo, T. (s.f.). Google Accadémico. Recuperado el 26 de Septiembre de 2014, de <http://jornada2011.sld.cu/index.php/jornada/2011/paper/viewFile/105/39>

Vaswani, V. (2010). FUNDAMENTOS DE PHP. MÉXICO: McGraw Hill.

Web Design. (s.f.). Recuperado el 15 de 08 de 2015, de <http://www.webdesigncuba.net/ventajas-y-desventajas-de-joomla>

Web Empresa. (s.f.). Recuperado el 15 de 08 de 2015, de <http://www.webcreacion.com/disenio-web/web-empresa.html>

Wikipedia. (06 de 06 de 2008). Recuperado el 15 de 08 de 2015, de <http://www.ammanu.edu.jo/wiki1/es/articles/d/r/u/Drupal.html>

Wikipedia. (18 de 01 de 2013). Wikipedia. Recuperado el 15 de 08 de 2015, de <https://es.wikipedia.org/wiki/Framework>

Wikipedia. (03 de 09 de 2015). Wikipedia. Recuperado el 15 de 08 de 2015, de https://es.wikipedia.org/wiki/Sistema_de_gesti%C3%B3n_de_contenidos

4.4. ANEXOS

ENTREVISTA REALIZADA AL DR. FABIÁN ARIAS GALENO DEL CONSULTORIO MÉDICO LA DOLOROSA.

INGRESOS:

Se necesita toda la información necesaria para poder levantar una base de datos que permita automatizar la información.

Para la búsqueda de información se debe tomar todo el tiempo necesario ya que de esto dependerá el funcionamiento correcto de la aplicación web planteada.

INFORMACIÓN RECOPIADA:

¿El consultorio cuenta con una base de datos para el agendamiento de citas médicas de sus pacientes?

- No cuenta con una base de datos, la atención médica se la realiza en orden de llegada o se entregan turnos por teléfono a una hora determinada.

¿Qué dificultad tiene al momento de atender a sus pacientes?

- Las citas médicas se las agenda de forma manual.
- No se cuenta con datos actualizados (automatizados).
- No hay un programa que facilite el ahorro de tiempo y recursos, todo se lo hace manualmente.
- Las recetas médicas se las escribe a mano.

¿Si usted contara con una aplicación web para el manejo de sus pacientes, facilitaría su trabajo?

- Por supuesto que sí, esto ahorraría tiempo y recursos, tanto para mis pacientes como para mi persona.

PROCESO:

Luego de obtener la información se realizará todos los procesos necesarios para diseñar una aplicación web para historias clínicas que pueda ser utilizada por el doctor del Consultorio Médico La Dolorosa para mejorar la calidad de atención médica a los pacientes que acuden a este centro de salud.

Riobamba, 08 de octubre de 2015

Doctor

Fabián Arias

PROPIETARIO DEL CONSULTORIO MÉDICO “LA DOLOROSA”

Presente

De mi consideración:

Reciba un cordial y atento saludo, a la vez augurarle el mejor de los éxitos en la labor que desempeña a favor de la salud.

Por medio de la presente pongo a consideración el siguiente proyecto de tesis:

Elaboración de un software denominado: **ANÁLISIS DE VULNERABILIDADES EN FRAMEWORKS DE CÓDIGO ABIERTO PARA OPTIMIZAR LA GESTION DE NOTIFICACIONES ELECTRONICAS. CASO APLICATIVO: HISTORIAS CLÍNICAS EN EL CONSULTORIO MÉDICO “LA DOLOROSA”**. Proyecto que será guiado por el Ingeniero Jorge Delgado Docente de la Universidad Nacional de Chimborazo, para su correcta aplicación.

Por lo expuesto anteriormente solicito de la manera más comedida se dignen autorizar la realización del sistema propuesto para su consultorio médico y a la vez facilite toda la información que se requiera según el avance del proyecto.

Segura de contar con una respuesta favorable a la presente le anticipo mi más sincero agradecimiento.

Atentamente,



Ligia Elena Ayala Torres

EGRESADA DE ING. EN COMPUTACIÓN – UNACH

Recibido 8 de Octubre / 2015



Centro Médico La Dolorosa
Dr. Fabián Arias N.
MÉDICO CIRUJANO
Libro XII Folio 158
CD. MD: 473

CONSULTORIO MEDICO "LA DOLOROSA"



Dr. Fabián Arias Naranjo

MÉDICO - CIRUJANO

Medicina y Cirugía general - Pediatría

Gineco Obstetricia - Planificación familiar

CERTIFICADO

A petición verbal de la parte interesada tengo a bien Certificar que la señora **AYALA TORRES LIGIA ELENA** con C.C. N° **0603585522**, ha desarrollado e implementado en su totalidad el Sistema para **HISTORIAS CLÍNICAS EN EL CONSULTORIO MÉDICO "LA DOLOROSA"**, cumpliendo todos los requerimientos que fueron solicitados para su elaboración, se realizó la verificación y pruebas del sistema, y al no encontrarse requerimientos incompletos, defectuosos o no aceptables, se considera que el estado del mismo es satisfactorio.

Es todo cuanto puedo Certificar en honor a la verdad, autorizando a la interesada que haga uso del presente como crea conveniente.

Riobamba, 23 de febrero de 2016.



Dr. Fabián Arias Naranjo

PROPIETARIO DEL CONSULTORIO MÉDICO "LA DOLOROSA"