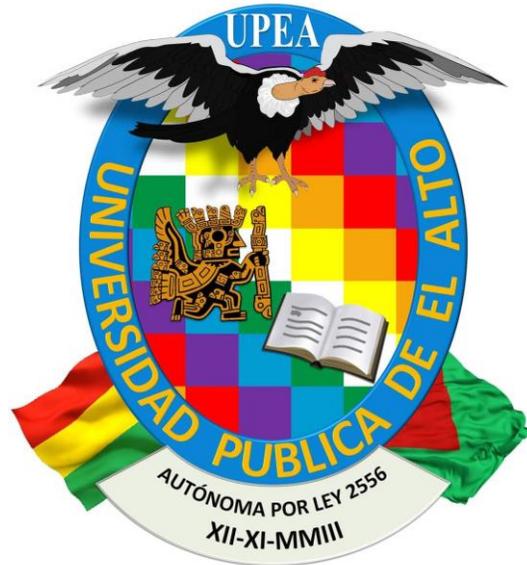


UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

“SISTEMA WEB PARA EL SEGUIMIENTO DE DIAGNÓSTICO Y TRATAMIENTO DE PACIENTES”

CASO: CENTRO MÉDICO “SAN SIMÓN”

**Para Optar al Título de Licenciatura en Ingeniería de Sistemas
MENCIÓN: INFORMÁTICA Y COMUNICACIONES**

Postulante: Julio Cesar Lecoña Laura

Tutor Metodológico: Ing. Dionicio Henry Pacheco Rios

Tutor Especialista: Ing. Gabriel Reynaldo Sirpa Huayhua

Tutor Revisor: Ing. Milton Osvaldo Zurita Benito

EL ALTO – BOLIVIA

2021

DEDICATORIA:

A Dios por haberme dado la oportunidad de llegar hasta este momento y lograr este objetivo, gracias por regalarme la vida y salud durante todo este tiempo, por poner a mi alrededor personas de bien, por guiar mi camino lejos del peligro y por estar siempre conmigo en mis éxitos, pero sobre todo en mis fracasos ayudándome a levantarme y seguir adelante.

Este proyecto de grado es dedicado a mi familia por el tiempo y paciencia que me brindaron y en especial a mi mamita Fortunata quien siempre confió en mí durante todos estos años, para la obtención de este primer camino al éxito en mi vida.

A mis amigos, por motivarme cuando quise rendirme y porque me demostraron su cariño. Muchas gracias.

AGRADECIMIENTOS

A Dios, por haberme dado la vida y permitirme haber llegado hasta este momento tan importante de mi formación profesional por protegerme durante todo mi camino y darme fuerzas para superar obstáculos y dificultades a lo largo de toda mi vida.

Agradecer a todos los que tuvieron que ver en el desarrollo y conclusión de este proyecto de grado.

A mi Tutor Metodológico, Ing. Dionicio Henry Pacheco Riospor su enseñanza, confianza, paciencia, tiempo, apoyo y motivación que me brindo desde el momento que empecé con este proyecto.

A mi Tutor Especialista, Ing. Gabriel Reynaldo Sirpa Huayhua por compartir sus conocimientos, brindándome recomendaciones con paciencia y motivación durante el desarrollo de este proyecto.

A mi Tutor Revisor, Ing. Milton Osvaldo Zurita Benito por su disponibilidad de tiempo, sus observaciones y recomendaciones brindadas en la realización del presente proyecto.

A la Universidad Pública de El Alto, por acogerme en sus aulas durante todos los años de estudio, así también a la carrera Ingeniería de Sistemas y a mis compañeros (as) de estudio por su apoyo incondicional.

Al Centro Médico San Simón por brindarme la oportunidad y la confianza para el desarrollo de mi proyecto de grado.

RESUMEN

El presente proyecto de grado se desarrolló para el Centro Médico San Simón de la Ciudad de El Alto, en atención a la problemática de no contar con un sistema web de seguimiento de diagnóstico y tratamiento de pacientes.

En el Centro Médico “San Simón” realiza el registro de diagnósticos y tratamientos, los cuales se realizan de forma manual a falta de un sistema que controle dichas tareas y pueda facilitar lo que conlleva una demora en la entrega de resultados a los pacientes lo que dificulta varios factores, entre los cuales los registros no se encuentran centralizados, lo cual dificulta llevar a cabo un control adecuado de la información del paciente provocando retrasos considerables en la atención o en ocasiones pérdida de pacientes.

Es así que se plantea desarrollar e implementar un sistema web para el seguimiento de diagnóstico y tratamiento de pacientes, con la utilización de la metodología UWE (UML-Based Web Engineering) ingeniería web basada en UML y herramientas de desarrollo de aplicación web, para el lado del servidor (back-end) tendremos el lenguaje de programación PHP con el Framework Laravel y para el lado del cliente (front-end) se aplicará el Framework Bootstrap con todas sus características incluidas.

Así mismo, para determinar la calidad del sistema web desarrollado, se hace uso de los factores de calidad ISO/IEC 9126.

Finalmente se realiza el análisis de costos utilizando COCOMO II y su modelo de estimación post arquitectura el cual es más detallado y se aplica cuando la arquitectura del proyecto está completamente definida.

INDICE INDICATIVO

1	MARCO PRELIMINAR	1
1.1	INTRODUCCIÓN	1
1.2	ANTECEDENTES	2
1.2.1	Antecedentes Institucionales	2
1.2.2	Antecedentes Institucionales	2
1.3	PLANTEAMIENTO DEL PROBLEMA	4
1.3.1	Problema Principal	4
1.3.2	Problemas Secundarios	4
1.4	OBJETIVOS	5
1.4.1	Objetivo General	5
1.4.2	Objetivos Específicos	5
1.5	JUSTIFICACIÓN	5
1.5.1	Justificación Técnica	5
1.5.2	Justificación Económica	6
1.5.3	Justificación Social	6
1.6	METODOLOGÍA	6
1.6.1	Metodología de Desarrollo UWE	7
1.6.2	Métodos	8
1.7	HERRAMIENTAS	9
1.8	LÍMITES Y ALCANCES	11
1.8.1	Límites	11
1.8.2	Alcances	12
1.9	APORTES	12
2	MARCO TEÓRICO	13
2.1	INTRODUCCIÓN	13
2.2	SISTEMA	13
2.3	WEB	13
2.4	SISTEMA WEB	14
2.5	DIAGNÓSTICO Y TRATAMIENTO DE PACIENTES	15
2.5.1	SEGUIMIENTO	15
2.6	INGENIERÍA DE SOFTWARE	16
2.6.1	Modelos de desarrollo de Software	16
2.6.2	MÉTRICAS DE CALIDAD	19

2.6.3	MÉTODO DE ESTIMACIÓN DE COSTOS DE SOFTWARE.....	26
2.7	METODOLOGÍA DE DESARROLLO DEL SISTEMA WEB.....	29
2.7.1	METODOLOGÍA UWE.....	29
2.7.2	Características de una aplicación Web	30
2.7.3	Características de la metodología UWE.....	34
2.7.4	Fases de la metodología UWE	35
2.7.5	Ciclo de la metodología UWE	36
2.7.6	Características de una aplicación Web.....	37
2.7.7	Requisitos de Desarrollo de una Aplicación Web.....	37
2.8	ARQUITECTURA DEL SOFTWARE.....	38
2.8.1	Patrón Modelo Vista Controlador (MVC).....	38
2.8.2	Ciclo de Vida del patrón MVC	39
2.8.3	Ventajas y Desventajas del Patrón MVC.....	40
2.9	HERRAMIENTAS	41
2.9.1	Base de datos.....	41
2.9.2	Frontend.....	42
2.9.3	Backend	44
2.9.4	Frameworks.....	45
2.10	SEGURIDAD DEL SISTEMA.....	47
2.10.1	ISO/IEC 27000	48
2.11	ENCRIPCIÓN AES-256	49
2.11.1	Características de AES-256.....	49
2.11.2	Formas de uso.....	49
3	MARCO APLICATIVO.....	50
3.1	INTRODUCCIÓN.....	50
3.2	CAPTURA Y ANALISIS DE REQUERIMIENTOS	51
3.2.1	Requerimientos Funcionales	51
3.2.2	Requerimientos No Funcionales	52
3.3	DISEÑO DEL SISTEMA WEB	53
3.3.1	Casos de Uso.....	53
3.3.2	Diseño Conceptual	62
3.3.3	Diseño de Navegación.....	63
3.3.4	Diseño de Presentación.....	68
3.4	DESARROLLO DEL SISTEMA WEB.....	74
3.5	PRUEBAS DEL SISTEMA WEB	81
3.5.1	Prueba de Caja Blanca.....	81

3.5.2	Prueba de Caja Negra	87
3.6	FASE DE MANTENIMIENTO.....	88
4	ANÁLISIS DE CALIDAD, COSTOS Y SEGURIDAD	89
4.1	CALIDAD DE SOFTWARE	89
4.1.1	Introducción.....	89
4.1.2	Técnica ISO 9126.....	89
4.2	COSTOS DEL SOFTWARE.....	99
4.2.1	Introducción.....	99
4.2.2	Análisis de Costos del Sistema.....	99
LDC = 9 636	101
4.2.3	Costo Total del software	103
4.3	SEGURIDAD DEL SOFTWARE.....	103
4.3.1	Introducción.....	103
4.3.2	Seguridad de base de datos	104
4.3.3	Seguridad de autenticación.....	104
4.3.4	Seguridad de la aplicación.....	105
5	CONCLUSIONES Y RECOMENDACIONES	106
5.1	CONCLUSIONES.....	106
5.2	RECOMENDACIONES.....	106
	Bibliografía	108

INDICE DE FIGURAS

Figura 2. 1	Funcionamiento de internet y Petición.....	15
Figura 2. 2	Modelo de Desarrollo Incremental.....	17
Figura 2. 3	Modelo de desarrollo Ágil.....	18
Figura 2. 4	Modelo de Desarrollo Ágil basado en un Plan.....	19
Figura 2. 5	Ingeniería web basada en UML.....	30
Figura 2. 6	Actividades de la Metodología UWE	31
Figura 2. 7	Casos de usos	31
Figura 2. 8	Análisis de casos de uso (Diagrama de Contenido)	32
Figura 2. 9	Elementos del Diseño Navegacional	32
Figura 2. 10	Modelo Navegacional UWE.....	33
Figura 2. 11	Elementos de Diseño de Presentación.....	33
Figura 2. 12	Diagrama de Presentación	34
Figura 2. 13	Ciclo de la Metodología UWE.....	36
Figura 2. 14	Arquitectura de Aplicación Web con el Patrón MVC	39
Figura 2. 15	Ciclo de Vida del MVC	40
Figura 2. 16	Bootstrap: Tamaño de diferentes dispositivos	46
Figura 3. 1	Caso de uso general	54
Figura 3. 2	Caso de Uso - Inicio de sesión.....	55
Figura 3. 3	Registro de Paciente.....	57
Figura 3. 4	Registro de Usuarios.....	59
Figura 3. 5	Médico	61
Figura 3. 6	Modelo Físico de la Base de Datos.....	63
Figura 3. 7	Navegación - General	64
Figura 3. 8	Diagrama de Navegación Nueva Consulta.....	64
Figura 3. 9	Diagrama de Navegación Nuevo Paciente	65
Figura 3. 10	Diagrama de Navegación Historial	66
Figura 3. 11	Diagrama de Navegación Médico.....	66
Figura 3. 12	Diagrama de Navegación Enfermero	67
Figura 3. 13	Diagrama de Navegación Especialidades	68
Figura 3. 14	Diagrama de Presentación del Sistema.....	68
Figura 3. 15	Diagrama de Presentación del Login.....	69
Figura 3. 16	Diagrama de Presentación del Menú Principal.....	70

Figura 3. 17	Diagrama de presentación de Consultas.....	70
Figura 3. 18	Diagrama de Presentación del Historial Clínico.....	71
Figura 3. 19	Diagrama de Presentación del Historial Médico	71
Figura 3. 20	Diagrama de Presentación de Medición Paciente	72
Figura 3. 21	Diagrama de Presentación de Historial Signos Vitales	72
Figura 3. 22	Diagrama de Presentación de Médicos	73
Figura 3. 23	Diagrama de Presentación de Enfermeros.....	73
Figura 3. 24	Diagrama de Presentación de Especialidad	74
Figura 3. 25	Interfaz - Principal del Sistema.....	74
Figura 3. 26	Interfaz - Principal de Inicio de Sesión	75
Figura 3. 27	Interfaz – Menú Principal.....	75
Figura 3. 28	Interfaz – Pacientes.....	76
Figura 3. 29	Interfaz – Nuevo Paciente	76
Figura 3. 30	Interfaz – Registro de Nuevo Consulta.....	76
Figura 3. 31	Interfaz – Historial del Paciente.....	77
Figura 3. 32	Interfaz – Registro de Nuevo Paciente	77
Figura 3. 33	Interfaz – Historial Médico de Pacientes.....	78
Figura 3. 34	Interfaz – Medición del Paciente	78
Figura 3. 35	Interfaz – Historial Signos Vitales del Paciente.....	78
Figura 3. 36	Interfaz – Médicos.....	79
Figura 3. 37	Interfaz – Registro de Nuevo Medico	79
Figura 3. 38	Interfaz – Enfermeros.....	79
Figura 3. 39	Interfaz – Registro de Nuevo Enfermero	80
Figura 3. 40	Interfaz – Especialidades	80
Figura 3. 41	Interfaz – Registro de Nueva Especialidad.....	80
Figura 4. 1	Autenticación de Usuario.....	105

INDICE DE TABLAS

Tabla 2. 1	Dominios de información de Punto Función.....	20
Tabla 2. 2	Cálculos de los Punto Función.....	21
Tabla 2. 3	Preguntas para calcular la usabilidad	24
Tabla 2. 4	Preguntas para Calcular la Eficiencia.	26
Tabla 3. 1	Fases del Modelado UWE	50
Tabla 3. 2	Requerimientos funcionales.....	52
Tabla 3. 3	Requerimientos No Funcionales	52
Tabla 3. 4	Descripción de Actores.....	53
Tabla 3. 5	Caso de Uso – Inicio de sesión.....	56
Tabla 3. 6	Caso de Uso – Registro de Pacientes	57
Tabla 3. 7	Caso de uso – Designación de Especialidad	58
Tabla 3. 8	Caso de Uso – Registro de Usuarios.....	59
Tabla 3. 9	Caso de Uso – Médico.....	61
Tabla 3. 10	Prueba de Caja Blanca - Camino 1.....	83
Tabla 3. 11	Prueba de Caja Blanca - Camino 2.....	84
Tabla 3. 12	Prueba de Caja Blanca - Camino 3.....	84
Tabla 3. 13	Prueba de Caja Blanca - Camino 4.....	85
Tabla 3. 14	Prueba de Caja Blanca - Camino 5.....	86
Tabla 3. 15	Prueba de Caja Blanca - Camino 6.....	87
Tabla 3. 16	Prueba Autenticación.....	87
Tabla 4. 1	Cálculo de Punto Función.....	90
Tabla 4. 2	Valores de Ajuste de Complejidad	91
Tabla 4. 3	Ajuste de Complejidad del Sistema.....	91
Tabla 4. 4	Valores de ajuste de complejidad	95
Tabla 4. 5	Evaluación de preguntas para calcular la usabilidad.....	96
Tabla 4. 6	Evaluación de preguntas para calcular la usabilidad.....	98
Tabla 4. 7	Resultado total de Calidad.....	98
Tabla 4. 8	Conversión de puntos función.....	100
Tabla 4. 9	Relación de valores del modelo COCOMO	101
Tabla 4. 10	Costos Total del Proyecto.....	103

1 MARCO PRELIMINAR

1.1 INTRODUCCIÓN

La realidad actual se enfrenta a una tecnología que ha ido evolucionando cada vez más, debido a esto los sistemas informáticos se han convertido en una parte importante de instituciones y empresas para mejorar la administración de la información, siendo este un motivo necesario para estar a la par de las nuevas tecnologías colaborando con un mejor control y funcionamiento de las mismas.

El Centro Médico “San Simón”, entidad que brinda el servicio de atención en salud a la población de la ciudad de El Alto, procesa la información derivada de la atención médica de sus pacientes de manera manual. En el momento de la evaluación a la clínica, se evidenció que este procedimiento toma un tiempo significativo en la atención al paciente y no hay un correcto control al gestionar los datos de los pacientes.

El presente proyecto se centra en la gestión eficiente de la información, la cual, tiene como objetivo ser de primordial ayuda en el seguimiento de diagnósticos y tratamientos médicos mediante un sistema, puesto que el sistema planteado permitirá generar información de manera ágil y eficiente, permitiendo que, con todas las medidas de seguridad y confidencialidad, el personal médico tenga mayor control y seguimiento de los tratamientos de enfermedades y, como resultado, coadyuvar a ofrecer mejores servicios a los pacientes del centro médico.

El presente sistema facilitará al personal médico del Centro Médico San Simón la manipulación de registros de cada paciente de forma rápida, información que actualmente se maneja de forma manual, lo que ocasiona la reducción en la cantidad de atención de pacientes.

1.2 ANTECEDENTES

1.2.1 Antecedentes Institucionales

El Centro Médico “San Simón”, institución de salud de carácter privado, se encuentra ubicado en el Complejo Fabril, Surtidor San Pablo Av. Sucre (a dos cuadras hacia arriba), Parada Bus 34, Zona Alto Lima, 3ra Sección de la ciudad de El Alto.

Tiene como finalidad brindar atención médica a la población de todas las edades, que permitan responder las dificultades de las diferentes enfermedades que afectan a la población.

1.2.2 Antecedentes Institucionales

Cómo resultado de la revisión bibliográfica de trabajos académicos afines, se encontraron las siguientes referencias a nivel internacional y nacional.

Internacional

- **Desarrollo de un Sistema Web de Control de Citas, para un Hospital del Día**, Quito Ecuador, 2013. Tiene como objetivo Implementar un sistema informatizado que pretende conseguir la identificación correcta de los pacientes atendidos, monitorización cronológica e inmediata de los circuitos asistenciales, disminución de los errores en el manejo de la información asistencial, registro de signos y síntomas del paciente, registro de los actos y decisiones asistenciales tomadas por los profesionales. El modelo de software que se empleó para la elaboración de la aplicación fue la metodología XP para la programación del Sistema usando un patrón de Modelo-Vista-Controlador. (DAGNINO, 2013)
- **Implementación del Sistema Web para la Gestión de Citas Horarias en el Hospital María Auxiliadora**, Lima Perú, 2017. Tiene como objetivo Implementar un sistema web para mejorar la gestión de citas Horarias en el Hospital María Auxiliadora. El modelo de software que se

empleó para la elaboración de la aplicación fue el modelo RUP y Metodología XP para la programación del Sistema, el cual permitió la recopilación de información para definir los requisitos y la arquitectura del sistema. (VALDIVIA, 2017).

- **Sistema Informático de Administración de Pacientes y Control de Citas e Inventario para la Clínica del ISTA.** Con este sistema informático se genera un expediente clínico de los pacientes ya sean empleados o beneficiarios, lo cual permite el registro de cada consulta médica-odontológica que se les brinde, así como los insumos utilizados, además de generar citas médicas para mejorar el proceso de atención de pacientes. (Carlos Migueo López Leon, 2016)

Antecedentes Nacionales

- **Sistema de Administración y Control de Historiales Clínicos para los Consultorios de la UMSA.** El proyecto tiene como objetivo almacenar los datos relevantes del paciente (universitario/a). Ha sido desarrollado para los consultorios dependientes del Departamento de Bienestar Social de la Universidad Mayor de San Andrés, cuya actividad principal es brindar atención médica eficaz y eficiente. (Flores, 2014)
- **Sistema de Gestión Hospitalario Modulo Consulta Externa Caso: Seguro Social Universitario (Universidad Mayor de San Andrés)** Presenta una propuesta para el Seguro Social Universitario para sistematizar la información sobre los historiales clínicos de los pacientes obteniendo de esa manera un mayor control de tratamiento de enfermedades y, como resultado proporcionar mejores servicios. (Nina, 2013)

- **Sistema Web de Registro y Seguimiento de Pacientes, Elaboración y Emisión de Análisis Efectuados, Registro y Guía Médica, Farmacéutica y Hospitalaria Caso: Laboratorio Clínico – Adolfo Kolping**, La Paz Bolivia, 2014. Se propone la creación de un sistema en plataforma Web, que facilite el manejo administrativo de pacientes y mejore los servicios que se prestan en el ámbito médico. (LLipe, 2014)

1.3 PLANTEAMIENTO DEL PROBLEMA

1.3.1 Problema Principal

Actualmente el Centro Médico “San Simón” realiza el registro de diagnósticos y tratamientos de forma manual, lo que conlleva una demora en la entrega de resultados a los pacientes.

1.3.2 Problemas Secundarios

- Los registros no se encuentran centralizados, lo cual dificulta llevar a cabo un control adecuado de la información.
- Los datos de los pacientes se encuentran registrados en archivos y folders, dando lugar a que se pierda la información, se produzca una inapropiada manipulación y se dificulte su acceso y distribución.
- El historial de los tratamientos de cada paciente se lleva de forma desordenada en diferentes folders comunes, dificultándose la obtención de información laboral del personal médico y administrativo.
- Las recetas son realizadas de forma manuscrita en cual genera desconfianza.
- Falta de información acerca de pacientes (no existe fácil acceso a los datos de un paciente)

Frente a los problemas mencionados surge la siguiente interrogante:

¿De qué manera se podrá mejorar el flujo de información del seguimiento de diagnóstico y tratamiento de pacientes del Centro Médico San Simón?

1.4 OBJETIVOS

1.4.1 Objetivo General

Desarrollar e implementar un sistema web que coadyuve a mejorar el flujo de información de seguimiento de diagnóstico y tratamiento de pacientes del Centro Médico San Simón de la ciudad de El Alto.

1.4.2 Objetivos Específicos

- Analizar la situación actual de los procesos en la administración de consultas médicas del Centro Médico San Simón para conocer las necesidades que se presentan en el Centro Médico.
- Almacenar la información de los pacientes del Centro Médico San Simón, evitando su pérdida.
- Centralizar la información de los pacientes del Centro Médico San Simón en una base de datos con soporte informático que posibilite un control adecuado de la información.
- Desarrollar un módulo de historial de diagnósticos de cada paciente.
- Desarrollar un módulo para el seguimiento de los tratamientos de los pacientes.
- Agilizar las tareas de búsqueda de información de los pacientes del Centro médico “San Simón” de la ciudad de El Alto.
- Diseñar reportes específicos requeridos por el personal médico y administrativo.

1.5 JUSTIFICACIÓN

1.5.1 Justificación Técnica

El desarrollo del Sistema Web para el Seguimiento de Diagnóstico y Tratamiento de Pacientes para el Centro Médico “San Simón”, se justifica porque la institución cuenta con los recursos informáticos necesarios para su implementación. El sistema se desarrollará en un lenguaje de código abierto usando el framework Laravel y un gestor de base de datos MySQL.

1.5.2 Justificación Económica

Con la implementación del Sistema Web para el Seguimiento de Diagnóstico y Tratamiento de Pacientes se obtendrá información confiable, lo que implica que, no existirá pérdida de información, también se harán uso de tecnologías y software libres.

La reducción de tiempo implica obtener el diagnóstico de una manera más ágil optimizando e incrementando así las ganancias en términos monetarios del centro médico.

El desarrollo del sistema Web coadyuvará a reducir el tiempo de atención y a brindar una atención de calidad a los pacientes, reduciendo gastos en material (formularios para historiales, folders, libros de consultas y libros de enfermería), con lo que se ahorrará en costos.

1.5.3 Justificación Social

El implementar el sistema web permite mejorar las tareas que se realizan, referentes a los diagnósticos, los tratamientos y el manejo de datos personales de los pacientes, porque proveerá información rápida y actualizada al médico a cargo, ofreciendo un mejor entorno de trabajo, comodidad laboral y mejorando la calidad de servicio.

En cuanto a los pacientes, se coadyuva a incrementar su satisfacción por la atención por parte del centro médico. Este proyecto se justifica socialmente porque coadyuvará a brindar una atención mejorada, adecuada y sistematizada para los pacientes que vienen a consulta.

1.6 METODOLOGÍA

Para el desarrollo del proyecto se optado por utilizar la UWE, la que se describe a continuación.

1.6.1 Metodología de Desarrollo UWE

UWE es una propuesta basada en UML (Lenguaje de Modelado Unificado) y en el proceso unificado para modelar aplicaciones web. Esta propuesta está formada por una notación para especificar el dominio (basada en UML) y un modelo para llevar a cabo el desarrollo del proceso de modelado.

Fases:

Las fases que conforman la metodología UWE, se especifican a continuación:

- 1) **Captura, análisis y especificación de requisitos:** Durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación web.
- 2) **Diseño del sistema:** Se basa en la especificación de requisitos producido por el análisis de los requerimientos, el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación Web.
- 3) **Codificación del software:** Se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.
- 4) **Pruebas:** Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código.
- 5) **La Instalación o Fase de Implementación:** Proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino.
- 6) **El Mantenimiento:** Es el proceso de control, mejora y optimización del software ya desarrollado e instalado.

UWE propone, a su vez, una extensión de UML, consistente en las actividades descritas a continuación:

Análisis de requisitos: Su objetivo es encontrar los requisitos funcionales de la aplicación Web para representarlos como casos de uso.

Diseño conceptual: Se refiere a construir un modelo conceptual del dominio de la aplicación, considerando los requisitos reflejados en los casos de uso. Da como resultado un diagrama de clases de dominio.

Diseño de navegaciones: Como resultado de este diseño, se obtiene el modelo de espacio de navegación y el modelo de estructura de navegación. Se producen los diagramas de clases.

Diseño de presentación: De este paso se obtienen una serie de vistas de interfaz de usuario que se presentan mediante diagramas de interacción. (German Sánchez, 2017)

ACTIVIDADES DEL MODELADO

Las actividades del modelado mediante UML, implica elaborar los siguientes modelos:

- Modelo de Casos de Uso
- Modelo de Contenido
- Modelo de Usuario
- Modelo de Estructura
- Modelo Abstracto
- Modelo de Adaptación

1.6.2 Métodos

Las técnicas de recolección de información a utilizar durante el proceso investigativo, se describen a continuación.

Observación

La observación como método, consiste en la utilización de los sentidos, para obtener de forma consciente y dirigida, datos que brindan elementos para la investigación. Constituye el primer paso del método científico, que permite, a partir de ello, elaborar una hipótesis, y luego vuelve a aplicarse la observación, para verificar si dicha hipótesis se cumple.

Entrevista

La entrevista constituye un instrumento de comunicación y una técnica de investigación fundamental en las ciencias humanas. Se emplea en la investigación social (recopilación de datos) y en tareas profesionales con diversos fines. Es el proceso de relación que se da en el encuentro de dos personas, entrevistador y entrevistado. Su finalidad puede ser investigativa, de asesoramiento educativo, preventivo, de desarrollo vocacional, de crecimiento personal, informativo o, también evaluativa. (Leopoldo De Los Ríos, 2014)

1.7 HERRAMIENTAS

Para la realización del proyecto se vio por conveniente usar software libre que se adapte a los requerimientos del centro médico. Entre las herramientas de desarrollo de software a utilizar, se consideran las siguientes:

- **APACHE:** Es un servidor Web HTTP de código abierto para plataformas Unix-like (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. (GDR Soluciones, 2017).

Apache es un servidor HTTP que permite servir contenido a las peticiones que vienen desde los clientes Web (navegadores).

Entre las principales características de Apache, se encuentran las siguientes:

- Es gratuito y de código abierto.
- Su instalación y configuración es sencilla.
- Es altamente extensible y adaptable mediante módulos.
- Dispone de funciones incorporadas para autenticación y validación de usuarios.
- Cuenta con soporte para lenguajes como Perl, PHP y Python. (Álvaro De León, 2019).

- **MariaDB:** Es un sistema de gestión de base de datos con licencia GPL, derivado como fork o bifurcación de MySQL, líder tradicional en su segmento (Dans, 2013).

El API y protocolo de MariaDB es compatible con los que usa MySQL, más algunas características para soportar operaciones no bloqueantes nativas y reportes de progreso. Esto implica que todos los conectores, librerías y aplicaciones que funcionan con MySQL también deberían hacerlo con MariaDB. (Salas, 2020).

- **PHP:** (acrónimo recursivo de PHP: *Hypertext Preprocessor*): Es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo Web y que puede ser incrustado en HTML. (The PHP Group, 2018).

Todos los códigos PHP se procesan en el lado del servidor. Esto significa que los scripts PHP se ejecutan directamente, a diferencia de otros códigos de programación como JavaScript, que se ejecutan del lado de quien solicita el acceso. Cuando un usuario crea una solicitud ante un servidor, el respectivo script PHP es enviado al intérprete PHP del servidor. Este procesa la secuencia de comandos y la envía al navegador. Normalmente se trata de documentos HTML, pero en ocasiones también de otro tipo de archivos. PHP permite, además, generar y enviar archivos JPG o PDF al servidor.

- **XAMPP:** Es un servidor independiente multiplataforma, de software libre, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl. (Zepeda, 2015).

- **Composer:** Es un gestor de dependencias en proyectos, para programación en PHP. Eso quiere decir que permite gestionar (declarar, descargar y mantener actualizados) los paquetes de software en los que se basa el proyecto PHP. (Alvarez, 2014).

- **Laravel:** Es un framework de código abierto para desarrollar aplicaciones y servicios Web con PHP 5 y PHP 7. Intenta aprovechar lo mejor de otros frameworks y aprovechar las características de las últimas versiones de PHP. (Andrés, 2014).

Laravel pone énfasis en la calidad del código, la facilidad de mantenimiento y escalabilidad, lo que permite realizar proyectos desde pequeños a grandes o muy grandes. Además permite y facilita el trabajo en equipo y promueve las mejores prácticas.

- **Bootstrap:** Es un framework originalmente, que permite crear interfaces web con CSS y JavaScript, cuya particularidad es la de adaptar la interfaz del sitio Web al tamaño del dispositivo en que se visualice. Es decir, el sitio web se adapta automáticamente al tamaño de una PC, una Tablet u otro dispositivo. Esta técnica de diseño y desarrollo se conoce como “responsive design” o diseño adaptativo. (Acedo, 2015).
- **Visual Studio Code:** Es un editor de código fuente ligero pero potente que se ejecuta en su escritorio y está disponible para Windows, macOS y Linux. Viene con soporte incorporado para JavaScript, TypeScript y Node.js y tiene un rico ecosistema de extensiones para otros lenguajes como C++, C#, Java, Python, PHP, GO y otros. (Microsoft, 2018).

1.8 LÍMITES Y ALCANCES

En el desarrollo del presente proyecto se considera la situación actual de las operaciones dentro del Centro Médico “San Simón”, de ahí que se identifican los siguientes límites y alcances.

1.8.1 Límites

De acuerdo a los requerimientos de la institución, el presente trabajo se limita a:

- El sistema será manipulado solo por el personal médico y administrativo con sus respectivos roles de usuarios.
- El sistema no realizará el control de cajas y el ámbito contable dentro del centro médico.
- El sistema no efectuará el control de fichas y reservas de las citas médicas.

1.8.2 Alcances

El presente proyecto tiene los siguientes alcances, los cuales se instrumentan mediante el desarrollo de los siguientes módulos:

- Módulo de administración: Destinado a efectuar la gestión de usuarios al sistema.
- Módulo de registro de nuevos pacientes.
- El sistema contará con un panel administrativo y personalizado para el médico, según su rol en el sistema.
- Contará con módulo seguimiento del historial de tratamientos en los pacientes.
- Contará con registro de actividades en cuanto a diagnóstico y tratamiento, según el examen realizado por el médico.

1.9 APORTES

Los aportes que ofrecerá este proyecto será computarizar los procesos rutinarios como el registro de diagnósticos y el tratamiento, optimizando tiempos en el registro de información médico, que coadyuve a la fácil y correcta toma de decisiones del Centro Médico San Simón

La búsqueda de datos, será de manera rápida ya que el medico podrá tener acceso a la información en tiempo real.

Se aplicará los conocimientos adquiridos en ingeniería de software para el desarrollo del sistema Web.

2 MARCO TEÓRICO

2.1 INTRODUCCIÓN

En el presente capítulo se desarrolla el sustrato teórico indispensable en que se sustenta el proyecto de grado, donde los elementos teóricos están extraídos de varias fuentes por lo tanto constituyen la base para la descripción y explicación del problema planteado, como también las herramientas de desarrollo de software son definidos con una descripción detallada de cada uno de los elementos de la teoría que serán directamente utilizados en el desarrollo del proyecto.

2.2 SISTEMA

Es un conjunto ordenado de componentes o elementos interrelacionados, interdependientes e interactuantes, que tienen por finalidad el logro de objetivos determinados en un plan (Cajizo, 1975).

Un sistema informático puede ser definido como un sistema de información que basa la parte fundamental de su procesamiento, en el empleo de la computación. Como cualquier sistema, es un conjunto de funciones interrelacionadas, hardware, software y de recursos humanos. Un sistema informático convencional emplea un sistema que usa dispositivos que se utilizan para programar y almacenar programas y datos. (Encinoza, 2008)

2.3 WEB

Web es un vocablo inglés que significa “red”, “telaraña” o “malla”. El concepto se utiliza en el ámbito tecnológico para nombrar a una red informática y, en general, a Internet (en este caso, suele escribirse como Web, con la W mayúscula).

El término, de todas formas, tiene varios usos. Además de nombrar a Internet en general, la palabra web puede servir hacer mención a una página web, un sitio web o hasta un servidor web.

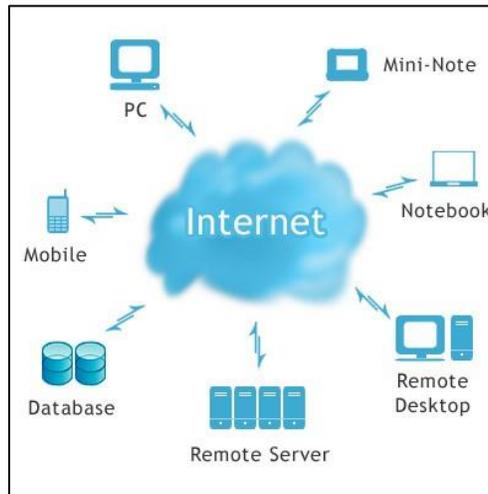
Es importante establecer que este término además forma parte de lo que se conoce como World Wide Web que es la red informática que se emplea en todo el mundo. A finales de la década de los años 80 fue cuando nació aquella que permite que hoy cualquier ciudadano, a través de una conexión a Internet junto a un navegador y un ordenador, pueda acceder desde cualquier rincón del mundo a la web que desea. Lo puede hacer para informarse, para entretenerse o simplemente por simple curiosidad. (Pérez, Merino, 2013)

2.4 SISTEMA WEB

Los "Sistemas Web" o también conocidos como "Aplicaciones Web" son aquellos que están creados e instalados no sobre una plataforma o sistemas operativos, sino que se alojan en un servidor en Internet o sobre una intranet (red local). Su aspecto es muy similar a páginas Web que se ven normalmente, pero en realidad los Sistemas Web tienen funcionalidades muy potentes que brindan respuestas a casos particulares. (Knowdo, 2012).

Los sistemas Web se pueden utilizar en cualquier navegador Web sin importar el sistema operativo. Para utilizar las aplicaciones Web no es necesario instalarlas en cada computadora ya que los usuarios se conectan a un servidor donde se aloja el sistema. Además si se trata de un sistema Web, involucra la construcción de una pieza de software con bastantes procesos, existen más puntos a tomar en cuenta, como, la configuración de servidores, el diseño e implementación de bases de datos, la seguridad de los datos, entre otros.

Figura 2. 1
Funcionamiento de internet y Petición



Fuente (Knowdo, 2012).

2.5 DIAGNÓSTICO Y TRATAMIENTO DE PACIENTES

El diagnóstico es la primera y más importante herramienta con la que cuenta un profesional de la salud de cualquier área para acercarse a la comprensión y posible tratamiento de las condiciones salubres de un individuo. El diagnóstico es el resultado del análisis que se realiza en una primera instancia y que tiene como fin permitir conocer las características específicas de la situación determinada para así poder actuar en consecuencia, sugiriendo tratamiento o no. Ese análisis diagnóstico se basa en la observación de síntomas existentes en el presente o en el pasado. (Bembibre, 2009)

2.5.1 SEGUIMIENTO

Es la acción y efecto de seguir o seguirse, en el contexto popular suele usarse como sinónimo de persecución, observación o vigilancia. Siendo este mismo usado principalmente en el contexto de investigaciones policiales, detectivescas, jurídicas, medicas, científicas, estadística, entre otras; para observar y analizar la evolución de un determinado caso. Aunque el término puede aplicarse a cualquier investigación, proceso o proyecto con observación constante. (Yirda, 2021).

2.6 INGENIERÍA DE SOFTWARE

La Ingeniería de Software es una disciplina o área de la Informática que comprende todos los aspectos de la producción de software desde las etapas iniciales de la especificación del sistema, hasta el mantenimiento de este después de que se utiliza. Sin embargo, aunque la ingeniería consiste en seleccionar el método más apropiado para un conjunto de circunstancias, un enfoque más informal y creativo de desarrollo podría ser efectivo en algunas circunstancias. El desarrollo informal es apropiado para el desarrollo de sistemas basados en Web, los cuales requieren una mezcla de técnicas de software y de diseño gráfico. (Sommerville, 2011, pág. 7).

La ingeniería de software es el establecimiento y uso de principios fundamentales de la ingeniería con objeto de desarrollar en forma económica software que sea confiable y que trabaje con eficiencia en máquinas reales. La Ingeniería del Software trata con áreas muy diversas de la Informática y de las ciencias de la computación, tales como construcción de compiladores, sistemas operativos o desarrollos de Intranet/Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información y aplicables a una infinidad de áreas tales como: negocios, investigación científica, medicina, producción, logística, banca, control de tráfico, meteorología, el mundo del derecho, la red de redes Internet, redes Intranet y Extranet, etc. (Pressman, 2010, pág. 11).

Por lo tanto, la ingeniería de software viene a ser un conjunto de métodos, herramientas y técnicas que se utilizan al momento de desarrollar un sistema informático.

2.6.1 Modelos de desarrollo de Software

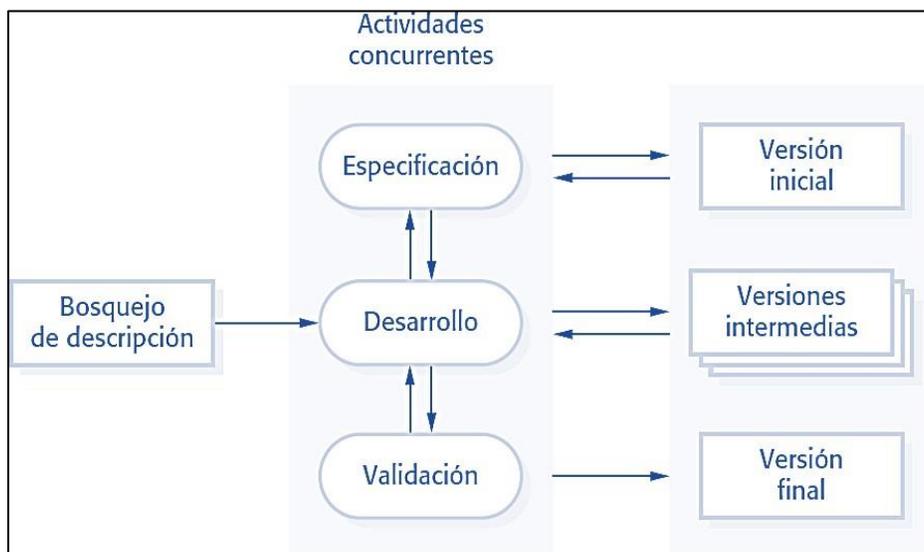
Un modelo de proceso de software es una representación simplificada de este proceso. Cada modelo del proceso representa a otro desde una particular perspectiva y, por lo tanto, ofrece sólo información parcial acerca de dicho proceso. Por ejemplo, un modelo de actividad del proceso muestra las actividades y su secuencia, pero quizá sin presentarlos roles de las personas que intervienen en esas actividades.

De los diferentes modelos de desarrollo de software, por las características del proyecto, resultan de interés los modelos de desarrollo incremental y los métodos de desarrollo ágil, que se describen a continuación.

- **Modelo de desarrollo Incremental**

El desarrollo incremental se basa en la idea de diseñar una implementación inicial, exponer ésta al comentario del usuario, y luego desarrollarla en sus diversas versiones hasta producir un sistema adecuado. Ver la Figura 2.2.

Figura 2. 2
Modelo de Desarrollo Incremental



Fuente: (Sommerville, 2011)

El desarrollo de software incremental, que es una parte fundamental de los enfoques ágiles, es mejor que un enfoque en cascada para la mayoría de los sistemas empresariales, de comercio electrónico y personales. El desarrollo incremental refleja la forma en que se resuelven problemas. Rara vez se trabaja por adelantado una solución completa del problema, más bien se avanza en una serie de pasos hacia una solución y se retrocede cuando se detecta que se cometieron errores. Al desarrollar el software de manera incremental, resulta más barato y fácil realizar cambios en el software conforme éste se diseña.

Cada incremento o versión del sistema incorpora algunas de las funciones que necesita el cliente. Por lo general, los primeros incrementos del sistema incluyen la función más importante o la más urgente. Esto significa que el cliente puede evaluar el desarrollo del sistema en una etapa relativamente temprana, para constatar si se entrega lo que se requiere. En caso contrario, sólo el incremento actual debe cambiarse y, posiblemente, definir una nueva función para incrementos posteriores. (Sommerville, 2011, pág. 33)

- **Métodos de desarrollo Ágil**

Los métodos ágiles son métodos de desarrollo incremental donde los incrementos son mínimos y, por lo general, se crean las nuevas liberaciones del sistema, y cada dos o tres semanas se ponen a disposición de los clientes. Involucran a los clientes en el proceso de desarrollo para conseguir una rápida retroalimentación sobre los requerimientos cambiantes. Minimizan la cantidad de documentación con el uso de comunicaciones informales, en vez de reuniones formales con documentos escritos.

Los enfoques ágiles en el desarrollo de software consideran el diseño y la implementación como las actividades centrales en el proceso del software. Incorporan otras actividades en el diseño y la implementación, como la adquisición de requerimientos y pruebas. Ver Figura 2.3.

Figura 2. 3
Modelo de desarrollo Ágil



Fuente: (Sommerville, 2011)

En contraste, un enfoque ágil basado en un plan para la ingeniería de software identifica etapas separadas en el proceso de software con salidas asociadas a cada etapa. Las salidas de una etapa se usan como base para planear la siguiente actividad del proceso, como puede observarse en la Figura 2.4. (Sommerville, 2011, pág. 63).

Figura 2. 4
Modelo de Desarrollo Ágil basado en un Plan



Fuente: (Sommerville, 2011)

2.6.2 MÉTRICAS DE CALIDAD

Las métricas de calidad de software son un conjunto de medidas utilizadas para estimar la calidad de un proyecto a desarrollar, entre otros conceptos, y que permiten comparar o planificar estas aplicaciones. Si no se mide, no hay una forma real de determinar si se está mejorando y si no se está mejorando, se está perdido. (Barrientos, 2018).

2.6.2.1 ISO/IEC 9126

La ISO 9126 es un estándar internacional para evaluar la calidad del software en base a un conjunto de características y sub-características de la calidad. Cada sub-característica consta de un conjunto de atributos que son medidos por una serie de métricas. (Prieto Medina, 2017).

El estándar ISO 9126 se desarrolló con la intención de identificar los atributos clave del software de cómputo. Este sistema identifica seis atributos clave de la calidad, que describe a continuación.

2.6.2.2 Funcionalidad

Grado en el que el software satisface las necesidades planteadas según las establecen los atributos siguientes: adaptabilidad, exactitud, interoperabilidad, cumplimiento y seguridad.

Los puntos de función (PF) se describen como medidas básicas desde donde se calculan métricas de productividad. Los datos de PF se utilizan de dos formas durante la estimación del proyecto software:

- Como una variable de estimación que se utiliza para dimensionar cada elemento del software.
- Como métricas de línea base recopilada de proyectos anteriores, y utilizados junto con variables de estimación para desarrollar proyecciones de costo y esfuerzo.

Para estimaciones de PF, la descomposición funciona de la siguiente manera:

Tabla 2. 1
Dominios de información de Punto Función.

DOMINIO DE INFORMACIÓN	DESCRIPCIÓN
Número de entradas de usuario	Una petición se define como una entrada interactiva que produce la generación de alguna respuesta del software inmediata en forma de salida interactiva. Se cuenta cada petición por separado. Dentro de un informe no se cuentan de forma separada.
Numero de archivos	Se cuenta cada archivo maestro lógico (esto es, un grupo lógico de datos que puede ser una parte de una gran base de datos o un archivo independiente).
Número de interfaces externas	Se cuenta todas las interfaces legibles por la maquina (por ejemplo: archivos de datos de disco) que se utilizan para transmitir información a otros sistemas.

Número de salidas de usuario

Se cuenta cada salida que proporciona al usuario o información orientada a la aplicación. En este contexto la salida se refiere a informes, pantallas, mensajes de error, y demás. Los elementos de datos particulares.

Fuente: (Pressman, 2010)

Los puntos de función se calculan completando la Tabla 2.2.

Tabla 2. 2
Cálculos de los Punto Función.

PARÁMETROS DE MEDICIÓN	FACTOR DE PONDERACIÓN				
	CUENTA	SIMPLE	MEDIO	COMPLEJO	RESULTADO
Número de entradas de Usuario	N1	3	4	6	N1 *factor
Número de salidas de usuario.	N1	4	5	7	N2 *factor
Número de peticiones de usuario.	N1	3	4	6	N3 *factor
Numero de archivos.	N1	7	10	15	N4 *factor
Numero de interfaces externas.	N1	5	7	10	N5 *factor
CUENTA TOTAL					$\Sigma (Ni*factor)$

Fuente: (Pressman, 2010, pág. 532)

Para calcular los PF, se utiliza la relación siguiente:

$$PF = Cuenta_{Total} \times [0.65 + 0.01 \times \sum Fi]$$

CUENTA TOTAL = Sumatoria de todas las entradas de la Tabla 2.2.

Donde CUENTA TOTAL es la suma de todas las entradas de PF obtenidas de la tabla anterior. El valor obtenido de esta relación es sometido a la siguiente comparación:

300 < PF – Optimo
200 < PF <= 300 – Bueno
100 < PF <= 200 – Suficiente
PF <= 100 – Deficiente

Fi = Son valores de ajuste a la complejidad según las respuestas a las preguntas siguientes.

1. ¿Requiere el sistema copias de seguridad y recuperación flexible?
2. ¿Se requiere comunicación de datos?
3. ¿Existen funciones del procedimiento distribuido?
4. ¿Es crítico el rendimiento?
5. ¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?
6. ¿Requiere el sistema entrada de datos interactiva?
7. ¿Requiere la entrada de datos interactiva que las transiciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?
8. ¿Se actualiza los archivos maestros de forma interactiva?
9. ¿Son complejas las entradas, las salidas, los archivos y las peticiones?
10. ¿Es complejo el procesamiento interno?
11. ¿Se ha diseñado el código para ser reutilizable?
12. ¿Están concluidas en el diseño la conversión y la instalación?
13. ¿Se ha desarrollado el sistema para soportar múltiples instalaciones en diferentes organizaciones?
14. ¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizado por el usuario?

Cada una de las preguntas, es respondida usando una escala con rangos desde 0 (no importante), hasta 5 (absolutamente esencial).

2.6.2.3 Mantenibilidad

La capacidad de mantenimiento es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales.

Formula:

$$IMS = \frac{[MT - (Fc + Fa + Fe)]}{MT}$$

Dónde:

MT = Número de módulos en la versión actual.

Fc = Número de módulos en la versión actual que se han cambiado

Fa = Número de módulos en la versión actual que se han añadido

Fe = Número de módulos en la versión actual que se han eliminado

A medida que el sistema se aproxima a 1 el producto se pone más estable.

75% <= IMS <= 100% -- Optima

50% <= IMS <= 75% -- Buena

25% <= IMS <= 50% -- Suficiente

0% <= IMS <= 25% -- Deficiente

2.6.2.4 Portabilidad

La portabilidad de un sistema de información, se define como la factibilidad de transferir un producto a diferentes entornos de hardware/software, sin necesidad de aplicar acciones o mecanismos distintos. También es considerado como la capacidad del producto software para ser usado en lugar de otro producto software, para el mismo propósito dentro del mismo entorno. Las características más importantes que se consideran para este factor son: la facilidad de instalación, facilidad de ajuste y adaptación al cambio (Piattini & Garcia, 2003).

Es la facilidad de transportar productos software a varios ambientes de hardware – software. Se mide probando el sistema en diferentes sistemas operativos. El criterio se subdivide en facilidad de instalación, facilidad de ajuste, facilidad de adaptación al cambio.

La portabilidad viene dada por:

$$P = 1 - \frac{EP}{EI}$$

Dónde:

P = Portabilidad

EP = Número de días para portar el sistema

EI = Número de días para implementar el sistema

Luego de obtener el resultado se hace una verificación con los siguientes valores:

75% <= IMS <= 100% -- Optima

50% <= IMS <= 75% -- Buena

25% <= IMS <= 50% -- Suficiente

0% <= IMS <= 25% -- Deficiente

2.6.2.5 Usabilidad

Se refiere a un conjunto de atributos relacionados con el esfuerzo necesario para su uso y a la valoración individual de tal uso (ver tabla 2.3.).

Tabla 2. 3
Preguntas para calcular la usabilidad

NRO.	PREGUNTAS	EVALUACIÓN (XI)
1	¿El sistema satisface los requerimientos de manejo de información?	X
2	¿Las salidas del sistema están de acuerdo a sus requerimientos?	X
3	¿Cómo considera el ingreso de datos del sistema?	X
4	¿Cómo considera los formularios que elabora el sistema?	X
5	¿El sistema facilita el trabajo que realiza?	X

Fuente: (Pressman, 2010)

La fórmula para calcular la facilidad de uso es:

$$FU = \frac{\sum X_i}{n} \times 100$$

2.6.2.6 Confiabilidad

Aquí se agrupan un conjunto de atributos que se refieren a la capacidad del software de mantener su nivel de ejecución bajo condiciones normales en un periodo de tiempo establecido. Las sub características que el estándar sugiere son:

- **Nivel de Madurez.** Permite medir la frecuencia de falla por errores en el software.
- **Tolerancia a fallas.** Se refiere a la habilidad de mantener un nivel específico de funcionamiento en caso de fallas del software o de cometer infracciones de su interfaz específica.
- **Recuperación.** Se refiere a la capacidad de restablecer el nivel de operación y recobrar los datos que hayan sido afectados directamente por una falla, así como al tiempo y el esfuerzo necesarios para lograrlo.

Los primeros trabajos sobre confiabilidad del software trataban de extrapolar la teoría matemática de la confiabilidad del hardware a la predicción de la confiabilidad del software. La mayor parte de modelos relacionados con el hardware se abocan a la falla debida al uso, en lugar de a la que tiene su origen en los defectos de diseño. En el hardware, las fallas debidas al uso físico (por ejemplo, los efectos de temperatura, corrosión y golpes) son más probables que las debidas al diseño. Desafortunadamente, con el software ocurre lo contrario. En realidad, todas las fallas del software pueden rastrearse en problemas de diseño o de implementación.

La confiabilidad del Software se mide con la siguiente formula:

$$R(t) = e^{-\lambda T}$$

Dónde:

R(t) = Confiabilidad del Sistema

λ = Error de tasa constante de fallas

T = Tiempo de operación del sistema (meses)

2.6.2.7 Eficiencia

Consiste en la relación entre el nivel de funcionamiento del software y la cantidad de recursos usados. Los aspectos a evaluar son:

- Comportamiento con respecto al Tiempo. Atributos del software relativos a los tiempos de respuesta y de procesamiento de los datos.
- Comportamiento con respecto a Recursos. Atributos del software relativos a la cantidad de recursos usados y la duración de su uso en la realización de sus funciones. (Abud Figueroa, 2012).

Consideramos la tabla 2.4 para calcular la eficacia:

Tabla 2. 4
Preguntas para Calcular la Eficiencia.

Nº	PREGUNTA	EVALUACIÓN
1	¿La respuesta es rápida al utilizar las funciones?	X
2	¿Tiene rendimiento de acuerdo a los factores que utiliza?	X
3	¿Responde adecuadamente cuando utiliza las funciones?	X
TOTAL		X

Fuente: (Pressman, 2010)

2.6.3 MÉTODO DE ESTIMACIÓN DE COSTOS DE SOFTWARE

Existen tres principales parámetros que se deben usar al calcular los costos de un proyecto de desarrollo de software:

- Costos de esfuerzo (los costos de pagar a los ingenieros y administradores desoftware).
- Costos de hardware y software, incluido el mantenimiento.
- Costos de viajes y capacitación.

Para la mayoría de los proyectos, el mayor costo es el primer rubro. Debe estimarse el esfuerzo total (en meses-hombre) que es probable se requiera para completar el trabajo de un proyecto. Desde luego, se cuenta con datos limitados para realizar tal valoración, de manera que habrá que hacer la mejor evaluación posible y a continuación agregar contingencia significativa (tiempo y esfuerzo adicionales) en caso de que la estimación inicial sea optimista.

2.6.3.1 Análisis de costo de software COCOMO

Constructive Cost Model del (COCOMO) es un algorítmico Modelo de la valoración del coste del software. El modelo utiliza una básica fórmula, con los parámetros que se derivan de datos históricos del proyecto y de características actuales del proyecto.

COCOMO consiste en una jerarquía de tres cada vez más detallados y de formas exactas. El primer nivel, COCOMO básico es buena la orden temprana, áspera de las estimaciones de la magnitud de los costes del software, pero su exactitud debe limitado a su carencia de diferencia en cualidades del proyecto (Conductores del coste). COCOMO intermedio toma estos conductores del coste en consideración y COCOMO detallado explica además la influencia de las fases del proyecto individual.

COCOMO es uno de los sistemas de estimación de costes más utilizados en proyectos de desarrollo de software. La estandarización de su uso y la facilidad de la aplicación del mismo junto con la aproximación al coste real, han convertido a este modelo en uno de los referentes en este tipo de proyectos. (Calero, 2010).

Modelos

- COCOMO básico. Calcula el esfuerzo y el costo del desarrollo en función del tamaño del programa estimado en LDC (líneas de código).
- COCOMO intermedio. Calcula el esfuerzo del desarrollo en función del tamaño del programa y un conjunto de conductores de costo que incluyen la evaluación subjetiva del producto, del hardware, del personal y de los atributos del proyecto.

- COCOMO detallado. Incorpora las características de la versión intermedia y lleva a cabo una evaluación del impacto de los conductores de costo en cada fase (análisis, desarrollo, etc.) del proceso.

2.6.3.2 Método de estimación COCOMO II

El modelo COCOMO original se convirtió en uno de los modelos de estimación de costo más ampliamente utilizados y estudiados en la industria. Evolucionó hacia un modelo de estimación más exhaustivo, llamado COCOMO II. Como su predecesor, COCOMO II en realidad es una jerarquía de modelos de estimación que aborda las áreas siguientes:

- Modelo de composición de aplicación. Se usa durante las primeras etapas de la ingeniería de software, cuando son primordiales la elaboración de prototipos de las interfaces de usuario, la consideración de la interacción del software y el sistema, la valoración del rendimiento y la evaluación de la madurez de la tecnología.
- Modelo de etapa temprana de diseño. Se usa una vez estabilizados los requisitos y establecida la arquitectura básica del software.
- Modelo de etapa post arquitectónica. Se usa durante la construcción del software.

Como todos los modelos de estimación para software, los modelos COCOMO II requieren información sobre dimensionamiento. Como parte de la jerarquía del modelo, están disponibles tres diferentes opciones de dimensionamiento: puntos objeto, puntos de función y líneas de código fuente.

Fórmula para hallar el factor de complejidad TCF

$$TCF = (0.65 + 0.01 \times PF)$$

El procesamiento de datos del punto función se basa en la fórmula siguiente:

$$PF = \text{Cuenta Total} \times TCF$$

El factor LDC/PF se calcula con la fórmula:

$$LDC = PF \times \text{Factor} \frac{LDC}{PF}$$

El número estimado de líneas de código distribuidas en miles se calcula con la siguiente fórmula:

$$KLCD = \frac{LDC}{1000}$$

Las ecuaciones del COCOMO básico tienen la siguiente forma:

$$E = a_b(KLCD)^{bb}$$
$$D = c_b D^{db}$$

Dónde:

E: Esfuerzo aplicado en personas por mes.

D: Tiempo de desarrollo en meses cronológicos.

KLDC: Número estimado de líneas de código distribuidas (en miles).

El número de programadores (Nº Prog) se obtiene con la siguiente fórmula:

$$N^{\circ} \text{ Prog} = \frac{E}{D}$$

Costo del software desarrollado por persona = Numero de programadores * salario de un programador.

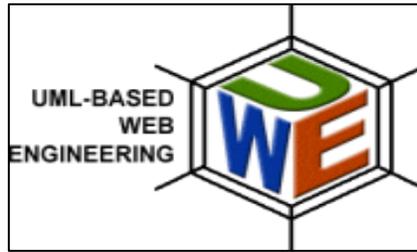
2.7 METODOLOGÍA DE DESARROLLO DEL SISTEMA WEB

Actualmente existen muchas metodologías para el desarrollo de software que son utilizadas dependiendo del sistema a crear, los cuales pueden dividirse en grupos comunes como son: escritorio, móvil y web, siendo este último de los que más impulso ha venido acumulando, pues la necesidad del cliente de ser reconocido por medio del Internet se ha intensificado. (Segarra,2017)

2.7.1 METODOLOGÍA UWE

UWE (UML-Based Web Engineering) ingeniería Web basada en UML es una metodología de desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web.

Figura 2. 5
Ingeniería web basada en UML



Fuente: (Galiano, 2013)

UWE es una propuesta basada en el proceso unificado y UML, pero adaptados a la web. En requisitos separa las fases de captura, definición y validación. Hace además una clasificación y un tratamiento especial dependiendo del carácter de cada requisito. Entre los principales modelos de UWE se puede citar: el modelo de requerimientos, el modelo lógico-conceptual, modelo navegacional y modelo de presentación. UWE define vistas especiales representadas gráficamente por diagramas en UML. Además, UWE no limita el número de vistas posibles de una aplicación, pues UML proporciona mecanismos de extensión basados en estereotipos. Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son lo que finalmente se utilizarán en las vistas especiales para el modelado de aplicaciones Web. De esta manera, se obtiene una notación UML adecuada a un dominio en específico a la cual se le conoce como Perfil UML. UWE está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario (Blanco, 2015).

2.7.2 Características de una aplicación Web

Las actividades UWE son procesos que se utilizan y permiten identificar necesidades de la aplicación o sistema web a desarrollar. Estas actividades se describen y representan en cuatro fases, presentadas a continuación:

Figura 2. 6
Actividades de la Metodología UWE

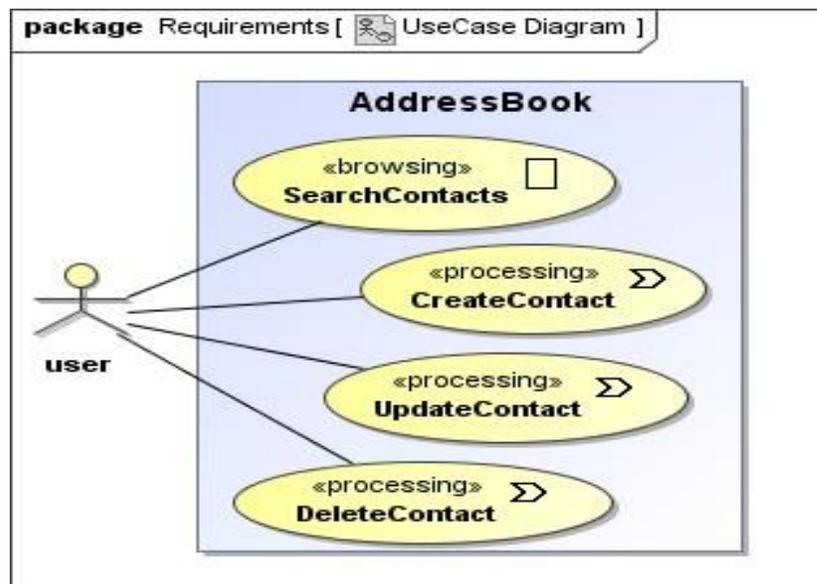


Fuente: (Elaboración propia)

2.7.2.1 Análisis de Requisitos

Como en otras metodologías, la primera fase o actividad es el análisis de requisitos funcionales, que permite visualizar los procesos y funciones que debe cumplir el software, se utilizan diagramas de casos de uso.

Figura 2. 7
Casos de usos

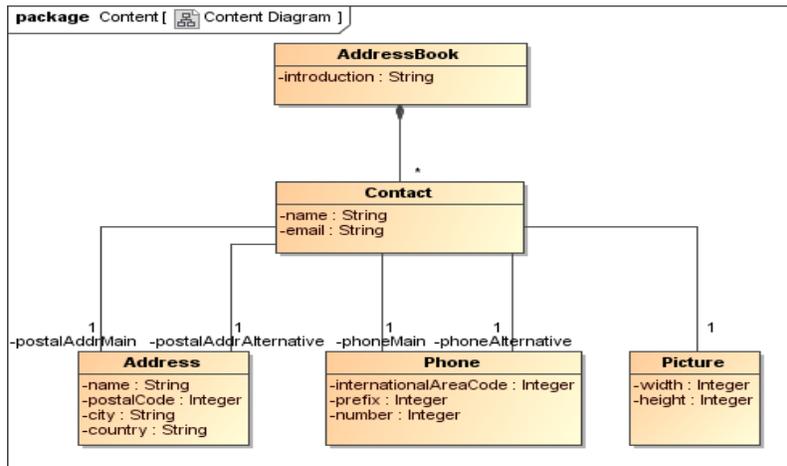


Fuente: (Maximilians, 2015)

2.7.2.2 Modelo Conceptual

El modelo conceptual se basa en el análisis y requisitos reflejados en los casos de uso, comprende el modelo de dominio que al igual que los casos de uso se debe cumplir con las funcionalidades requeridas por el sistema web a desarrollar, el diseño conceptual no sufre ningún cambio con el modelo o diagrama de clases correspondientes a UML.

Figura 2. 8
Análisis de casos de uso (Diagrama de Contenido)

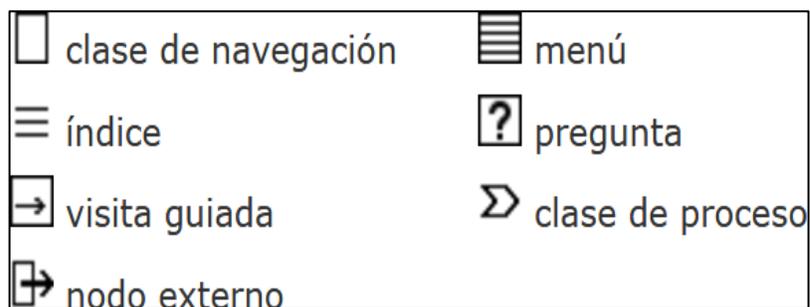


Fuente: (Maximilians, 2015)

2.7.2.3 Modelo Navegacional

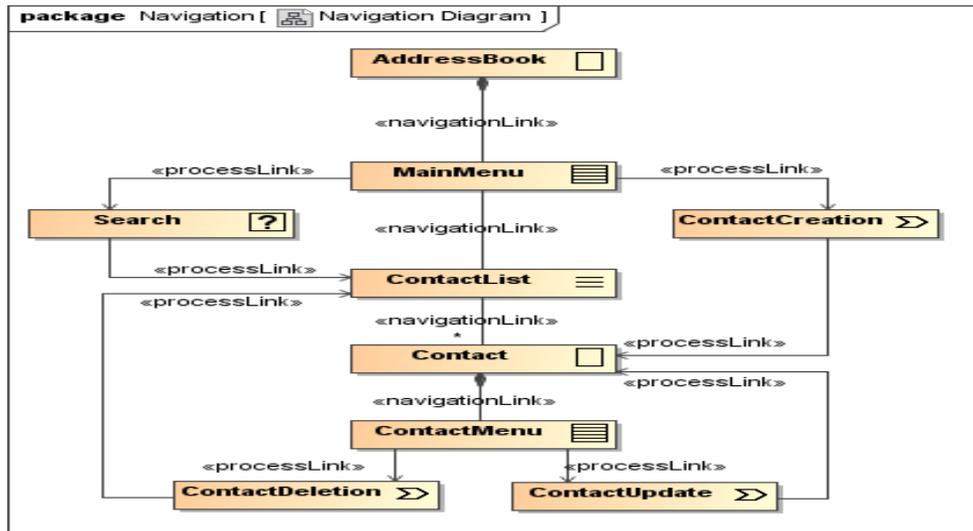
Cuando se habla del desarrollo de un sistema web, es necesario conocer la relación y enlaces entre las páginas web, es por eso que en la fase de diseño se describen a través de diagramas la navegación del sistema cumpliendo con lo que se diseñó en los casos de uso. Los elementos que se utilizan en el diagrama son:

Figura 2. 9
Elementos del Diseño Navegacional



Fuente: (Maximilians, 2015)

Figura 2. 10
Modelo Navegacional UWE

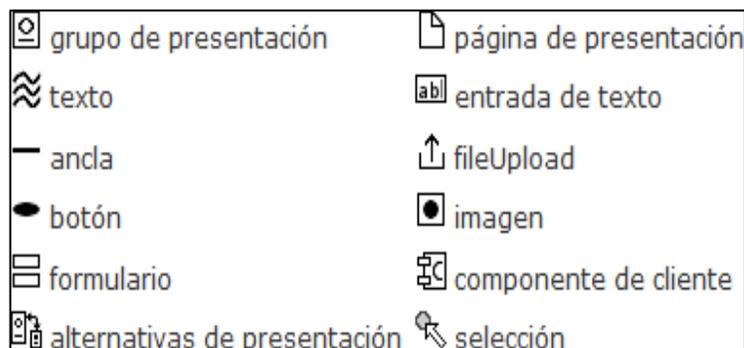


Fuente: (UWE, 2020)

2.7.2.4 Modelo de Presentación

El modelo de la presentación permite una visión amplia de los procesos de las páginas o aplicaciones web que se representan en los diagramas de navegación, también puede interpretarse con las interfaces del sistema o aplicación web. Para cada caso se tiene estereotipos o iconos que ayudan al diseño de los diagramas de presentación. Los iconos que permiten la realización de los diagramas de presentación poseen una característica y permite que los diagramas de presentación sean entendibles, como se muestran a continuación:

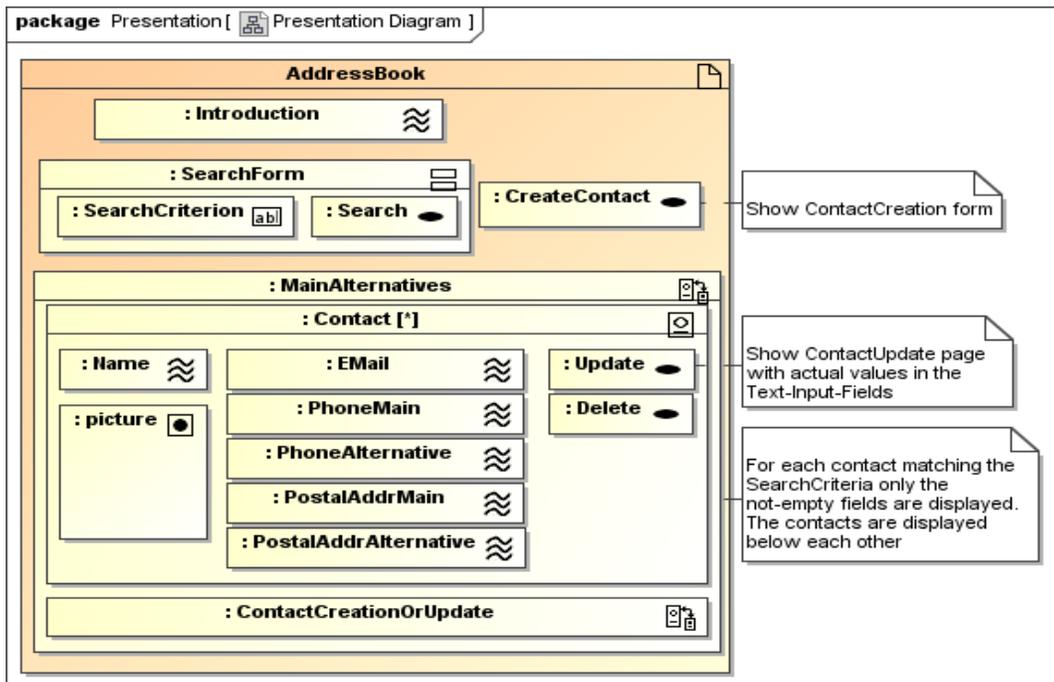
Figura 2. 11
Elementos de Diseño de Presentación



Fuente: (UWE, 2020)

Para el diseño de presentación, se debe tener en cuenta la funcionalidad que se requiere para el cumplimiento de los requerimientos del usuario. El diagrama de presentación de la metodología UWE, permite al usuario comprender y analizar, sobre el área de trabajo al que se someterá con la implementación del sistema. En la siguiente figura 2.12, se muestra la aplicación de los iconos que pertenecen a los diagramas de presentación:

Figura 2. 12
Diagrama de Presentación



Fuente: (UWE, 2020)

2.7.3 Características de la metodología UWE

La metodología UWE define vistas especiales representadas gráficamente por diagramas en UML, tales como el modelo de navegación y el modelo de presentación. Los diagramas se pueden adaptar como mecanismos de extensión basados en estereotipos que proporciona UML. Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son los que finalmente se utilizarán en las vistas especiales para el modelado de aplicaciones Web. De esta manera, se obtiene una notación UML adecuada para un dominio específico a la que se conoce como “Perfil UML” (Schwabe & Olsina, 2008).

Un perfil de UML consiste en una jerarquía de estereotipos y un conjunto de restricciones. Los estereotipos son utilizados para representar instancias de las clases. La ventaja de utilizar los perfiles de UML es que casi todas las herramientas CASE de UML los reconocen. Los modelos deben ser fácilmente adaptables al cambio en cualquier etapa del desarrollo.

2.7.4 Fases de la metodología UWE

Las fases o etapas que utiliza la metodología UWE son:

- a) Captura y análisis de requisitos.
- b) Diseño del sistema.
- c) Codificación del software.
- d) Pruebas.
- e) Fase de Implementación.
- f) El Mantenimiento.

2.7.4.1 Captura y análisis de requisitos

Básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación Web. Trata de diferente forma las necesidades de información. Centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.

2.7.4.2 Diseño del sistema

Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación Web.

2.7.4.3 Codificación del software

Durante esta etapa se realizan las tareas que comúnmente se conocen como programación que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.

2.7.4.4 Pruebas

Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código en este caso se usarán las pruebas de caja blanca y caja negra.

2.7.4.5 Fase de Implementación

Es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados y eventualmente, configurados todo ello con el propósito de ser ya utilizados por el usuario final.

Esto incluye la implementación de la arquitectura, de la estructura del hiperespacio, del modelo de usuario, de la interfaz de usuario, de los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones.

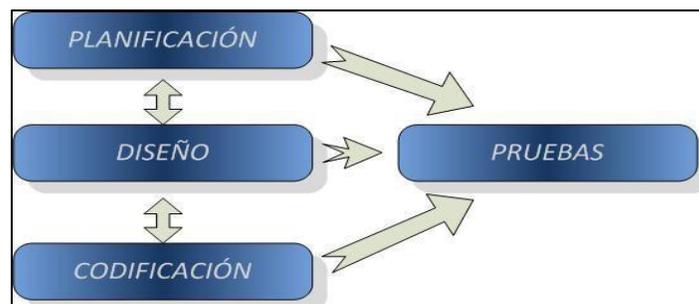
2.7.4.6 El Mantenimiento

Es el proceso de control, mejora y optimización del software ya desarrollado e instalado, que también incluye depuración de errores y defectos que puedan haberse filtrado de la fase de pruebas de control.

2.7.5 Ciclo de la metodología UWE

UWE cubre todo el ciclo de vida de este tipo de aplicaciones (Ver Figura 2.13) centrandose además su atención en aplicaciones personalizadas o adaptativas.

Figura 2. 13
Ciclo de la Metodología UWE



Fuente: (Galiano, 2013)

2.7.6 Características de una aplicación Web

Las Aplicaciones Web tienen una serie de rasgos comunes que las diferencia de otros tipos de aplicaciones software, y que son:

- Desde el punto de vista del usuario, se ha universalizado su accesibilidad: Actualmente un usuario experto y un usuario con habilidad limitada en el uso de aplicaciones informáticas acceden al mismo tipo de aplicación. Aún más, el número y tipo de usuario de las Aplicaciones Web no siempre es predecible, lo que obliga a tener el concepto de facilidad de uso aún más presente que en otros tipos de aplicaciones.
- Desde el punto de vista de la plataforma se realiza un uso intensivo de la red y la conexión se establece desde distintos tipos de dispositivo de acceso.
- Desde el punto de vista de la información, se asiste en la actualidad a una disponibilidad global de fuentes heterogéneas de información, estructurada y no estructurada, pertenecientes a distintos dominios y que colaboran en el cumplimiento de los objetivos de la aplicación.

2.7.7 Requisitos de Desarrollo de una Aplicación Web

Cada una de estas perspectivas introduce una serie de requisitos que deben ser tenidos en cuenta durante el proceso de desarrollo de cualquier tipo de Aplicación Web, con el fin de incrementar su probabilidad de éxito de implantación y que pueden ser estructuradas como sigue:

- Portabilidad. Debido a la dinamicidad del entorno tecnológico, a menudo es necesario implantar una misma aplicación en distintas plataformas, con distintas arquitecturas, con distintas tecnologías y/o atendiendo a distintos dispositivos de acceso, lo que obliga a desarrollar técnicas, modelos y herramientas que faciliten la reutilización e independiza hasta donde sea posible en el desarrollo de la aplicación.
- Inmediatez (Rapidez de Implantación). El desarrollo de aplicaciones web requiere un período de implantación mucho más reducido, que influye en todo su ciclo de desarrollo.

- Creación de contenidos como parte integrante de la fase de ingeniería de la aplicación. Aunque en este trabajo se caracteriza la especificación de aplicaciones orientadas a ofrecer funcionalidad compleja, más allá de la mera diseminación de información, el diseño y producción de textos, gráficos, vídeos etc., que conforman la estructura informacional de la aplicación es una tarea que debería ser realizada en paralelo al diseño de la propia aplicación.
- Integración (disponibilidad global) de fuentes heterogéneas de información. La posible necesidad de manejo integrado de contenido estructurado y no estructurado, almacenado en distintos formatos (bases de datos, sistemas de ficheros, dispositivos multimedia) y accesibles de forma distribuida mediante múltiples aplicaciones, es otro de los factores que condiciona el proceso de diseño de este tipo de aplicaciones.

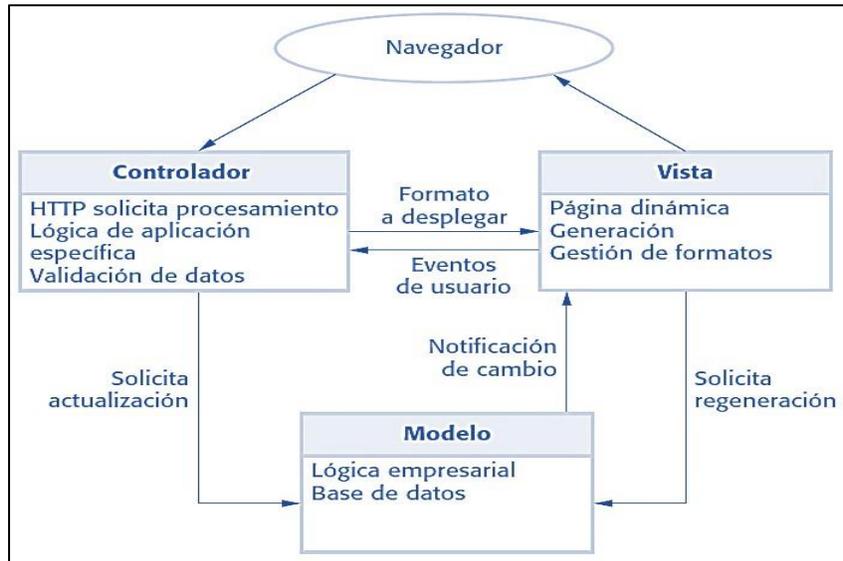
2.8 ARQUITECTURA DEL SOFTWARE

En los procesos ágiles, por lo general se acepta que una de las primeras etapas en el proceso de desarrollo debe preocuparse por establecer una arquitectura global del sistema. La arquitectura de software es importante porque afecta el desempeño y la potencia, así como la capacidad de distribución y mantenimiento de un sistema. (Sommerville, 2011)

2.8.1 Patrón Modelo Vista Controlador (MVC)

El patrón MVC es un patrón de diseño de arquitectura de software usado principalmente en aplicaciones que manejan gran cantidad de datos y transacciones complejas donde se requiere una mejor separación de conceptos para que el desarrollo esté estructurado de una mejor manera, facilitando la programación en diferentes capas de manera paralela e independiente. MVC sugiere la separación del software en tres estratos: Modelo, Vista y Controlador, los cuales serán explicados en la siguiente figura.

Figura 2. 14
Arquitectura de Aplicación Web con el Patrón MVC.



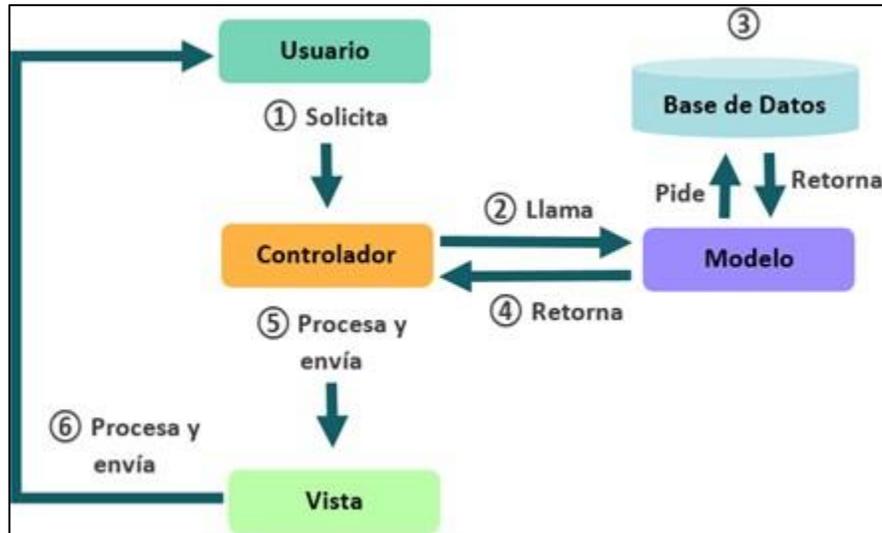
Fuente: (Sommerville, 2011, pág. 157)

- **Modelo:** Es la representación de la información que maneja la aplicación. El modelo en sí son los datos puros que puestos en contexto del sistema proveen de información al usuario o a la aplicación misma.
- **Vista:** Es la representación del modelo en forma gráfica disponible para la interacción con el usuario. En el caso de una aplicación Web, la “Vista” es una página HTML con contenido dinámico sobre el cuál el usuario puede realizar operaciones.
- **Controlador:** Es la capa encargada de manejar y responder las solicitudes del usuario, procesando la información necesaria y modificando el Modelo en caso de ser necesario.

2.8.2 Ciclo de Vida del patrón MVC

El ciclo de vida de MVC es normalmente representado por las tres capas presentadas anteriormente y el cliente (también conocido como usuario). El siguiente diagrama representa el ciclo de vida de manera sencilla:

Figura 2. 15
Ciclo de Vida del MVC



Fuente: (Gómez, 2015)

- El usuario realiza una petición.
- El controlador captura la petición del usuario.
- El controlador llama al modelo.
- El modelo interactúa con la base de datos, y retorna la información al controlador.
- El controlador recibe la información y la envía a la vista.
- La vista procesa la información recibida y la entrega de una manera visualmente entendible al usuario.

2.8.3 Ventajas y Desventajas del Patrón MVC

• Ventajas de MVC

Las principales ventajas del uso del patrón MVC son:

- La separación del Modelo y la Vista, lo cual logra separar los datos, de su representación visual.
- Facilita el manejo de errores.
- Permite que el sistema sea escalable si es requerido.
- Es posible agregar múltiples representaciones de los datos.

- **Desventajas de MVC**

Las principales desventajas del uso del patrón MVC son:

- La cantidad de archivos que se deben mantener incrementa considerablemente.
- La curva de aprendizaje es más alta que utilizando otros modelos.
- Su separación en capas, aumenta la complejidad del sistema.
- Puede implicar código adicional y complejidad de código cuando el modelo de datos y las interacciones son simples. (Sommerville, 2011).

2.9 HERRAMIENTAS

Las herramientas que se usaran en el desarrollo del Sistema Web son las siguientes:

2.9.1 Base de datos

- **MariaDB:** MariaDB es un reemplazo mejorado y directo para el servidor de bases de datos MySQL y está disponible bajo la licencia GPL v2. Es desarrollado por la comunidad MariaDB con MariaDB Fundación como su principal desarrollador.

El proyecto mantiene su propio conjunto de seguridad. Parches en la parte superior de MySQL. De hecho, muchos de los problemas de seguridad encontrados en MySQL y MariaDB han sido encontrados e informados por el equipo de MariaDB.

MariaDB se mantiene al día con la última versión de MySQL y en la mayoría de los aspectos MariaDB trabajará exactamente igual que MySQL. Todos los comandos, interfaces, bibliotecas y API que existen en MySQL también existen en MariaDB.

No es necesario convertir bases de datos para cambiar a MariaDB. MariaDB es una verdadero reemplazo de MySQL, además, MariaDB tiene muchas características nuevas y agradables que puede aprovechar. (MariaDB Foundation, 2009).

2.9.2 Frontend

- **HTML5:** HTML5 no es una nueva versión del antiguo lenguaje de etiquetas, ni siquiera una mejora de esta ya antigua tecnología, sino un nuevo concepto para la construcción de sitios web y aplicaciones en una era que combina dispositivos móviles, computación en la nube y trabajos en red. El limitado objetivo de HTML motivó a varias compañías a desarrollar nuevos lenguajes y programas para agregar características a la web nunca antes implementada. Estos desarrollos iniciales crecieron hasta convertirse en populares y poderosos accesorios. Simples juegos y bromas animadas pronto se transformaron en sofisticadas aplicaciones ofreciendo nuevas experiencias que cambiaron el concepto de la web para siempre.

JavaScript era claramente el lenguaje que permitía a los desarrolladores innovar y hacer cosas que nadie había podido hacer antes en la web. En los últimos años, programadores y diseñadores web alrededor del mundo surgieron con los más increíbles trucos para superar las limitaciones de esta tecnología y sus iniciales deficiencias en portabilidad. Gracias a estas nuevas implementaciones, Javascript, HTML y CSS se convirtieron pronto en la más perfecta combinación para la necesaria evolución de la web, HTML5 que propone estándares para cada aspecto de la web y también un propósito claro para cada una de las tecnologías involucradas. El futuro de la web es prometedor y la evolución y combinación de estas tres tecnologías (HTML, CSS y Javascript) en una poderosa especificación está volviendo a Internet la plataforma líder de desarrollo. (Gauchat, 2012).

De acuerdo a lo citado anteriormente se puede concluir que HTML 5, no es la continuación de HTML, es un lenguaje independiente capaz de combinar dispositivos móviles e incluso trabajar en la red, surgió principalmente porque la versión de HTML era muy limitada surgiendo así la idea de varias compañías de crear programas que permitan mejorar características de la web siendo las primeras propuestas juegos animados y más tarde las mejores propuestas fueron Java y Flash pero decayeron por no estar integrados, luego JavaScript permitía a los desarrolladores innovar y hacer

cosas que nadie había podido hacer antes en la web, y así conjuntamente con CSS y HTML se convirtieron en la perfecta combinación y por ende HTML5 es, de hecho, una mejora de esta combinación.

- **CSS:** El CSS (Hojas de estilo en Cascada) es un lenguaje de estilos empleado para definir la presentación, el formato y la apariencia de un documento de marcaje, sea html, xml, o cualquier otro. Comúnmente se emplea para dar formato visual a documentos html o xhtml que funcionan como espacios web, además las hojas de estilos nacen de la necesidad de diseñar la información de tal manera que podemos separar el contenido de la presentación y, así, por una misma fuente de información, generalmente definida mediante un lenguaje de marcaje, ofrecer diferentes presentaciones en función de dispositivos, servicios, contextos o aplicativos. Por lo que un mismo documento HTML, mediante diferentes hojas de estilo, puede ser presentado por pantalla, por impresora, por lectores de voz o por tabletas braille. (Collell, s.a.).

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas, y al crearlas se utiliza en primer lugar el lenguaje HTML/XHTML para marcarlos contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc. Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc. (Eguiluz, 2015).

En concordancia con lo mencionado por los anteriores autores, CSS es un lenguaje que trabaja conjuntamente con HTML para proveer hojas de estilo visuales que define apariencias gráficas y por ende mejora la apariencia del

software que se realiza llamando de manera evidente la atención de quien lo utiliza ya que se ven animados, dinámicos como por ejemplo los juegos.

2.9.3 Backend

- **PHP:** Page Hypertext Pre-processor, más conocido por el acrónimo PHP, es un lenguaje de programación muy extendido especialmente en el entorno del desarrollo web. La primera versión de PHP fue creada por Rasmus Lerdorf en 1995, y fue uno de los primeros lenguajes de programación del lado del servidor, es decir: al ser ejecutado en el servidor web directamente tiene acceso a bases de datos, conexiones en red y otras tareas antes de que se envíe la página final a la cual el cliente tendrá acceso. Además, PHP tenía la peculiaridad de poder incorporar directamente un documento HTML sin la necesidad de tener que llamar a un archivo externo que procese los datos. (Arias, 2017).
- **Apache:** El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual según la normativa RFC 2616. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que alguien quería que tuviese la connotación de algo que es firme y enérgico, pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de Estados Unidos, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además, Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. En inglés, a patchy server (un servidor "parcheado") suena igual que Apache Server.

Apache es un servidor web de código libre robusto cuya implementación se

realiza de forma colaborativa, con prestaciones y funcionalidades equivalentes a las de los servidores comerciales. El proyecto está dirigido y controlado por un grupo de voluntarios de todo el mundo que, usando Internet y la web para comunicarse, planifican y desarrollan el servidor y la documentación relacionada, estos voluntarios se conocen como el Apache Group. Además del Apache Group, cientos de personas han contribuido al proyecto con código, ideas y documentación. (Carles, 2004).

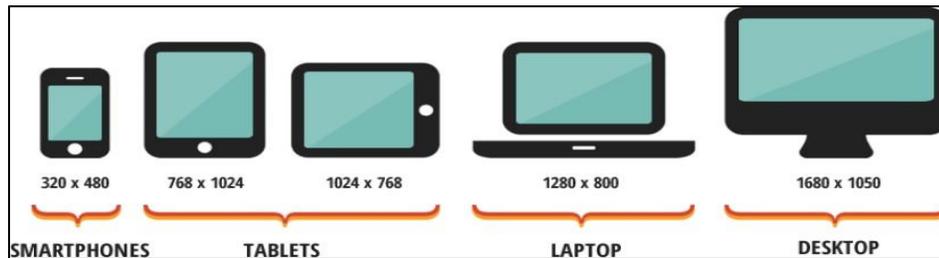
2.9.4 Frameworks

- **BOOTSTRAP:** El framework Bootstrap vio la luz en el año 2011. En un principio fue desarrollado por Twitter aunque posteriormente fue liberado bajo licencia MIT. Hoy en día continúa su desarrollo en un repositorio de GitHub. Se trata de un framework que ofrece la posibilidad de crear un sitio web totalmente responsivo mediante el uso de librerías CSS. En estas librerías, nos podemos encontrar un gran número elementos ya desarrollados y listos para ser utilizados como pueden ser botones, menús, cuadros e incluso un amplio listado de tipografías. (acensTechnologies, 2016)

Desde que vio la luz, Bootstrap se ha caracterizado por tratarse de una excelente herramienta para crear interfaces de usuarios limpias y totalmente adaptables a cualquier tipo de dispositivo y pantalla, independientemente de su tamaño.

Bootstrap establece Media Queries para cuatro tamaños de dispositivos diferentes, variando dependiendo del tamaño de su pantalla. Estas Media Queries permiten desarrollar en dispositivos móviles y tabletas de forma mucho más fácil. (González & Galarza, 2016).

Figura 2. 16
Bootstrap: Tamaño de diferentes dispositivos



Fuente: (González & Galarza, 2016)

- **LARAVEL:** Laravel es un marco de trabajo para aplicaciones web que utiliza una sintaxis de PHP expresiva y elegante. Laravel es accesible, potente y proporciona herramientas necesarias para aplicaciones grandes, robustas y seguras. Se basa en que el desarrollo de un producto de software debe ser una experiencia agradable y creativa para ser verdaderamente satisfactoria. (Encalada, 2018).

Laravel intenta facilitar algunas tareas comunes de los proyectos web, tales como:

- Enrutamiento.
- Inyección de dependencias.
- Múltiples backend para almacenamiento de sesiones y caché.
- Eloquent, motor ORM expresivo e intuitivo.
- Migraciones de esquemas de base de datos agnósticos.
- Robusto procesamiento de trabajos en segundo plano.
- Publicación de eventos en tiempo real.

Laravel simplifica enormemente la interacción con varios motores de base de datos en una variedad de entornos, se permite usar SQL puro, así como también el fluido constructor de consultas integrado con la herramienta ORM (Object Relation Mapper) Eloquent. Actualmente Laravel soporta cuatro motores por defecto, sin embargo, esta es una característica expandible según se incluyan dependencias al proyecto o cambie la conectividad del backend seleccionado: MySQL, PostgreSQL, SQLite y SQL Server.

La configuración de base de datos para cada aplicación se encuentra detallada en el archivo `database.php` del directorio de configuración. En este archivo se encontrarán definidas las propiedades para todas las conexiones que se requiera realizar, así como la conexión que debería ser usada por defecto.

Requisitos de funcionamiento:

- Versión de PHP equivalente o mayor a 7.0 10.
- Extensión OpenSSL de PHP: vincula las funciones de la librería OpenSSL para cifrado y descifrado simétrico y asimétrico, PBKDF2, PKCS7, PKCS12, X509, etc.
- Extensión PDO PHP: define una interfaz ligera y consistente para acceder a bases de datos usando PHP mediante una capa de abstracción de acceso a datos, lo que significa que independientemente de la base de datos se puede hacer uso de las mismas funciones para realizar consultas y recuperar datos sin reescribir SQL.
- Extensión Mbstring PHP: maneja la conversión de codificación de caracteres entre los posibles pares de codificación y proporciona funciones para manipulación de cadenas específicas de varios bytes que ayudan a manejar las codificaciones multibyte en PHP.
- Extensión Tokenizer PHP: funciones que proporcionan una interfaz para el tokenizer PHP para permitir escribir herramientas de análisis o modificación de fuentes PHP sin tener que lidiar con la especificación del lenguaje a nivel léxico.

2.10 SEGURIDAD DEL SISTEMA

Por seguridad de la información se entiende el conjunto de medidas preventivas y reactivas que permiten resguardar y proteger la información. Dicho de otro modo, son todas aquellas políticas de uso y medidas que afectan al tratamiento de los datos que se utilizan en una organización.

La seguridad de la información, como concepto, se basa en cuatro pilares: la disponibilidad, la integridad, la confidencialidad y la autenticación.

- **Disponibilidad:** Acceso a la información cuando se requiere, teniendo en cuenta la privacidad. Evitar “caídas” del sistema que permitan accesos ilegítimos, que impidan el acceso al correo.
- **Confidencialidad:** Información accesible solo para personal autorizado. La información no debe llegar a personas o entidades que no estén autorizados.
- **Integridad:** Información correcta sin modificaciones no autorizadas ni errores. Se protege frente a vulnerabilidades externas o posibles errores humanos.
- **Autenticación:** Información procedente de un usuario que es quien dice ser. Se verifica y se debe garantizar que el origen de los datos es correcto. (Tecon, 2019).

2.10.1 ISO/IEC 27000

La ISO 27000 es una norma que define de qué manera se debe implantar un Sistema de Gestión de la Seguridad de la Información en una empresa u organización.

Su implantación ofrece a la organización o empresa la ventaja de proteger su información de la forma más fiable posible, persiguiéndose para ello un total de tres objetivos principales:

- Preservar la confidencialidad de sus datos.
- Conservar la integridad de sus datos.
- Disponibilidad de la información protegida.

No en vano, tal es su importancia que la implantación de este tipo de herramienta garantiza que los riesgos de seguridad de la información son controlados por la organización eficientemente, tanto de forma interna como al resto de las empresas. (Orozco & Beas, 2018).

2.11 ENCRIPCIÓN AES-256

AES (Advanced Encryption Standard) significa Estándar de Cifrado Avanzado, es uno de los algoritmos de cifrado más utilizados y seguros actualmente disponibles la cual sigue siendo el único algoritmo en la lista del National Institute of Standards and Technology (NIST) para proteger datos clasificados.

AES es lo que se conoce como un cifrado simétrico por bloques, para ello, utiliza una clave criptográfica específica, que es efectivamente un conjunto de protocolos para manipular información. Esta clave puede ser de 128, 192 o 256 bits de tamaño.

AES-256 la versión clave de 256 bits de AES es el estándar de cifrado utilizado por Laravel. Es la forma más avanzada del cifrado y consiste en 14 rondas de sustitución, transposición y mezcla para un nivel de seguridad excepcionalmente alto.

2.11.1 Características de AES-256

AES-256 es el primer cifrado públicamente accesible y abierto aprobado por la Agencia Nacional de Seguridad de Estados Unidos (NSA) para la información ultra-secreta. Su tamaño de clave mayor hace que sea esencialmente irrompible. AES-256 también tiene la ventaja de ser extremadamente rápido. Cuando navegas por la web con una VPN que utiliza el cifrado AES-256 en sus servidores, no experimentarás ninguna disminución en el rendimiento en comparación con otro protocolo de seguridad (LeVPN, 2020).

2.11.2 Formas de uso

El estándar de cifrado avanzado 256 o AES-256 actualmente es utilizado por el framework Laravel por su alto nivel de seguridad de encriptación, una de las utilidades que tiene es proteger la confidencialidad de una contraseña, ya que podría estar en textoplano y ser accesible por cualquiera y aun así no poder ser capaces de deducirla.

3 MARCO APLICATIVO

3.1 INTRODUCCIÓN

En este capítulo se efectúa el diseño y desarrollo del sistema correspondiente.

Para poder aplicar la ingeniería de requisitos, se aplica las actividades correspondientes a la conceptualización, análisis, diseño y desarrollo del sistema de acuerdo a un proceso. Cuando comienza se determina que parte o módulos se van a desarrollar, tomando en cuenta criterios de prioridad del centro médico.

Para el desarrollo del sistema se escoge la metodología ágil UWE, que utiliza un modelo de desarrollo incremental, y este se complementa con la metodología UML para las etapas de desarrollo.

El uso de la metodología UWE implica la realización de actividades por cada una de sus fases, presentando entregables en cada una de ellas. A continuación, se describe por cada fase de la metodología, las actividades que se realizaran a lo largo de todo el desarrollo de este capítulo:

Tabla 3. 1
Fases del Modelado UWE

FASES	ACTIVIDADES
Captura y Análisis de requisitos	En simple palabras y básicamente, durante esta fase, se adquieren, reúnen y especifican las características funcionales y no funcionales que deberá cumplir la aplicación Web.
Diseño del sistema	Se basa en la especificación de requisitos producido por el análisis de los requerimientos (fase de análisis), el diseño define cómo estos requisitos se cumplirán, la estructura que debe darse a la aplicación Web.

Codificación del software	Durante esta etapa se realizan las tareas que comúnmente se conocen como programación; que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior.
Pruebas	Las pruebas se utilizan para asegurar el correcto funcionamiento de secciones de código, mensajes de error, y demás. Los elementos de datos particulares.
Implementación del Sistema	Es el proceso por el cual los programas desarrollados son transferidos apropiadamente al computador destino, inicializados y eventualmente, configurados, todo ello con el propósito de ser ya utilizados por el usuario final.
Mantenimiento	Es el proceso de control, mejora y optimización del software ya desarrollado e instalado.

Fuente: (Elaboración propia)

3.2 CAPTURA Y ANALISIS DE REQUERIMIENTOS

La captura de requerimientos se realizó de acuerdo al análisis realizado en el centro médico “San Simón” específicamente en el área de diagnóstico y tratamiento de pacientes, que vendrán a ser organizados por los módulos siguientes.

3.2.1 Requerimientos Funcionales

La captura de requerimientos se realizó de acuerdo al análisis realizado en el centro médico San Simón específicamente en el área de diagnóstico y tratamiento de pacientes, que será organizada por los siguientes módulos:

Tabla 3. 2
Requerimientos funcionales

R No	REQUERIMIENTOS	CATEGORIA
R1	Base de datos remota para almacenar los registros del sistema.	Evidente
R2	Autenticación y Validación de usuarios según el rol designado por el administrador.	Evidente
R3	Diseño de la interfaz de registros de personal médico, pacientes.	Evidente
R4	El sistema debe realizar búsquedas mediante filtros.	Evidente
R5	Proporcionar Informes de manera rápida y eficiente de los pacientes.	Evidente
R6	Gráficos estadísticos	Evidente

Fuente: (Elaboración propia)

3.2.2 Requerimientos No Funcionales

Los requerimientos no funcionales especifican los criterios que debe cumplir para que sea adecuado el uso, ya para el cual se tiene algunas especificaciones:

Tabla 3. 3 *Requerimientos No Funcionales*

CRITERIOS	DESCRIPCIÓN
Usabilidad	Interfaz que permita al administrador facilitar el manejo del sistema.
Seguridad	Verificación de la autenticidad de todos los usuarios registrados en la base de datos, antes de acceder al sistema Web.
Operatividad	Tener acceso al sistema de manera fácil.
Calidad	Brindar un sistema confiable, seguro en consecuencia de su proceso interno a de asegurar la calidad.

Fuente: (Elaboración propia)

3.3 DISEÑO DEL SISTEMA WEB

3.3.1 Casos de Uso

3.3.1.1 Descripción de Actores

Los actores representan a los usuarios que presenta el sistema. Se comprende como usuario cualquier persona que llegue a interactuar con el sistema. A continuación, se describe a los actores que interactúan con el sistema que se desarrolla.

Tabla 3. 4
Descripción de Actores

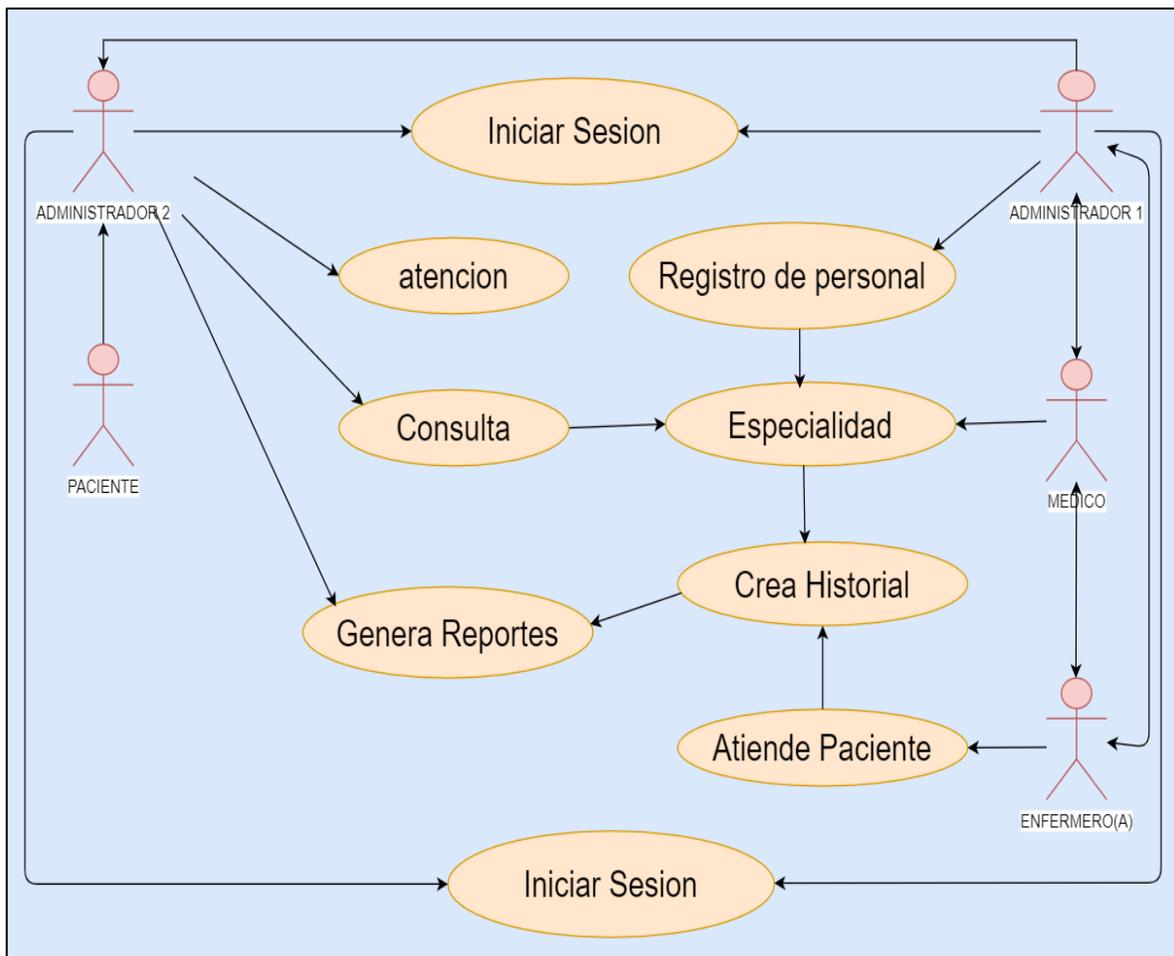
ACTORES	DESCRIPCION
 Administrador 1	En este caso es el Medico Gerente de la Clínica San Simón, es el actor con mayor privilegio. Es el usuario que autoriza y puede realizar cualquier tarea que del sistema.
 Administrador 2	En este caso es el que realiza la atención al cliente de la Clínica San Simón, es el actor con menor privilegio al administrador 1. Es el usuario con la tarea de ordenar a los pacientes según la especialidad que buscan.
 Médico	En este caso es el médico designado al paciente, es el actor con pocos privilegios, es el usuario que realiza o abre un historial para el paciente.
 Enfermero(a)	En este caso es el enfermero designado para la medición de los signos vitales al paciente, es el actor con menos privilegios al sistema, es el usuario que realiza el cuidado del paciente durante su estancia en la clínica.

Fuente: (Elaboración propia)

3.3.1.2 Caso de Uso General

El siguiente diagrama de caso de uso general se describe las actividades que se realizan. También se muestra a los actores pertenecientes al Centro Médico San Simón forman parte de esta descripción.

Figura 3. 1
Caso de uso general



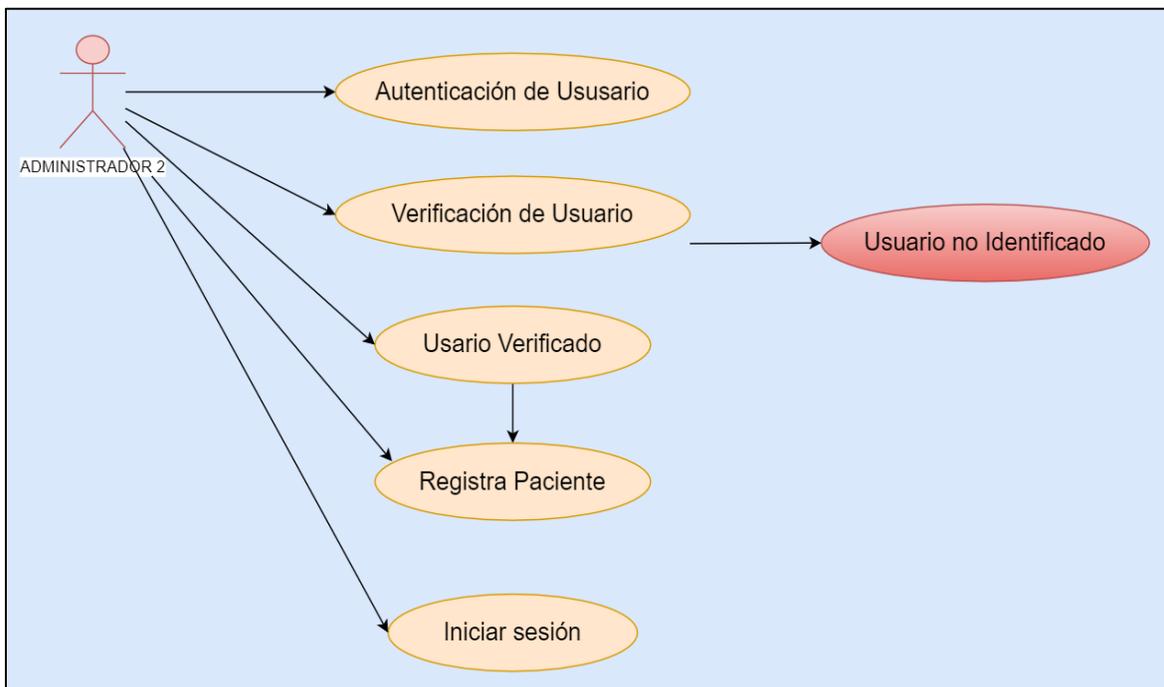
Fuente: (Elaboración propia)

3.3.1.3 Especificaciones de Casos de Uso

a) Caso de Uso: Inicio de sesión

En la figura se muestra el inicio de sesión, describe como un usuario (administrador2) ingresa al sistema web, para previamente acceder a la página de ingreso del sistema, ingresa su nombre y contraseña, si los datos son correctos ingresa al sistema, si no muestra un mensaje de error.

Figura 3. 2
Caso de Uso - Inicio de sesión



Fuente: (Elaboración propia)

En la siguiente tabla se describirá el caso de uso autenticación (nombre de usuario, contraseña) de usuario para el ingreso al sistema.

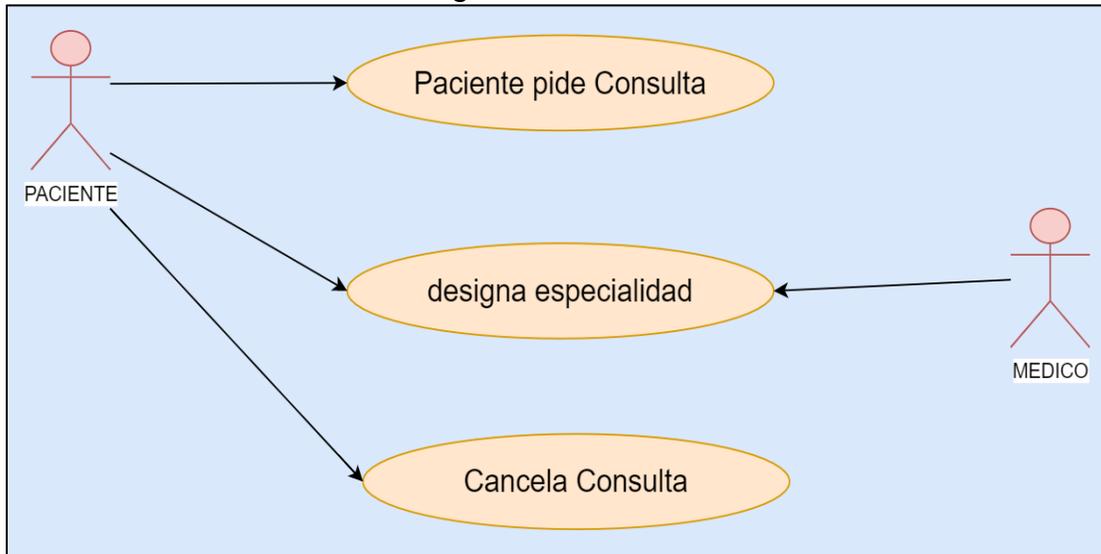
Tabla 3. 5
Caso de Uso – Inicio de sesión

CASO DE USO: INICIO DE SESIÓN	
Actor (es)	Administrador 2 (ventanilla de atención)
Descripción	En este caso de uso, el usuario se autentifica con un nombre de usuario y contraseña, estos datos son asignados por el administrador 1, para que de esta forma el usuario se autentifique y realice sus respectivas actividades en el sistema.
Precondición	Cada actor debe estar previamente registrado en la base de datos del sistema.
Pos condición	Se autentica al usuario, reconociendo sus datos y el rol de usuario que este tiene.
Flujo de Eventos	<p>Básico</p> <ul style="list-style-type: none"> • El usuario Ingresa al sistema • El sistema solicita nombre de usuario y contraseña. • El sistema valida los datos y verifica el rol que tiene el usuario. <p>Alternativo</p> <ul style="list-style-type: none"> • Si el sistema encuentra datos que no concuerdan con los requeridos, el sistema no permite el ingreso al sistema.

Fuente: (Elaboración propia)

b) Caso de Uso: Registro de Paciente

Figura 3. 3
Registro de Paciente



Fuente: (Elaboración propia)

En la siguiente tabla se describirá el caso de uso registrar pacientes en donde se describirán los procesos de esta actividad.

Tabla 3. 6
Caso de Uso – Registro de Pacientes
CASO DE USO: REGISTRO DE PACIENTE

Actor (es)	Paciente, Administrador 2
Descripción	En este caso de uso, el usuario paciente que llega a la clínica pide una consulta médica.
Precondición	Cada actor debe estar previamente registrado en la base de datos del sistema.
Pos condición	Los registros de paciente que se listarán, serán datos que previamente ya fueron registrados en la base de datos.
Flujo de Eventos	<p>Básico</p> <ul style="list-style-type: none"> • El usuario ingresa al sistema. • Se va al menú paciente, selecciona la opción de registrar nuevo paciente.

- Los datos pedidos del paciente que se registraron se almacenaran en la base de datos.
- El paciente registrado quedara en espera.

Alternativo

- .El paciente quedara registrado para un sub siguiente día.

Fuente: (Elaboración propia)

Caso de Uso: Designación de Especialidad

En la siguiente tabla se describe el caso de uso de la designación de especialidad al paciente según su estudio previo realizado.

Tabla 3. 7

Caso de uso – Designación de Especialidad

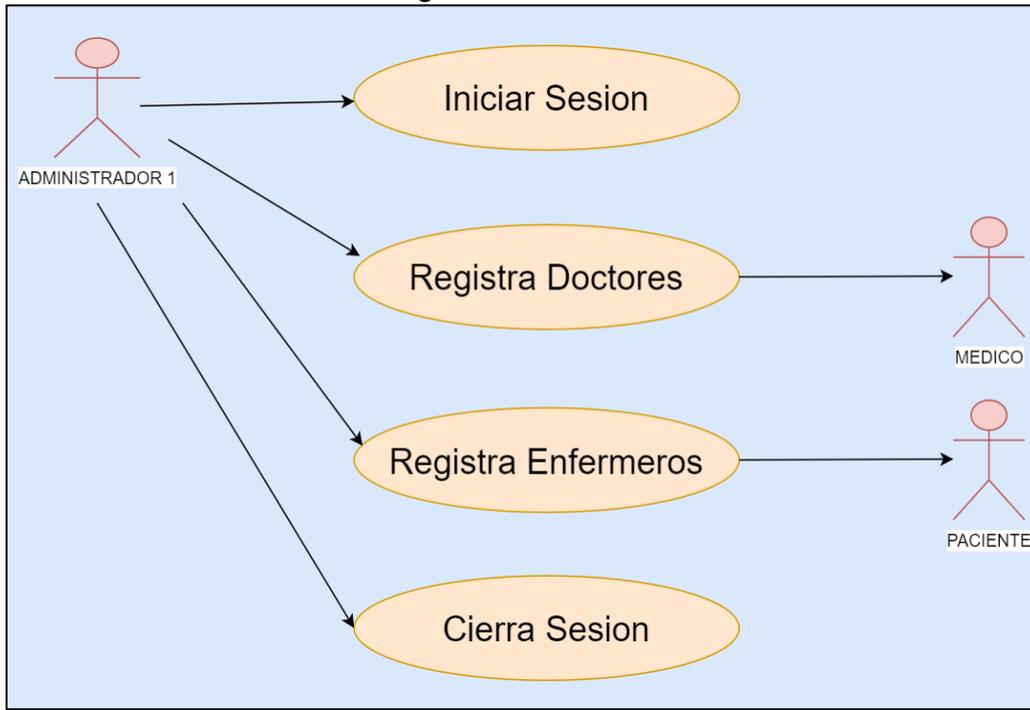
CASO DE USO: DESIGNACIÓN DE ESPECIALIDAD

Actor (es)	Paciente, Médico
Descripción	En este caso de uso, el usuario paciente que llega a la clínica pide una consulta médica, y tenga los estudios necesarios pedidos por el médico, se lo deriva a una especialidad según su estudio clínico.
Precondición	Cada actor debe estar previamente registrado en la base de datos del sistema.
Pos condición	Los registros de los pacientes, serán datos que previamente ya fueron registrados en la base de datos.
Flujo de Eventos	<p>Básico</p> <ul style="list-style-type: none"> • El paciente llega a la clínica. • El paciente realiza nuevamente una cita con el médico. • El muestra los estudios clínicos al médico. • El médico deriva al paciente a una especialidad. <p>Alternativo</p> <ul style="list-style-type: none"> •.El paciente quedara registrado para un sub siguiente día.

Fuente: (Elaboración propia)

c) Caso de Uso: Registro de Usuarios

Figura 3. 4
Registro de Usuarios



Fuente: (Elaboración propia)

En la siguiente tabla se describe la gestión global de las cuentas de usuarios, que se refiere a la creación, modificación e inhabilitación de un usuario del sistema, tomando en cuenta que primero un usuario debe ser registrado como administrador, para poder de esta manera poder tener una cuenta de usuario en el sistema.

Tabla 3. 8

Caso de Uso – Registro de Usuarios

CASO DE USO: REGISTRO DE USUARIOS

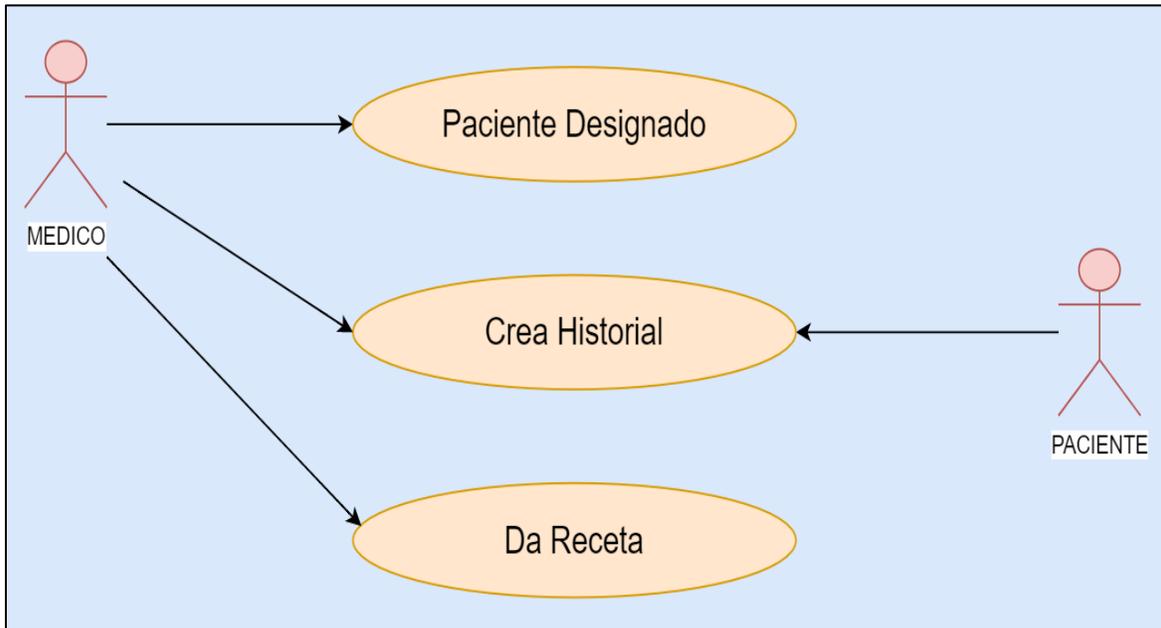
Actor (es)	Administrador 1 (doctor general del centro médico), médico, enfermero.
Descripción	Este caso de uso permite la administración de las cuentas de usuario, es decir, el registro de nuevo usuario como ser los doctores, enfermeros y otros, la modificación, y la inhabilitación de una cuenta de usuario.

Precondición	Cada actor se autentifica en el sistema.
Pos condición	El usuario asigna una contraseña y un nombre de usuario al nuevo registro, los datos son almacenados en la base de datos.
Flujo de Eventos	<p>Básico</p> <ul style="list-style-type: none"> • El usuario ingresa al sistema. • El usuario ingresa al menú autenticación y nuevo usuario. • El usuario administrador llena los datos del nuevo usuario, asignando también el rol que tendrá en el sistema. • El usuario asigna contraseña y su nombre de usuario al nuevo registro de usuario. • El sistema valida los datos. • El sistema guarda los datos en la base de datos. <p>Alternativo</p> <p>Si el sistema encuentra datos que no concuerdan con los requeridos, el sistema despliega una alerta de informe que no se introdujo los datos.</p> <ul style="list-style-type: none"> • El usuario ingresa al menú autenticación, listar usuarios y modificar datos de usuario, se desplegarán los datos del usuario, el usuario modifica los datos incluyendo la contraseña si fuera necesario y tiene la opción de guardar los cambios. • El usuario ingresa al menú autenticación, listar usuarios e inhabilitar al usuario, para restringir acceso total al sistema. • El usuario ingresa al menú autenticación, usuarios inhabilitados y habilitar al usuario, para devolver acceso al sistema.

Fuente: (Elaboración propia)

d) Caso de Uso: Médico

Figura 3.
5 Médico



Fuente: (Elaboración propia)

En la siguiente tabla se describirá el caso de uso Médico del sistema donde se detallarán las características del caso de uso.

Tabla 3. 9
Caso de Uso – Médico

CASO DE USO: MÉDICO	
Actor (es)	Médico, Paciente
Descripción	En este caso de uso, el usuario medico realiza la valoración del paciente, para que de esta forma el usuario veredicto con una acción de acuerdo su valoración.
Precondición	Cada actor debe estar previamente registrado en la base de datos del sistema.
Pos condición	Se autentica al usuario, reconociendo sus datos y el rol de usuario que este tiene.

Flujo de Eventos **Básico**

- El usuario medico hace la valoración.
- El médico abre un historial para el paciente.
- El médico autoriza estudios secundarios.
- El médico declara tratamiento.

Alternativo

- .El médico solo receta tratamiento básico (pastillas y otros).

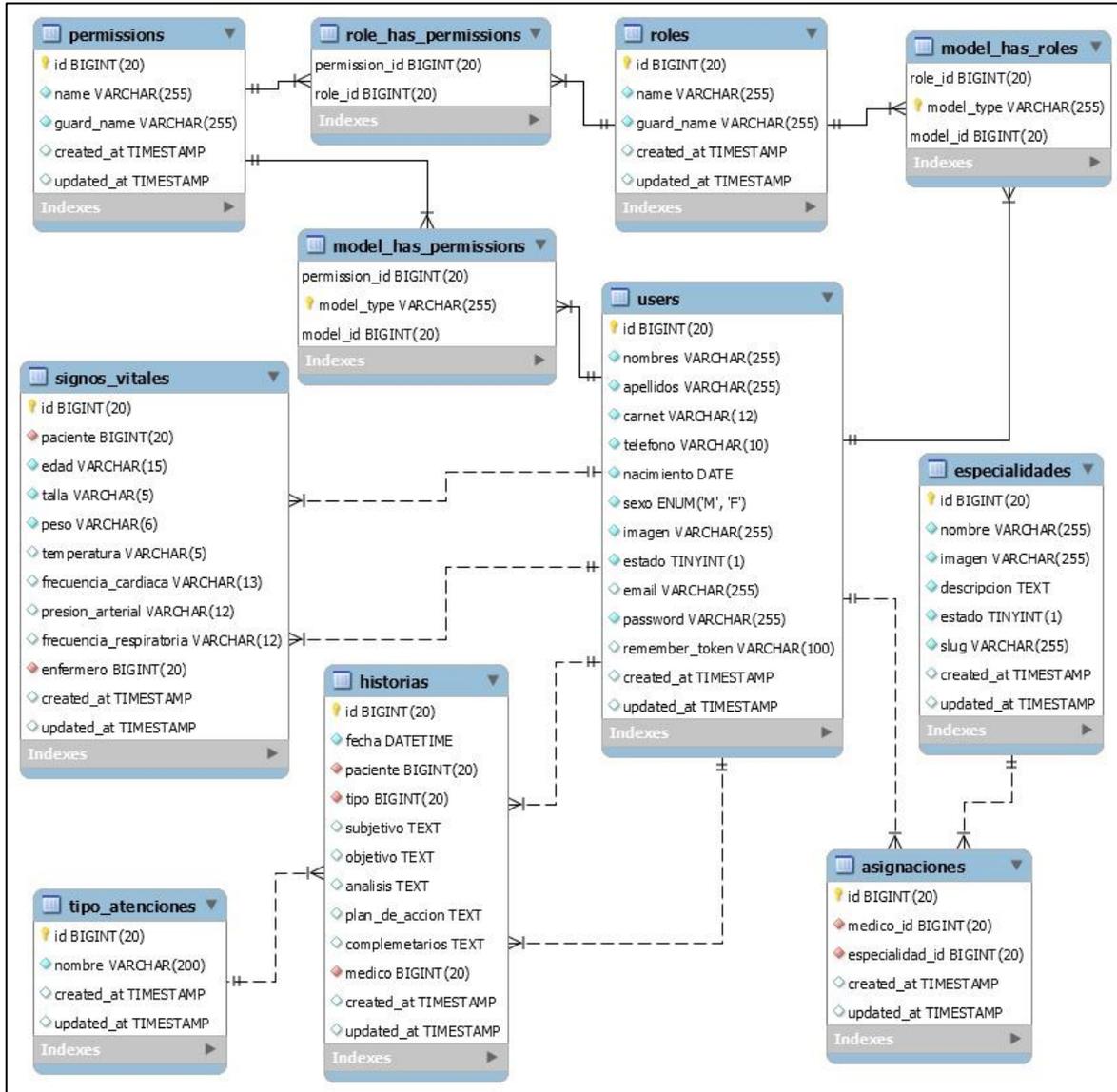
Fuente: (Elaboración propia)

3.3.2 Diseño Conceptual

3.3.2.1 Modelo Conceptual

El diagrama de diseño conceptual, describe cada una de las clases de dominio del sistema web y la relación con cada una de las clases, se presenta el a siguiente figura

Figura 3. 6
Modelo Físico de la Base de Datos



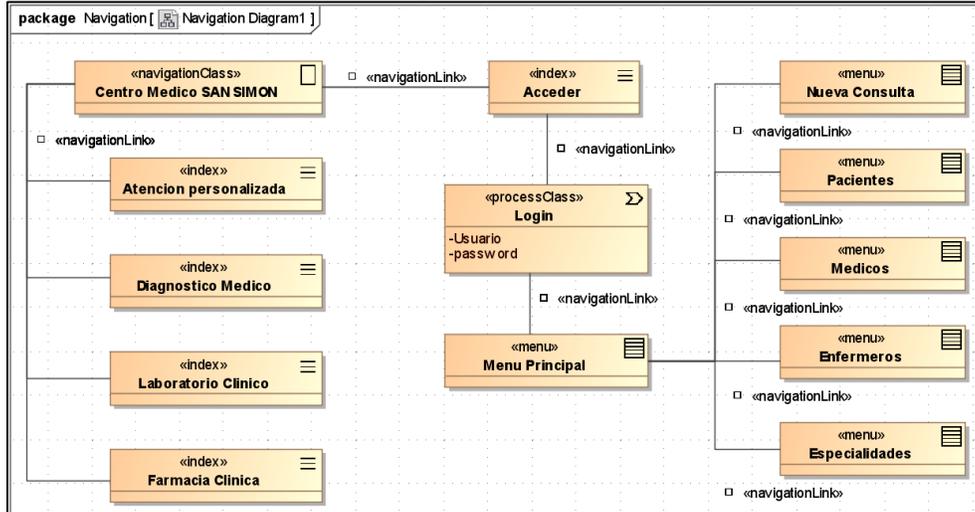
Fuente: (Elaboración propia)

3.3.3 Diseño de Navegación

3.3.3.1 Diagrama de Navegación General

En el diagrama navegación general. Se describe la función de cada actividad del sistema en forma general y como el usuario final podría navegar.

Figura 3.7
Navegación - General

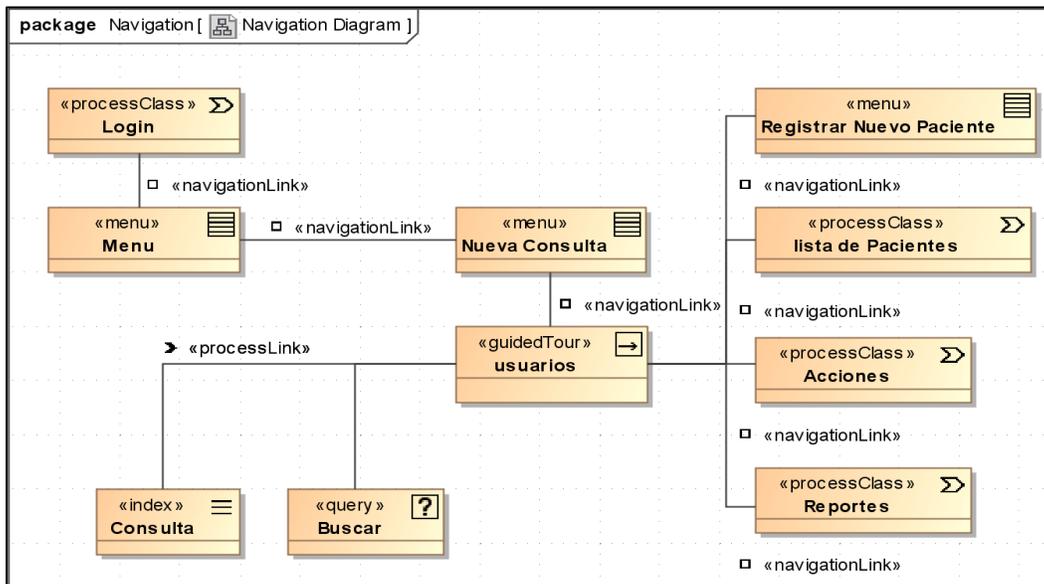


Fuente: (Elaboración propia)

3.3.3.2 Diagrama de Navegación Nueva Consulta

El diseño de navegación de nueva consulta se muestra las opciones de navegación, partiendo del Inicio de Sesión (login).

Figura 3.8
Diagrama de Navegación Nueva Consulta

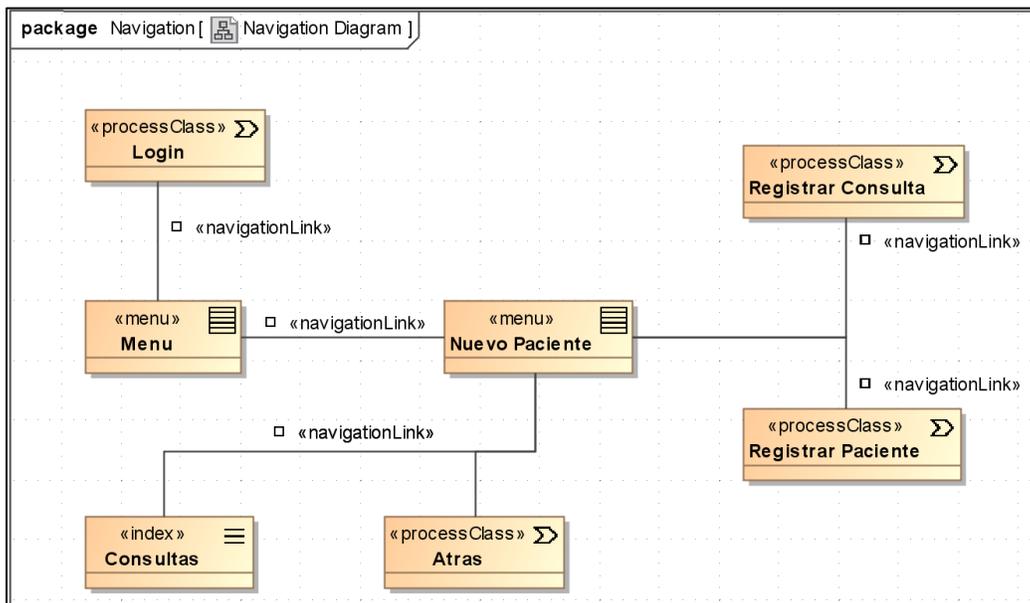


Fuente: (Elaboración propia)

3.3.3.3 Diagrama de Navegación Nuevo Paciente

El diseño de navegación de nuevo paciente se muestra las opciones de navegación, partiendo desde el login.

Figura 3. 9
Diagrama de Navegación Nuevo Paciente

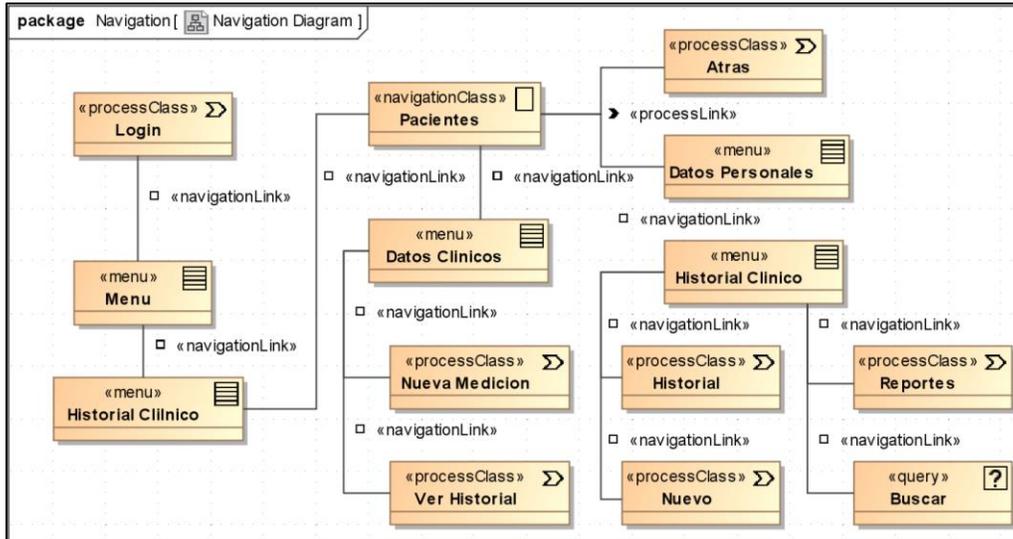


Fuente: (Elaboración propia)

3.3.3.4 Diagrama de Navegación Historial

El diseño de navegación del historial se muestra las opciones de navegación, partiendo desde el login que ve en el cuadro.

Figura 3. 10
Diagrama de Navegación Historial

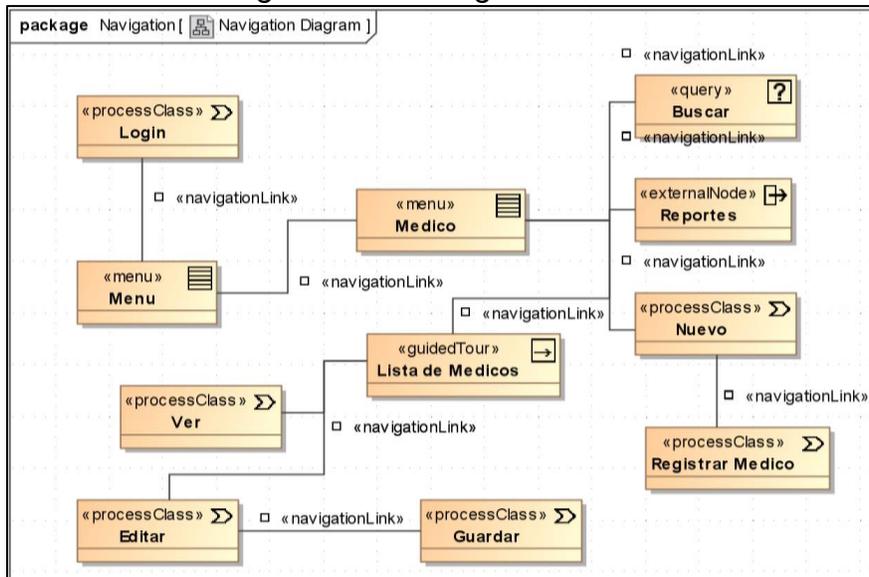


Fuente: (Elaboración propia)

3.3.3.5 Diagrama de Navegación Médico

El diseño de navegación médico se muestra las opciones de navegación, partiendo desde el login que se ve en el cuadro siguiente.

Figura 3. 11
Diagrama de Navegación Médico

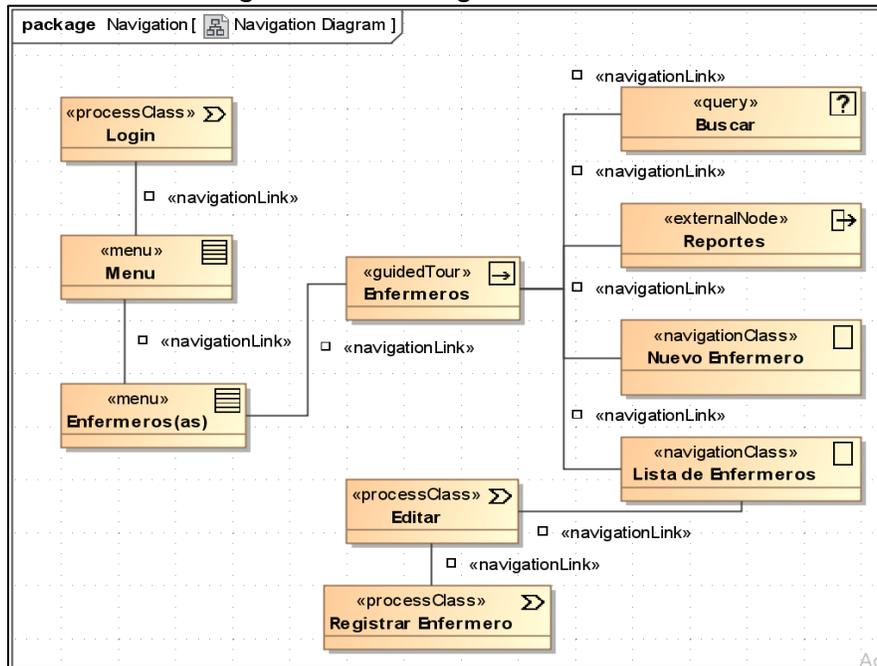


Fuente: (Elaboración propia)

3.3.3.6 Diagrama de Navegación Enfermero

El diseño de navegación enfermero se muestra las opciones de navegación, partiendo desde el login como se puede observar en el siguiente diagrama.

Figura 3.12
Diagrama de Navegación Enfermero

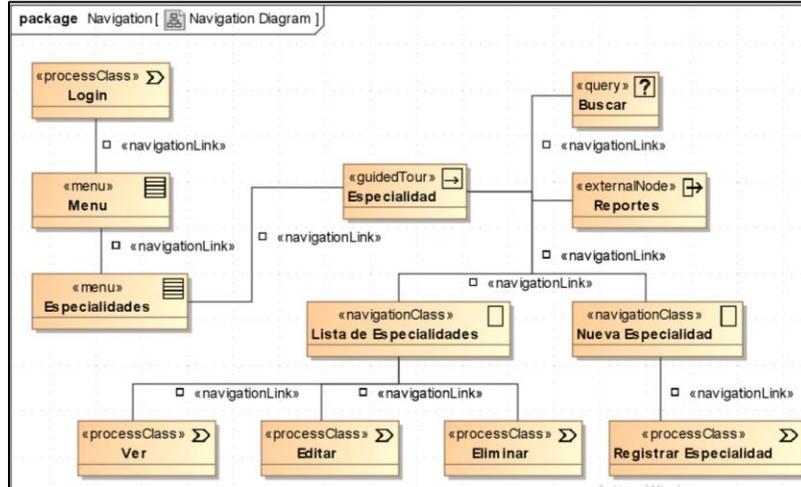


Fuente: (Elaboración propia)

3.3.3.7 Diagrama de Navegación Especialidades

El diseño de navegación especialidades se muestra las opciones de navegación, partiendo desde el login de cada especialidad.

Figura 3. 13
Diagrama de Navegación Especialidades



Fuente: (Elaboración propia)

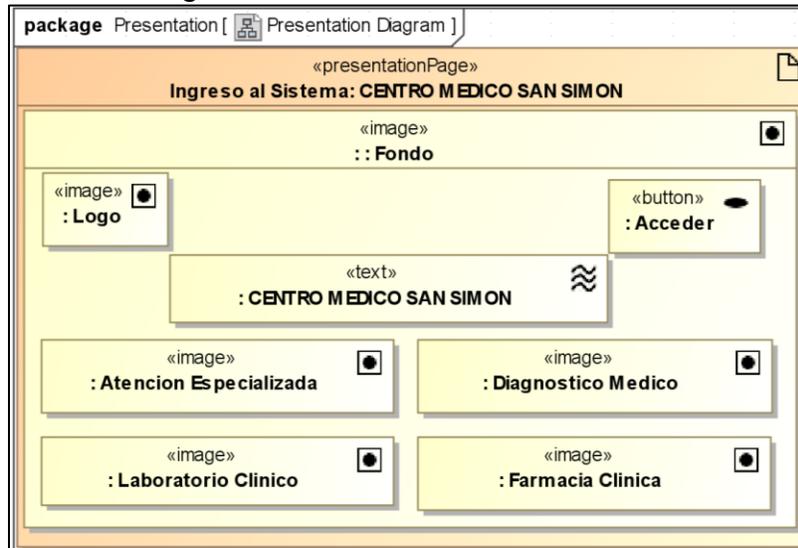
3.3.4 Diseño de Presentación

Los diagramas de presentación, que se describen a continuación muestran como están estructuradas las páginas del sistema Web.

3.3.4.1 Diagrama de Presentación del Sistema

A continuación se observa el diagrama de presentación del sistema

Figura 3. 14
Diagrama de Presentación del Sistema

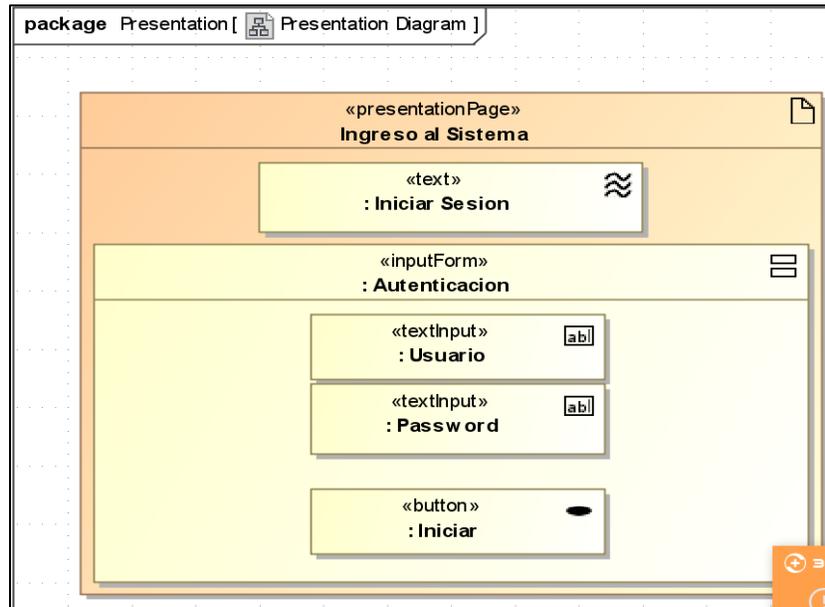


Fuente: (Elaboración propia)

3.3.4.2 Diagrama de Presentación del Login

A continuación se observa el diagrama de inicio de sesión del sistema mediante el login.

Figura 3.15
Diagrama de Presentación del Login

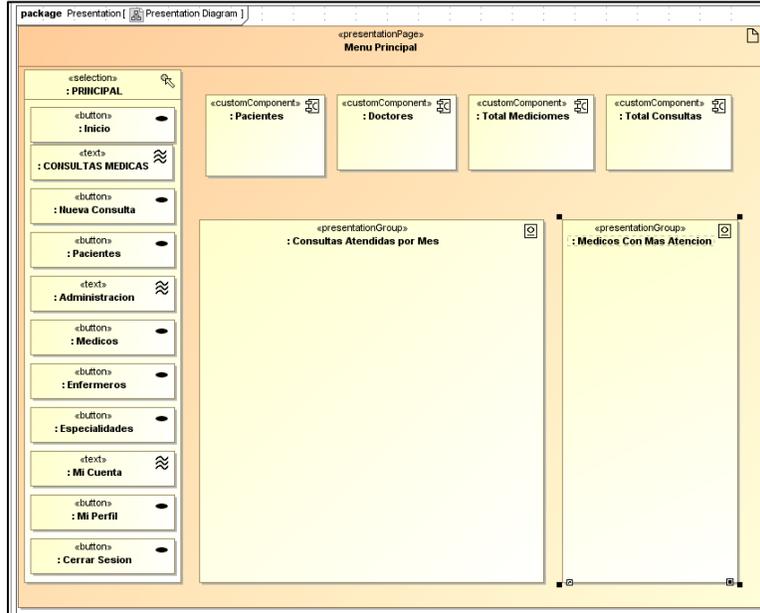


Fuente: (Elaboración propia)

3.3.4.3 Diagrama de Presentación del Menú Principal

El diseño del diagrama de presentación general, muestra las todas opciones con las que cuentan el sistema, ver diagrama siguiente.

Figura 3. 16
Diagrama de Presentación del Menú Principal

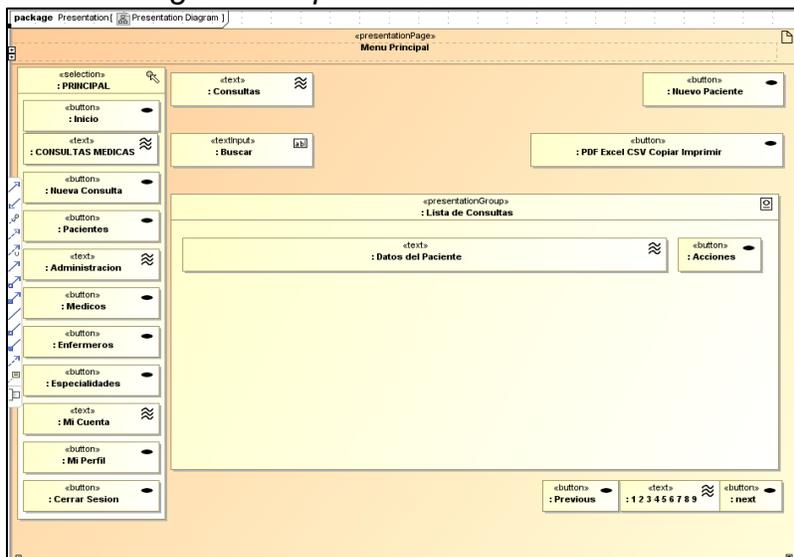


Fuente: (Elaboración propia)

3.3.4.4 Diagrama de Presentación de Consultas

La siguiente imagen describe el diagrama de presentación de la pantalla de consultas de nuevo paciente.

Figura 3. 17
Diagrama de presentación de Consultas

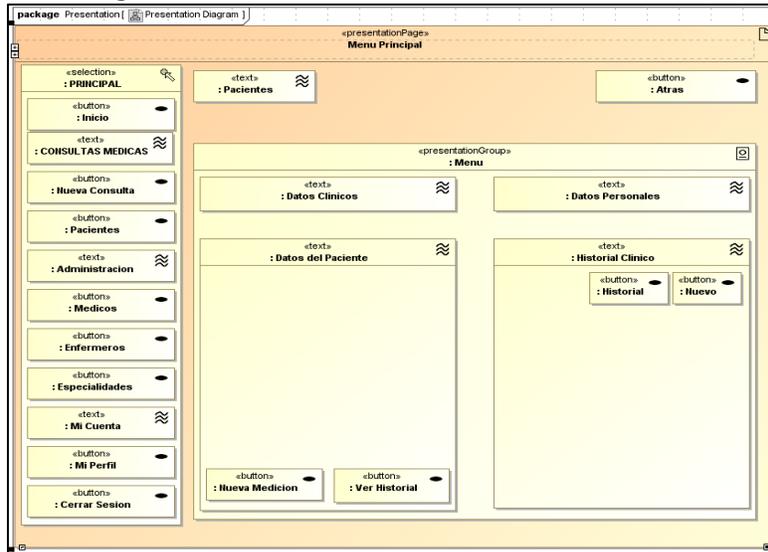


Fuente: (Elaboración propia)

3.3.4.5 Diagrama de Presentación del Historial Clínico

El diagrama de presentación del historial clínico, permite visualizar las interfaces que implica el registro de pacientes, así como sus modificaciones.

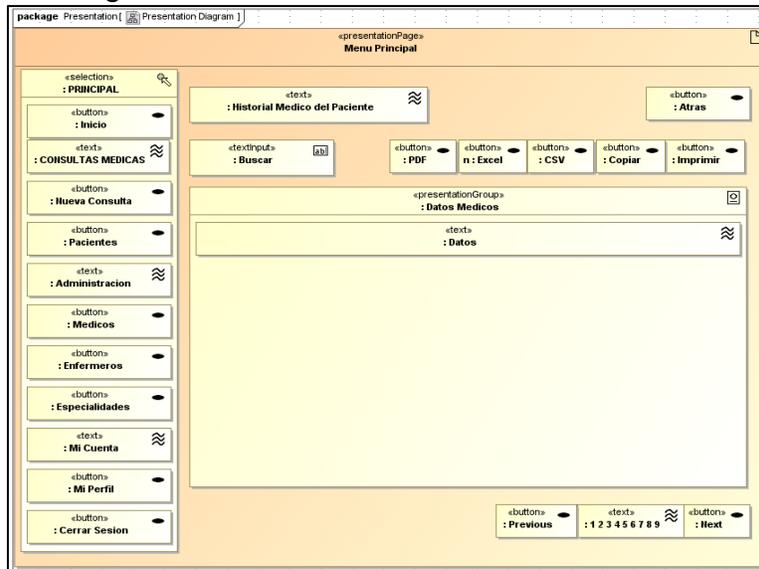
Figura 3.18
Diagrama de Presentación del Historial Clínico



Fuente: (Elaboración propia)

3.3.4.6 Diagrama de Presentación del Historial Médico

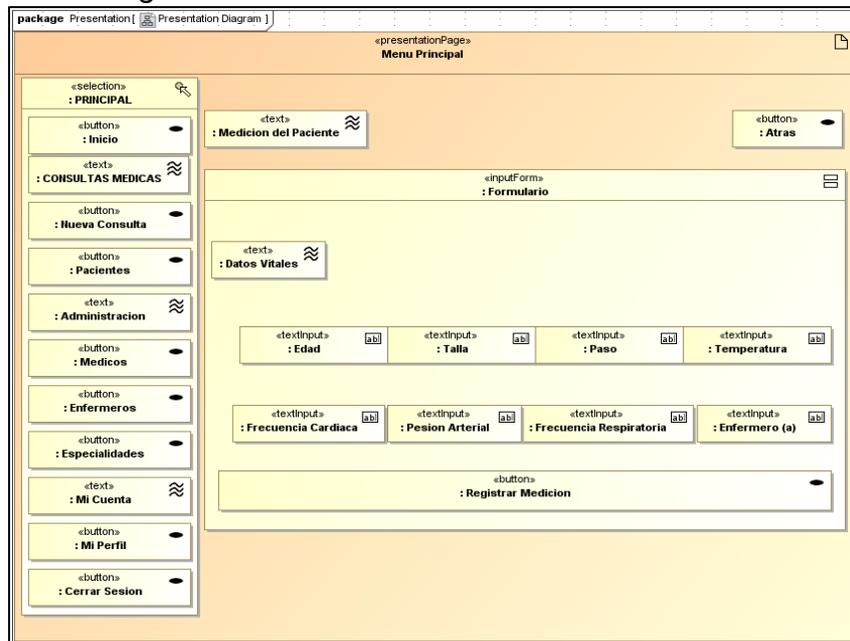
Figura 3.19
Diagrama de Presentación del Historial Médico



Fuente: (Elaboración propia)

3.3.4.7 Diagrama de Presentación de Medición Paciente

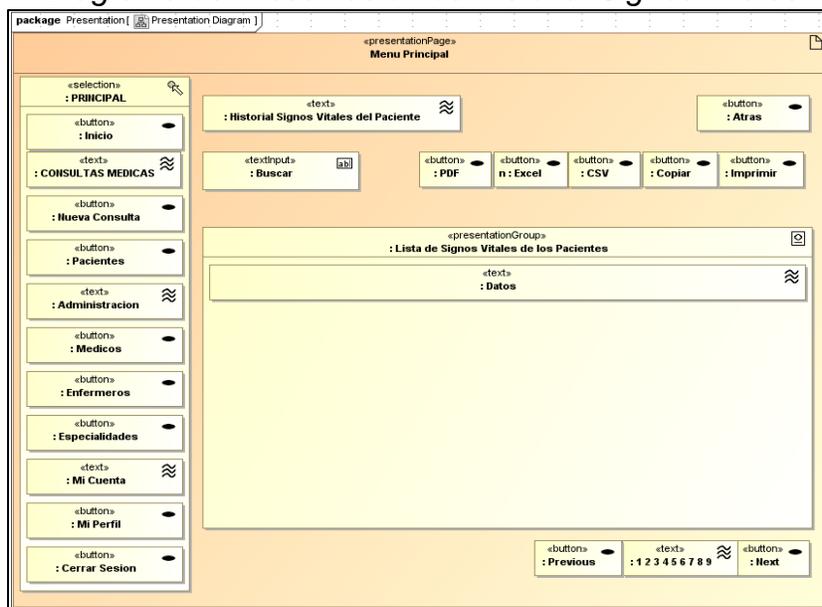
Figura 3. 20
Diagrama de Presentación de Medición Paciente



Fuente: (Elaboración propia)

3.3.4.8 Diagrama de Presentación de Historial Signos Vitales

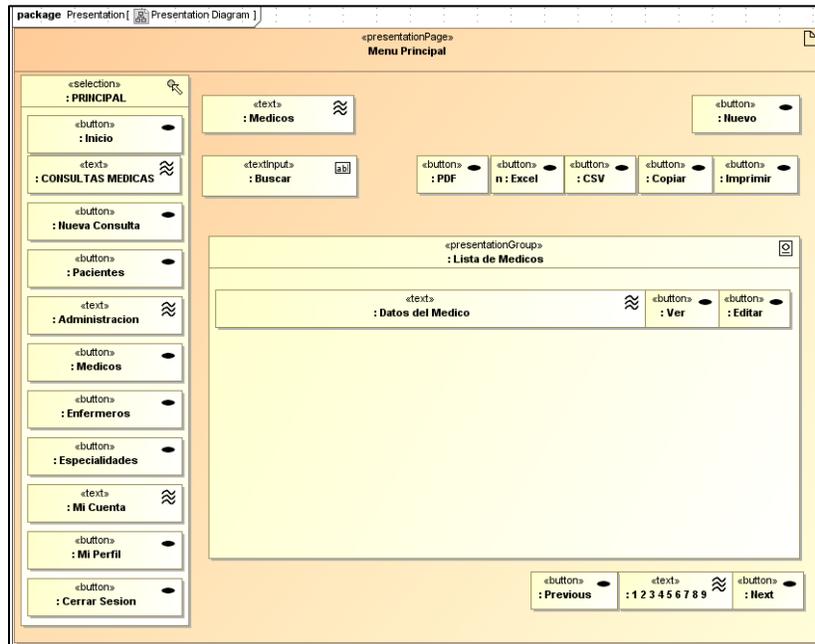
Figura 3. 21
Diagrama de Presentación de Historial Signos Vitales



Fuente: (Elaboración propia)

3.3.4.9 Diagrama de presentación de Médicos

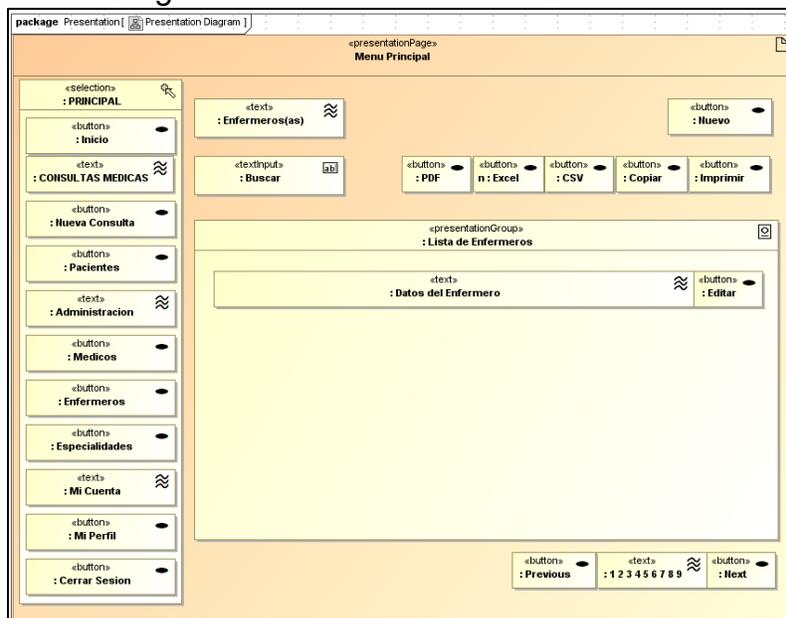
Figura 3.22
Diagrama de Presentación de Médicos



Fuente: (Elaboración propia)

3.3.4.10 Diagrama de Presentación de Enfermeros

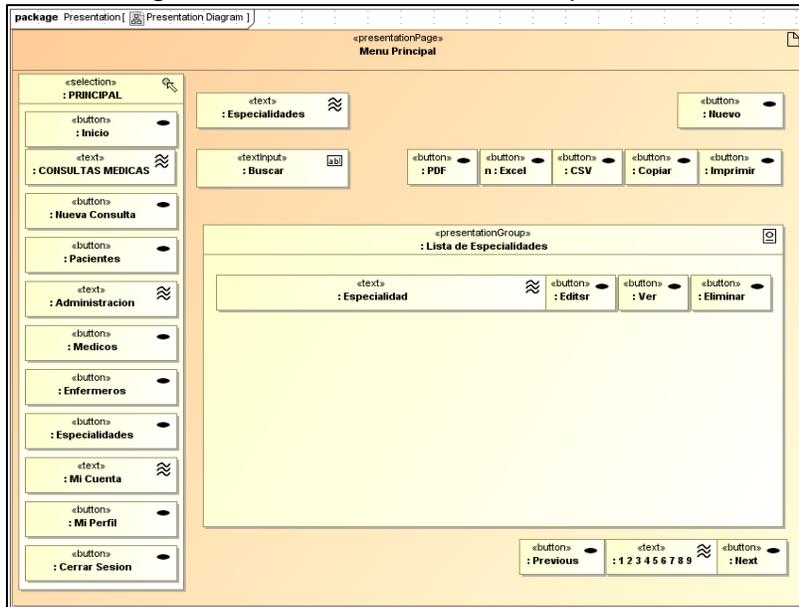
Figura 3.23
Diagrama de Presentación de Enfermeros



Fuente: (Elaboración propia)

3.3.4.11 Diagrama de Presentación del Especialidad

Figura 3. 24
Diagrama de Presentación de Especialidad



Fuente: (Elaboración propia)

3.4 DESARROLLO DEL SISTEMA WEB

Durante esta etapa se realizan las tareas que comúnmente se conocen como programación que consiste, esencialmente, en llevar a código fuente, en el lenguaje de programación elegido, todo lo diseñado en la fase anterior. A continuación, se visualizarán las páginas ya codificadas del sistema:

Figura 3. 25
Interfaz - Principal del Sistema



Fuente: (Elaboración propia)

Figura 3. 26
Interfaz - Principal de Inicio de Sesión

Iniciar Sesión

Carnet De Identidad

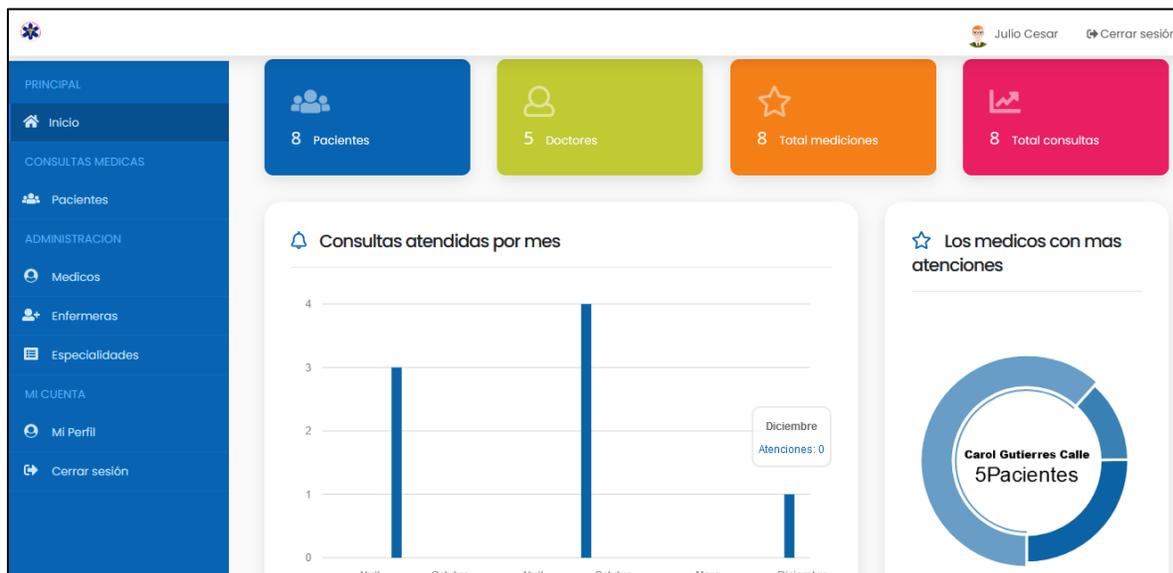
Contraseña

Recuérdame

Acceder

Fuente: (Elaboración propia)

Figura 3. 27
Interfaz – Menú Principal



Fuente: (Elaboración propia)

Figura 3. 28
Interfaz – Pacientes

#	Imagen	Nombres	Apellidos	Carnet	Nacimiento	Edad	Sexo	Telefono	Correo
1		Darwin	Callizaya Flores	9083573	2001-03-11	20	Masculino	76394633	darwind@gmail.com
2		Oliver	Callizaya Ramirez	7853936	2006-02-02	15	Masculino	793457322	oliverc@mail.com
3		Carlos	Colque Ramirez	9687537	1993-02-01	28	Masculino	79354388	carlos33@gmail.com
4		Teodoro	Villegas Mamani	9034916	1987-08-01	34	Masculino	74376173	teol23@gmail.com

Fuente: (Elaboración propia)

Figura 3. 29
Interfaz – Nuevo Paciente

Nombres
Ingrese el nombre completo

Apellidos
Ingrese los apellidos

Carnet de identidad
Ingrese el numero de carnet

Fecha de nacimiento
dd / mm / aaaa

Sexo
Masculino

Telefono/Celular
Ingrese numero de telefono o celular

Correo electronico
Ingrese el correo electronico

Datos vitales

Edad (años) Ingrese el nombre completo

Talla (cm) Ingrese el nombre completo

Peso (kg) Ingrese el nombre completo

Temperatura (°C) Ingrese el nombre completo

Frecuencia cardiaca (x min.) Ingrese el nombre completo

Presion arterial Ingrese el nombre completo

Frecuencia respiratoria (%) Ingrese el nombre completo

Enfermero/a Yuma Mayta

Registrar paciente

Fuente: (Elaboración propia)

Figura 3. 30
Interfaz – Registro de Nueva Consulta

Consultas ← atras

Fecha 2021-05-30

Titulo (SUBJETIVO) Ingrese los subjetivo

Medico Angela Rodriguez Apaza

Tipo de atencion:
 Consulta general
 Plan de tratamiento
 Vacuna
 Examen

Sintomas (OBJETIVO)

Diagnostico (ANALISIS)

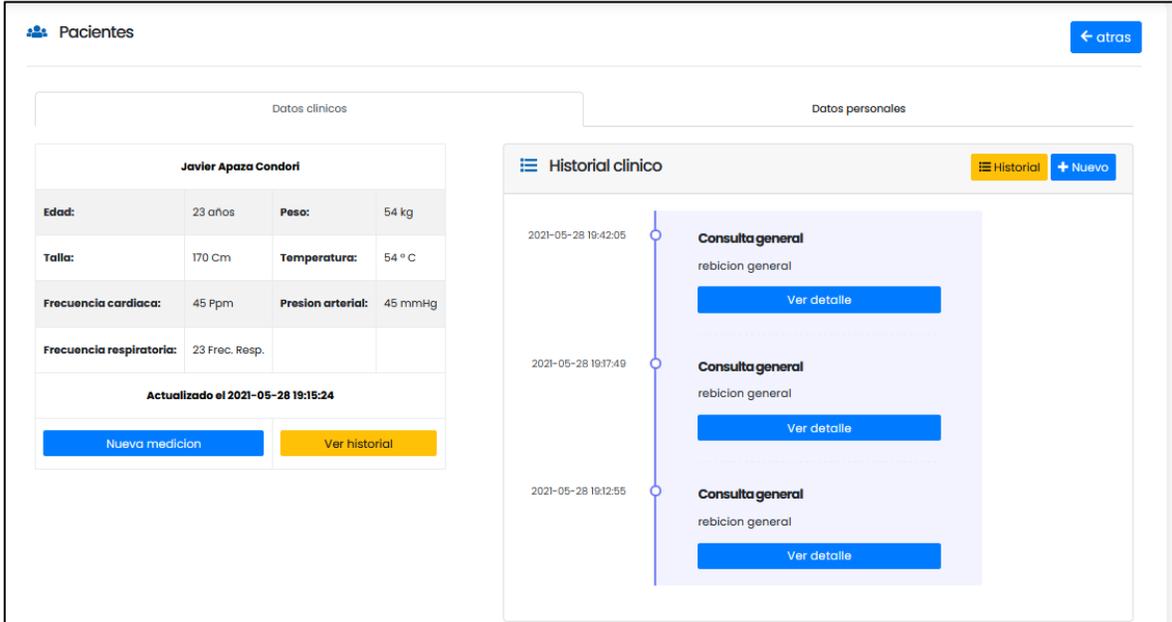
Tratamiento (PLAN DE ACCION)

Examenes (COMPLEMENTARIOS)

Registrar consulta

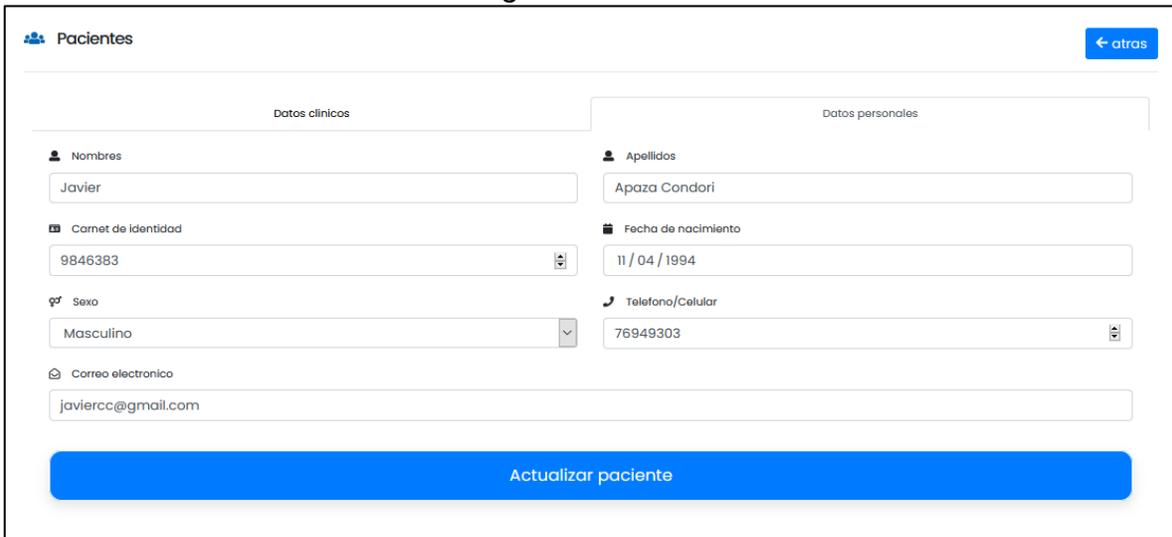
Fuente: (Elaboración propia)

Figura 3. 31
Interfaz – Historial del Paciente



Fuente: (Elaboración propia)

Figura 3. 32
Interfaz – Registro de Nuevo Paciente



Fuente: (Elaboración propia)

Figura 3. 33
Interfaz – Historial Médico de Pacientes

Historial medico de: Javier Apaza Condori ← Atras

Buscar

PDF Excel CSV Copiar Imprimir

#	Fecha	Tipo	Subjetivo	Objetivo	Analisis	Plan de accion	Exámenes Complementarios	Medico
1	2021-05-28 19:42:00	Consulta general	rebicion general	dolores de cabeza	resfrio	tomar 2 aspirinas cada 8 horas	ggg	Angela Rodriguez Apaza
2	2021-05-28 19:17:00	Consulta general	rebicion general	malestar general	resfrio	una tableta po dia	fcds	Angela Rodriguez Apaza
3	2021-05-28 19:12:00	Consulta general	rebicion general	dolores musculares	fiebre	despues de las co idas	tomografia	Gary Saul Huanca Apaza

Mostrando 1 al 3 de 3 registros Previous 1 Next

Fuente: (Elaboración propia)

Figura 3. 34
Interfaz – Medición del Paciente

Medicion paciente: Javier Apaza Condori ← atras

Datos vitales

Edad
 Talla
 Peso
 Temperatura

Frecuencia cardiaca
 Presion arterial
 Frecuencia respiratoria
 Enfermero/a

Registrar medicion

Fuente: (Elaboración propia)

Figura 3. 35
Interfaz – Historial Signos Vitales del Paciente

Historial signos vitales de: Javier Apaza Condori ← Atras

Buscar

PDF Excel CSV Copiar Imprimir

#	Fecha	Edad	Talla	Peso	Temperatura	Frecuencia cardiaca	Presion arterial	Frecuencia respiratoria	Enfermero
1	2021-05-28 19:15:24	23 años	170 cm	54 kg	54 °C	45 Ppm	45 mmHg	23 Frec. Resp.	Yuma Mayta
2	2021-05-28 19:10:31	22 años	45 cm	76 kg	54 °C	54 Ppm	76 mmHg	43 Frec. Resp.	Yuma Mayta

Mostrando 1 al 2 de 2 registros Previous 1 Next

Fuente: (Elaboración propia)

Figura 3. 36
Interfaz – Médicos

#	Imagen	Nombres	Apellidos	Carnet	Nacimiento	Edad	Sexo	Telefono	Correo	Especialidad	Acciones
1		Jose Carlos	Quispe	872654334	2000-02-29	21	Masculino	76543278	Jose@hotmail.com	Traumatología	
2		Gary Saul	Huanca Apaza	9054385	2009-05-21	12	Masculino	76539244	garyha@gmail.com	Medicina General	
3		Angela	Rodríguez Apaza	9058303	1995-03-22	26	Femenino	73545722	angelaro@gmail.com	Ginecología	

Mostrando 1 al 3 de 3 registros

Fuente: (Elaboración propia)

Figura 3. 37
Interfaz – Registro de Nuevo Medico

Medicos ← otros

Información! Acerca de las credenciales de acceso para el paciente.

Para acceder al sistema el paciente podrá acceder con las siguientes credenciales:

- **Usuario:** Numero de carnet
- **Contraseña:** Numero de telefono

Nombres
Ingrese el nombre completo

Apellidos
Ingrese los apellidos

Carnet de identidad
Ingrese el numero de carnet

Fecha de nacimiento
dd / mm / aaaa

Sexo
Masculino

Telefono/Celular
Ingrese numero de telefono o celular

Correo electronico
Ingrese el correo electronico

Especialidades

Registrar medico

Fuente: (Elaboración propia)

Figura 3. 38
Interfaz – Enfermeros

#	Imagen	Nombres	Carnet	Nacimiento	Edad	Sexo	Telefono	Acciones
1		Rolando Calle Quispe	9473045	1987-07-03	34	Masculino	763984977	
2		Neymar Callizaya Ramos	9064570	1995-03-03	26	Masculino	76938599	
3		Carmen Aquino Gutierrez	9037491	1985-03-01	36	Femenino	76394944	
4		Freddy Quispe Mamani	9847382	1987-02-01	34	Masculino	76593623	

Mostrando 1 al 4 de 4 registros

Fuente: (Elaboración propia)

Figura 3. 39
Interfaz – Registro de Nuevo Enfermero

Secretarios/as ← atrás

Información: Acerca de las credenciales de acceso para el secretario/a.

Para acceder al sistema el secretario/a podrá acceder con las siguientes credenciales:

- **Usuario:** Numero de carnet
- **Contraseña:** Numero de telefono

Nombre
 Ingrese el nombre completo

Apellidos
 Ingrese los apellidos

Carnet de identidad
 Ingrese el numero de carnet

Fecha de nacimiento
 dd / mm / aaaa

Sexo
 Masculino

Telefono/Celular
 Ingrese numero de telefono o celular

Correo electronico
 Ingrese el correo electronico

Registrar secretario/a

Fuente: (Elaboración propia)

Figura 3. 40
Interfaz – Especialidades

Fuente: (Elaboración propia)

Especialidades + Nuevo

Buscar

PDF Excel CSV Copiar Imprimir

#	Imagen	Nombre	Condicion	Acciones
1		Traumatología	Activo	
2		Medicina General	Activo	
3		Ginecología	Activo	
4		Rayos X	Activo	
5		Dentista	Activo	

Mostrando 1 al 5 de 5 registros Previous 1 Next

Figura 3. 41
Interfaz – Registro de Nueva Especialidad

Especialidades ← atrás

Nombre
 Ingrese el nombre de la especialidad

Imagen
 Examinar... No se ha seleccionado ningún archivo.

Descripcion

Ingrese una descripción aquí...

Registrar especialidad

Fuente: (Elaboración propia)

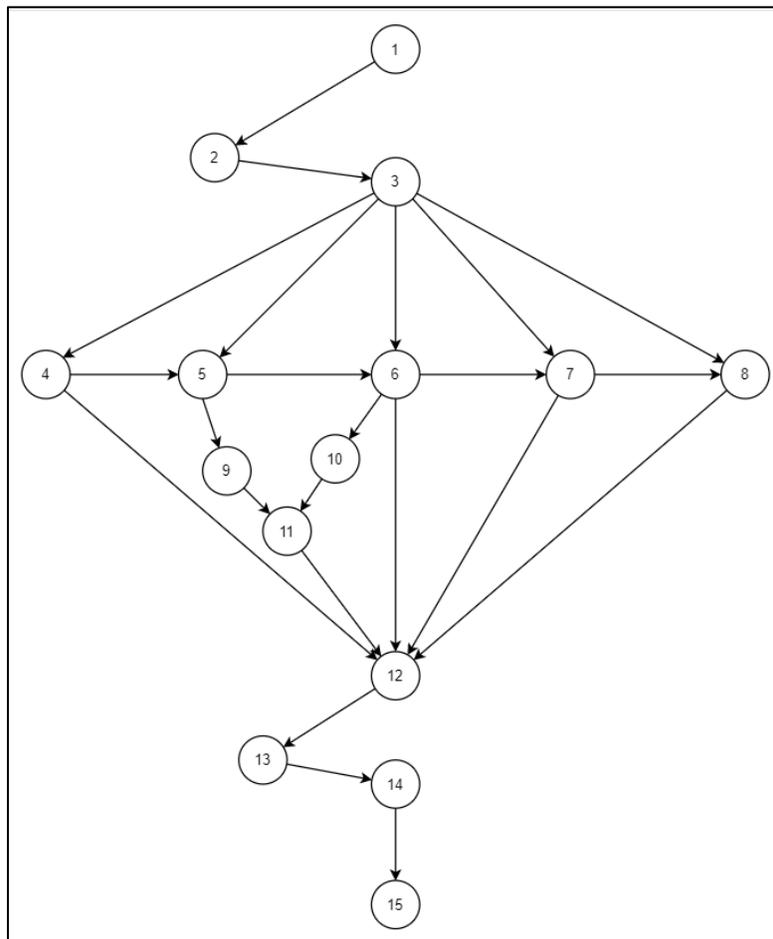
3.5 PRUEBAS DEL SISTEMA WEB

Las pruebas son una serie de pasos planificados para contar con una construcción de software de forma correcta, se realizarán dos tipos de pruebas: pruebas de caja blanca y pruebas de caja negra.

3.5.1 Prueba de Caja Blanca

Esta prueba se orienta al cálculo de las regiones que deben ser consideradas como partes independientes del sistema y estableciendo cuales son las entradas que se ejecutan en cada una de las regiones asegurando así que cada región se ejecuta al menos una vez. De forma general se debe seguir:

- Emplear el diseño del sistema para elaborar el grafo del programa.



Fuente: (Elaboración propia)

Donde se tiene como resultado:

- Inicio del programa (1)
- Menú principal (2)
- Módulo de registros (3)
- Nueva consulta (4)
- Paciente (5)
- Médicos (6)
- Enfermeros (7)
- Especialidades (8)
- Consultas (9)
- Mediciones (10)
- Historial (11)
- Reportes (12)
- Fin del ciclo de registros (13)
- Fin del ciclo del sistema (14)
- Fin del sistema (15)

Una vez analizando el grafo generado a partir de las características del sistema, ahora se procede a determinar la complejidad ciclomática del grafo mediante la fórmula:

$$V(G) = A - N + 2$$

Donde los valores de aristas y nodos son:

$$A = 19$$

$$N = 15$$

Reemplazando en la fórmula: $V(G) = A - N + 2$

$$V(G) = 19 - 15 + 2$$

$$V(G) = 6$$

Determinando los caminos $V(G) = 6$:

Camino 1: 1-2-3-4-12-13-14-15

Camino 2: 1-2-3-5-9-11-12-13-14-15

Camino 3: 1-2-3-6-10-11-12-13-14-15

Camino 4: 1-2-3-6-12-13-14-15

Camino 5: 1-2-3-7-12-13-14-15

Camino 6: 1-2-3-8-12-13-14-15

Luego de tener los caminos, se prepara los casos de prueba para forzar la ejecución de cada camino en esta última condición se establece, que para la ejecución de ciertos caminos se deben establecer las condiciones en las que se ejecuta los nodos establecidos en cada camino:

Tabla 3. 10
Prueba de Caja Blanca - Camino 1

CAMINO 1	
Descripción	El usuario debe iniciar sesión con su nombre de usuario asignado y su contraseña. Deberá seguir los siguientes pasos estando en el menú principal siguiendo los nodos del grafo se obtiene el registro de una consulta de un paciente nuevo, donde se podrá visualizar los detalles del registro.
Condición	El usuario debe tener asignado el rol de administrador, además este debe estar autenticado en el sistema para realizar el registro de un paciente.
Entrada	Los campos que deben llenar para un s consulta son: Nombre, paterno, materno, c.i. y especialidad
Resultado esperado	Guardados en la base de datos una vez hayan sido validados por el sistema.
Evaluación de la prueba	El sistema validara y guardara exitosamente los datos del nuevo paciente.

Fuente: (Elaboración propia)

Tabla 3. 11
Prueba de Caja Blanca - Camino 2

CAMINO 2	
Descripción	El usuario debe iniciar sesión con su nombre de usuario asignado y su contraseña. Deberá seguir los siguientes pasos estando en el menú principal siguiendo los nodos del grafo se obtiene el registro de un paciente nuevo, donde se podrá visualizar los detalles del registro y los pasos que siguen.
Condición	El cliente o paciente debe estar registrado en el sistema para realizarse las valoraciones correspondientes para su atención con el médico.
Entrada	Los campos que deben llenar para un paciente son: Nombre, paterno, materno, c.i. y especialidad
Resultado esperado	Guardados en la base de datos una vez hayan sido validados por el sistema.
Evaluación de la prueba	El sistema validara y guardara exitosamente los datos del nuevo paciente.

Fuente: (Elaboración propia)

Tabla 3. 12
Prueba de Caja Blanca - Camino 3

CAMINO 3	
Descripción	El usuario debe iniciar sesión con su nombre de usuario asignado y su contraseña. Deberá seguir los siguientes pasos estando en el menú principal siguiendo los nodos del grafo se obtiene la atención del paciente con doctor asignado y donde se creara un historial clínico, donde se podrá visualizar los detalles del registro y los

	pasos que siguen.
Condición	El paciente debe estar registrado en el sistema para realizarse las valoraciones correspondientes para su atención con el médico.
Entrada	Los campos que deben llenar para un paciente son: Nombre, paterno, materno, datos clínicos, como ser mediciones de edad, talla peso, y otros
Resultado esperado	Guardados en la base de datos una vez hayan sido validados por el sistema.
Evaluación de la prueba	El sistema validara y guardara exitosamente los datos del nuevo paciente.

Fuente: (Elaboración propia)

Tabla 3. 13
Prueba de Caja Blanca - Camino 4
CAMINO 4

Descripción	El usuario debe iniciar sesión con su nombre de usuario asignado y su contraseña. Deberá seguir los siguientes pasos estando en el menú principal siguiendo los nodos del grafo se obtiene la las actividades asignadas del doctor como crear un historial clínico del paciente, donde se podrá visualizar los detalles del registro y los pasos que siguen.
Condición	El paciente debe estar registrado en el sistema para realizarse las valoraciones correspondientes para su atención con el médico.
Entrada	Los campos que deben llenar para un paciente son: Nombre, paterno, materno, datos clínicos, como

Resultado esperado	ser mediciones de edad, talla peso, y otros Guardados en la base de datos una vez hayan sido validados por el sistema.
Evaluación de la prueba	El sistema validara y guardara exitosamente los datos del nuevo paciente.

Fuente: (Elaboración propia)

Tabla 3. 14
Prueba de Caja Blanca - Camino 5

CAMINO 5

Descripción	El usuario debe iniciar sesión con su nombre de usuario asignado y su contraseña. Deberá seguir los siguientes pasos estando en el menú principal siguiendo los nodos del grafo se obtiene las actividades del enfermero como es de atención a los pacientes, donde se podrá visualizar los detalles del registro y los pasos que siguen.
Condición	El cliente o paciente debe estar registrado en el sistema para realizarse las valoraciones correspondientes para su atención con el médico.
Entrada	Los campos que deben llenar para un paciente son: Nombre, paterno, materno, datos clínicos, como ser mediciones de edad, talla peso, y otros
Resultado esperado	Guardados en la base de datos una vez hayan sido validados por el sistema.
Evaluación de la prueba	El sistema validara y guardara exitosamente los datos del nuevo paciente.

Fuente: (Elaboración propia)

Tabla 3. 15
Prueba de Caja Blanca - Camino 6

CAMINO 6	
Descripción	En esta parte son de las especialidades con las que cuenta la clínica.
Condición	La clínica debe contar con especialidades.
Entrada	Las especialidades.
Resultado esperado	Guardados en la base de datos una vez hayan sido validados por el sistema.
Evaluación de la prueba	El sistema validara y guardara exitosamente los datos del nuevo paciente.

Fuente: (Elaboración propia)

3.5.2 Prueba de Caja Negra

La prueba de caja negra consiste en probar cada una de las funciones del sistema que fueron especificadas en el presente capítulo.

Con este tipo de prueba se debe buscar que las funciones sean operativas, además se debe agotar al sistema de tal manera buscar la mayor cantidad de errores. Son pruebas sobre la interfaz del software.

A continuación, se muestran algunas pruebas relevantes.

- **Prueba de autenticación**

Si el usuario que desea ingresar al sistema es el correcto desplegara la interfaz gráfica de ingreso, Bienvenida al usuario, caso contrario se mostrara un mensaje de acceso denegado.

Tabla 3. 16
Prueba Autenticación

PRUEBA AUTENTICACIÓN	
Descripción	La autenticación de inicio de sesión primeramente se va a la opción Iniciar Sesión para poder acceder al formulario.

Condiciones de ejecución	El usuario deberá iniciar y ejecutar el sistema.
Entradas	<ul style="list-style-type: none"> • Elegir la opción Iniciar Sesión • Llenar el formulario de Usuario y Contraseña.
Resultado esperado	Luego del proceso de autenticación el sistema deberá validar los datos haciendo una comparación en la base de datos permitiendo ingresar al sistema o bien si los datos son erróneos no dejara ingresar y mostrara un mensaje de validación.

Fuente: (Elaboración propia)

3.6 FASE DE MANTENIMIENTO

Con el paso del tiempo, el software desarrollado debe someterse a procesos de modificación que prolonguen su vida útil o mejoren sus características. Corrección de bugs, adaptación a nuevos entornos tecnológicos o incorporación de funcionalidades son algunas de las tareas incluidas en el mantenimiento del software, actividad que se repite periódicamente desde el momento en que se empieza a utilizar hasta el abandono definitivo del sistema.

4 ANÁLISIS DE CALIDAD, COSTOS Y SEGURIDAD

4.1 CALIDAD DE SOFTWARE

4.1.1 Introducción

La calidad del sistema es la concordancia con los requerimientos funcionales y de rendimiento directamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado. Un producto de alta calidad requiere menos mantenimiento y facilita tanto el desarrollo como el mantenimiento de la productividad.

A la conclusión del desarrollo del Sistema Web para el Seguimiento de Diagnóstico y Tratamiento de Pacientes se tiene la necesidad de conocer la calidad obtenida del sistema siendo este un factor importante, por tanto, existen dos tipos de medición directa e indirecta. En el presente proyecto se aplicará las medidas indirectas planteada por la norma ISO 9126. Si bien no se llega a obtener la calidad perfecta se desea lograr una calidad necesaria o suficiente para el momento cuando el usuario así lo requiera.

El desarrollar un software de calidad es el objetivo de todo desarrollador, por tanto, se le dedica muchos esfuerzos, pero se debe tomar en cuenta que todo software debe cumplir y/o superar las expectativas del cliente. El presente proyecto usara la técnica ISO 9126.

4.1.2 Técnica ISO 9126

El objetivo principal de esta técnica es alcanzar la calidad necesaria para satisfacer las necesidades del cliente. La calidad según esta norma ISO 9126 puede ser pedida de acuerdo a los factores:

- Funcionalidad
- Mantenibilidad
- Portabilidad
- Usabilidad
- Confiabilidad
- Eficiencia

4.1.2.1 Funcionalidad

La funcionalidad no se mide directamente, por tanto, no es necesario evaluar un conjunto de características y capacidades del sistema.

Para el cálculo de la funcionalidad se utilizara la métrica de “Punto Función” (PF), para esto se debe determinar cinco características de dominios de información necesarias para el cálculo de la misma.

- 1. Número de Entradas del Usuario.** Se cuenta cada entrada del usuario que proporciona diferentes datos al sistema, en el caso del sistema se identificaron 15 entradas del usuario.
- 2. Número de Salidas del Usuario.** Se cuenta cada salida que proporciona la información del usuario, estas pueden ser informes, reportes, etc. Se identificaron 10 salidas del usuario en el sistema.
- 3. Número de Peticiones del Usuario.** Se cuentan la cantidad de entradas interactivas que producen la generación de respuestas (salidas) inmediatas del sistema. Se apreciaron 6 peticiones de usuario.
- 4. Número de Archivos.** Se cuenta cada archivo maestro lógico es decir un grupo lógico de datos que sean parte de la base de datos, o archivos independientes. Se contaron 15 archivos.
- 5. Número de Interfaces Externas.** Se cuenta todas las interfaces legibles por la maquina (por ejemplo: archivos de datos de disco) que se utilizan para transmitir información a otros sistemas. En este caso no existen interfaces externas.

La tabla 4.1 muestra las cinco características con el factor de ponderación medio para el cálculo de punto función.

Tabla 4. 1
Cálculo de Punto Función

PARÁMETROS DE MEDICIÓN	FACTOR DE PONDERACIÓN				
	CUENTA	SIMPLE	MEDIO	COMPLEJO	RESULTADO
Número de entradas de Usuario.	15	3	4	6	60

Número de salidas de usuario.	10	4	5	7	50
Número de peticiones de usuario.	6	3	4	6	24
Número de archivos.	15	7	10	15	150
Número de interfaces externas.	0	5	7	10	0
CUENTA TOTAL					284

Fuente: (Elaboración propia)

Los valores de la variable F_i , se obtienen tomando en cuenta la ponderación que muestra en la tabla 4.2.

Tabla 4. 2
Valores de Ajuste de Complejidad

COMPLEJIDAD	ESCALA
Sin importancia	0
Incidencia	1
Moderado	2
Medio	3
Significativo	4
Esencial	5

Fuente: (Elaboración propia)

Respondiendo a las preguntas en la tabla 4.3.

Tabla 4. 3
Ajuste de Complejidad del Sistema

FACTOR DE AJUSTE DE COMPLEJIDAD	PONDERACIÓN
1 ¿Requiere que el sistema realice copias de seguridad y recuperación flexible?	5

2	¿Se requiere comunicación de datos?	5
3	¿Existen funciones del procedimiento distribuido?	3
4	¿Es crítico el rendimiento?	2
<hr/>		
5	¿Se ejecutara el sistema en un entorno operativo existente y fuertemente utilizado?	5
6	¿Requiere el sistema entrada interactiva?	4
7	¿Requiere la entrada de datos interactiva que las transiciones de entrada se lleven a cabo sobre múltiples pantallas u operaciones?	3
8	¿Se actualiza los archivos maestros de forma interactiva?	4
9	¿Son complejos las entradas, las salidas, los archivos y las peticiones?	2
10	¿Es complejo el procedimiento interno?	3
11	¿Se ha diseñado el código para ser reutilizable?	5
12	¿Están incluidas en el diseño la conversión y la instalación?	3
13	¿Se ha desarrollado el sistema para soportar múltiples instalaciones en diferentes organizaciones?	3
14	¿Se ha diseñado la aplicación para facilitar los cambios y para ser fácilmente utilizado por el usuario?	5
TOTAL		52

Fuente: (Elaboración propia)

PF es una métrica que mide el tamaño y la complejidad del sistema, el software en términos de las funciones del usuario y se define con la formula siguiente.

$$PF = Cuenta_{(Total)} \times [0.65 + 0.01 \times \sum Fi]$$

Dónde:

- **PF** = Medida de Funcionalidad.
- **Cuenta Total** = Es la suma de valor de las entradas, salidas, peticiones, interfaces externas y archivos.
- **Grado de Confiabilidad** = Es la confiabilidad estimada del sistema.
- **Tasa de error** = Probabilidad subjetiva estimada del dominio de la información, este error estimado es de 1%.
- **Fi** = Son valores de ajuste de complejidad que toman los valores de la tabla 4.2 y que dan respuesta a las preguntas de la tabla 4.3.

Entonces

$$\sum f_i = 52$$

Para calcular los puntos de función se utiliza la siguiente relación, para un nivel de confianza del 65%.

$$PF_{Real} = Cuenta_{Total} \times [0.65 + 0.01 \times \sum f_i]$$

$$PF_{Real} = 284 \times [0.65 + 0.01 \times 52]$$

$$PF_{Real} = 332.28$$

Comparando con los intervalos:

$$300 < PF - \text{Optimo}$$

$$200 < PF < 300 - \text{Bueno}$$

$$100 < PF \leq 200 - \text{Suficiente}$$

$$PF \leq 100 - \text{Deficiente}$$

Comparamos los valores de funcionalidad del sistema con el punto de función máximo que se puede alcanzar es:

$$PF_{Real} = Cuenta_{Total} \times [0.65 + 0.01 \times 70]$$

$$PF_{Real} = 284 \times [0.65 + 0.01 \times 70]$$

$$PF_{\text{Real}} = 383.4$$

Con estos dos últimos se obtiene el porcentaje de funcionalidad del sistema:

$$PF = \frac{PF_{\text{Real}}}{PF_{\text{Esperado}}}$$

$$PF = \frac{332.28}{383.4}$$

$$PF = 0.866$$

$$PF = 0.866 \times 100 \%$$

$$PF = 86.6 \%$$

Entonces la funcionalidad del sistema alcanza un 87%.

4.1.2.2 Mantenibilidad

Es un conjunto de atributos relacionados con la factibilidad de extender, modificar o corregir errores en un sistema software.

Para medir la mantenibilidad del sistema se utilizan los índices de madurez del software

$$IMS = \frac{(MT - (Fc + Fa + Fe))}{MT}$$

Dónde:

MT = Número de módulos en la versión actual.

Fc = Número de módulos en la versión actual que han cambiado.

Fa = Número de módulos en la versión actual añadido.

Fe = Número de módulos en la versión anterior que se ha borrado.

Entonces.

Tabla 4.4. Información requerida por el IMS

}

Tabla 4. 4
Valores de ajuste de complejidad

INFORMACIÓN	VALORES OBTENIDOS
MT	10
Fc	1
Fa	0
Fe	0

Fuente: (Elaboración propia)

Remplazando en la ecuación se tiene:

$$IMS = \frac{(10 - (1 + 0 + 0))}{10}$$

$$IMS = \frac{9}{10}$$

$$IMS = 0.90$$

$$IMS = 0.90 \times 100 \%$$

$$IMS = 90 \%$$

A medida que el sistema se aproxima a 1 el producto se sitúa más estable según la siguiente relación:

75 % <= IMS <= 100 % Optima.

50 % <= IMS <= 75 % Buena.

25 % <= IMS <= 50 % Suficiente.

0 % <= IMS <= 25 % Deficiente.

Entonces se ve que el valor obtenido se encuentra en el primer intervalo, con esto se puede afirmar que el sistema tiene una mantenibilidad optima del 90 %.

4.1.2.3 Portabilidad

La portabilidad se refiere al esfuerzo necesario para transferir el programa de un entorno ya sea de hardware y/o software a otro, es una característica deseable de todo software. La portabilidad tiene la siguiente fórmula:

$$P = 1 - \frac{EP}{EI}$$

Dónde:

P = Portabilidad

EP = Número de días para portar el sistema

EI = Número de días para implementar el sistema

Entonces se tiene que:

$$EP = 0.5 \text{ días}$$

$$EI = 3 \text{ días}$$

Remplazando en la formula se tiene:

$$P = 1 - \frac{0.5}{3}$$

$$P = 0.833$$

$$P = 0.833 \times 100 \%$$

$$P = 83.3 \%$$

Lo que significa que existe un 83% de que el usuario instale el software con éxito, entonces la portabilidad es óptima.

4.1.2.4 Usabilidad

La usabilidad o facilidad de uso (FU), se calcula con la siguiente formula.

$$FU = \frac{\sum X_i}{n} \times 100$$

En la siguiente tabla, se calcula la usabilidad utilizando la escala de evaluación:

Tabla 4. 5
Evaluación de preguntas para calcular la usabilidad.

NRO.	PREGUNTAS	EVALUACIÓN (XI)
1	¿El sistema satisface los requerimientos de manejo de información?	5

2	¿Las salidas del sistema están de acuerdo a sus requerimientos?	5
3	¿Cómo considera el ingreso de datos del sistema?	5
4	¿Cómo considera los formularios que elabora el sistema?	4
5	¿El sistema facilita el trabajo que realiza?	5
	$\sum X_i$	24

Fuente: (Elaboración propia)

Calculando FU:

$$FU = \frac{\frac{24}{5} \times 100}{5}$$

$$FU = 96 \%$$

Por lo tanto, la facilidad de uso es del 96%.

4.1.2.5 Confiabilidad

La confiabilidad es la capacidad del software de mantener su nivel de rendimiento bajo las condiciones establecidas por un periodo de tiempo.

La confiabilidad del Software se mide con la siguiente formula:

$$R(t) = e^{-\lambda T}$$

Dónde:

R(t) = Confiabilidad del Sistema

λ = Error de tasa constante de fallas

T = Tiempo de operación del sistema (meses)

La tasa de error o la probabilidad de error que puede tener el sistema es del 0.5%, si se considera un tiempo de 12 meses para el funcionamiento del sistema.

Remplazando los valores en la fórmula:

$$R(t) = e^{-0.005\% \times 12}$$

$$R(t) = 0.94 \%$$

Por lo tanto, la confiabilidad del sistema es del 94% lo cual significa que en 12 meses el sistema mantendrá un rendimiento óptimo.

4.1.2.6 Eficiencia

La eficiencia es el conjunto de atributos que se relacionan con el nivel de performance del software y de la cantidad de recursos usados, bajo las condiciones establecidas, en tiempo y en recursos.

Tabla 4. 6
Evaluación de preguntas para calcular la usabilidad.

NRO.	PREGUNTAS	EVALUACIÓN
1	¿La respuesta es rápida al utilizar las funciones?	90
2	¿Tiene rendimiento de acuerdo a los factores que utiliza?	90
3	¿Responde adecuadamente cuando utiliza las funciones?	90
TOTAL		90

Fuente: (Elaboración propia)

El sistema se considera eficiente por la óptima utilización de los recursos en un 90%.

4.1.2.7 Resultado Final

En la siguiente tabla se muestra el resultado final de los factores de la norma ISO 9126

Tabla 4. 7
Resultado total de Calidad

FACTOR	RESULTADO
Funcionabilidad	87%
Mantenibilidad	90 %

Portabilidad	83 %
Usabilidad	96 %
Confiabilidad	94 %
Eficiencia	90 %
<hr/>	
EVALUACIÓN TOTAL DE CALIDAD	90 %
<hr/>	

Fuente: (Elaboración propia)

Entonces se pudo evidenciar que el sistema tiene un 90 % de calidad y por lo tanto el dicho sistema está en un rango muy aceptable.

4.2 COSTOS DEL SOFTWARE

4.2.1 Introducción

Una de las tareas más importantes en la planificación de los proyectos de software es la estimación, el cual consiste en determinar, con cierto grado de certeza, los recursos de hardware y software, costos, tiempo y esfuerzo necesarios para el desarrollo del sistema mediante COCOMO II, modelo de estimación de costos, mediante él se obtendrá el esfuerzo, tiempo y personal necesarios para el desarrollo del software, también se realizara un cálculo de los costos de la implementación del sistema para el centro médico San Simón.

4.2.2 Análisis de Costos del Sistema

Se debe calcular todos los costos pronosticados asociados al sistema utilizando el modelo constructivo COCOMO II, que están orientados a los puntos de función. De la tabla 4.1 que se vio en el anterior capítulo tenemos que:

$$\text{Cuenta Total} = 284$$

Así también de la tabla 4.3 se vio que el factor punto función es:

$$\text{Factor punto función} = 52$$

Para el cálculo del factor de complejidad técnica TCF, se toma en cuenta los datos de la tabla 4.3, para considerar la siguiente fórmula:

$$\text{TCF} = (0.65 + 0.01 \times 52)$$

$$TCF = 1,17$$

El procesamiento de datos del punto función se basa en la formula siguiente:

$$PF = \text{Cuenta Total} \times TCF$$

$$PF = 284 \times 1,17$$

$$PF = 332.28$$

Conversión de los puntos de función a KLDC.

Para determinar el esfuerzo nominal en el modelo COCOMO II los puntos función no ajustados tienen que ser convertidos a miles de líneas de código fuente considerando el lenguaje de implementación que se muestra en la tabla

Tabla 4. 8
Conversión de puntos función

FACTOR	NIVEL	FACTOR LDC/PF
C	2.5	128
Ansi Basic	5	64
Java	6	53
Ansi Cobol 74	3	107
Visual Basic	7	46
ASP	9	36
PHP	11	29
Visual C++	9.5	34

Fuente: (Elaboración propia)

Así con el valor que extraemos de la tabla anterior del valor del factor LDC/PF tenemos que:

$$LDC = PF \times \text{Factor} \frac{LDC}{PF}$$

$$LDC = 332.28 \times 29$$

$$LDC = 9\,636$$

Las líneas de código en su totalidad son 9992,24, entonces el número estimado de líneas de código distribuidas en miles es:

$$KLCD = \frac{LDC}{1000}$$

$$KLCD = \frac{9\,636}{1000}$$

$$KLCD = 9,636$$

Por tanto, existen 9,636 líneas de código distribuidas para el proyecto. Ahora se aplican las formulas básicas de esfuerzo, tiempo calendario y personal requerido.

Las ecuaciones del COCOMO básico tienen la siguiente forma:

$$E = a_b(KLCD)^{b_b}$$

$$D = c_b D^{d_b}$$

Dónde:

E = Esfuerzo aplicado en personas por mes.

D = Tiempo de desarrollo en meses cronológicos.

KLDC = Número estimado de líneas de código distribuidas (en miles).

Tabla 4. 9
Relación de valores del modelo COCOMO

PROYECTO DE SOFTWARE	a_b	b_b	c_b	d_b
Orgánico	2,4	1,05	2,5	0,38
Semi-Acoplado	3	1,12	2,5	0,35
Empotrado	3,6	1,2	2,5	0,32

Fuente: (Elaboración propia)

En la tabla anterior se muestran los tipos de proyectos de software, como este es un proyecto intermedio, en tamaño y complejidad, se elige orgánico.

De esta manera se hallara el esfuerzo aplicado en personas por mes remplazando los valores en la fórmula:

$$E = a_b(KLCD)^{bb}$$

$$E = 2,4 \times (9,636)^{1,05}$$

$$E = 25,90$$

De la misma manera hallamos el tiempo de desarrollo en meses cronológico remplazando los valores en la fórmula:

$$D = c_b D^{db}$$

$$D = 2,5 \times 25,90^{0,38}$$

$$D = 8,60$$

El personal requerido, en este caso el número de programadores (Nº Prog) se obtiene con la siguiente fórmula:

$$N^{\circ} \text{ Prog} = \frac{E}{D}$$

$$N^{\circ} \text{ Prog} = \frac{25,90}{8,60}$$

$$N^{\circ} \text{ Prog} = 3,01$$

$$N^{\circ} \text{ Prog} = 3,01 \text{ programadores}$$

$$N^{\circ} \text{ Prog} = 3 \text{ programadores}$$

El salario de un programador puede oscilar entre los 300 \$us, cifra que es tomada en cuenta para la estimación siguiente:

$$\text{Costo } SD_{\text{Persona}} = N^{\circ} \text{ Prog} \times S_{\text{Prog}}$$

Dónde:

Costo SD_{Persona} = Costo del software desarrollado por persona

$N^{\circ} \text{ Prog}$ = Número de programadores (3)

S_{Prog} = Salario del programador en dólares (antes 500 ahora 300)

Entonces se tendrá:

$$\text{Costo } SD_{\text{Persona}} = 3 \times 300 \text{ \$us}$$

$$\text{Costo } SD_{\text{Persona}} = 900 \text{ \$us}$$

De donde se concluye que para el desarrollo del “Sistema Web para el Seguimiento de Diagnóstico y Tratamiento de Pacientes” Caso: Centro Medico “San Simón” es necesario contar con 3 programadores durante 8 meses para su respectivo desarrollo, el tendrá un costo de 7.200 \$us (dólares americanos).

4.2.3 Costo Total del software

El costo total del software se lo obtiene de la sumatoria del costo de: desarrollo, implementación y elaboración del proyecto:

Tabla 4. 10
Costos Total del Proyecto

DETALLES DE INVERSIÓN	IMPORTE (\$us)
Costo de desarrollo 8 (Meses)	7 200
COSTO TOTAL	7 200 \$us

Fuente: (Elaboración propia)

Por lo tanto, el costo total del proyecto incluido los costos de implementación serán aproximadamente 7.200 \$us (dólares americanos) o su equivalente en bolivianos 49 910 bs.

4.3 SEGURIDAD DEL SOFTWARE

4.3.1 Introducción

La seguridad de la información es el conjunto de medidas preventivas y reactivas de las organizaciones y de los sistemas tecnológicos que permiten resguardar y proteger la información buscando mantener la confidencialidad, la disponibilidad e integridad del software.

4.3.2 Seguridad de base de datos

Se usó como base de datos MariaDB. En cuanto a la forma de resguardo se realiza:

- Cuando una acción del usuario en el sistema requiere o solicita algunos registros de la base de datos, existe una conexión segura para esta acción.
- Para la seguridad de datos del sistema se tienen registrado de nombre de usuario y contraseña de acceso, según su nivel de acceso pueda realizar actividades en el sistema.

La información en una empresa es muy valiosa, por tanto su resguardo es fundamental, la conexión a la base de datos y el cierre de la conexión es de forma automática.

En cuanto a las amenazas de SQL Injection que es una de las más comunes amenazas, el framework Laravel incorpora un ORM para que el desarrollador deje de preocuparse por este tipo de ataques ya que el ORM está basado en una capa de objetos y así no se es capaz de interpretar el lenguaje SQL.

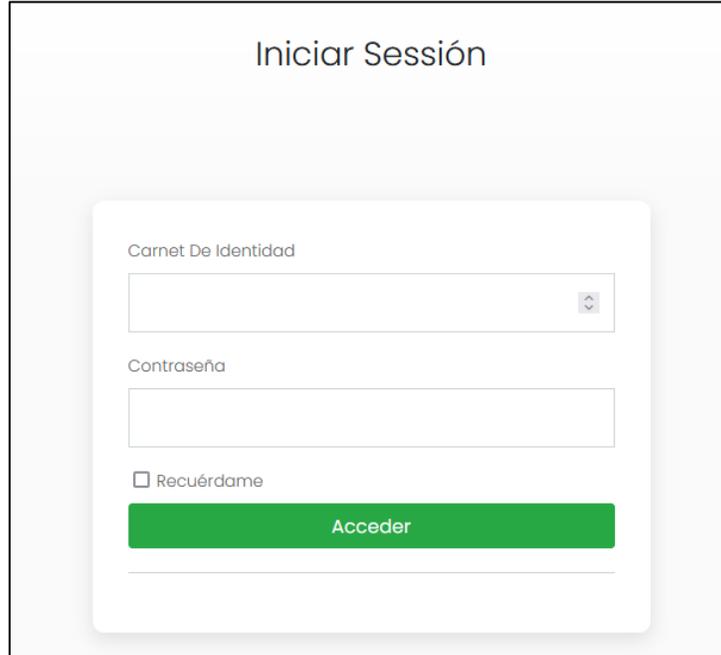
4.3.3 Seguridad de autenticación

Este control se refiere al control de sesión o verificación de la autenticación de un usuario con nombre de usuario y una contraseña, que ya anteriormente asignados.

Mientras el usuario ingresa la contraseña, esta no se puede mostrar en pantalla, también cabe resaltar que la contraseña de cada usuario este encriptado por el algoritmo AES- 256 y 128 que nos proporciona Laravel.

En la siguiente imagen observamos la pantalla de autenticación del Centro Médico "San Simón".

Figura 4. 1
Autenticación de Usuario



The image shows a login form titled "Iniciar Sesión". It contains the following elements:

- A label "Carnet De Identidad" above a text input field with a dropdown arrow on the right.
- A label "Contraseña" above a text input field.
- A checkbox labeled "Recuérdame".
- A green button labeled "Acceder".
- A horizontal line below the button.

Fuente: (Elaboración propia)

4.3.4 Seguridad de la aplicación

Se desarrolla un módulo de control de acceso al sistema para la restricción del acceso a usuario no autorizado. Este módulo verifica y autoriza a los usuarios por medio de permisos que son otorgados por el adiestrador del sistema, haciendo uso de las sesiones de PHP. Se realiza el registro del usuario que modifica la información la base de datos, para esto se registra en cada tabla el identificar del usuario que modifica la información.

5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Una vez finalizado el proyecto de grado Sistema Web para el Seguimiento de Diagnóstico y Tratamiento de Pacientes, Caso: Centro Medico “San Simón”, se ha logrado alcanzar el objetivo principal planteado que indicaba el desarrollar un sistema web que mejore el control de los pacientes, cumpliendo con las necesidades de la clínica.

Tomando en cuenta los objetivos planteados se llega a las siguientes conclusiones:

- Con el proyecto ejecutado se ha cumplido con el objetivo planificado, centralizar la información de los pacientes del Centro Médico San Simón en una base de datos con soporte informático que posibilite un control adecuado de la información.
- Se logró almacenar la información de los pacientes del Centro Médico San Simón, evitando su pérdida, uso inapropiado y facilitando su acceso y distribución.
- Se logró agilizar las tareas de búsqueda de información de los pacientes del Centro médico “San Simón” de la ciudad de El Alto.
- Se logró diseñar reportes específicos requeridos por el personal médico y administrativo.

5.2 RECOMENDACIONES

A partir de este trabajo se propone las siguientes recomendaciones, con el fin de buscar el mejoramiento del sistema:

- Con respecto al análisis y diseño del sistema, cuando se requiera la ampliación y creación de nuevos módulos, se recomienda lo primero revisar la documentación para poder tomar una buena decisión, ya que el sistema presenta elementos reutilizables que podrían ser utilizados en los módulos nuevos.

- Se recomienda para módulos futuros con características utilizar el framework Laravel para exista compatibilidad en el desarrollo del código.
- Se recomienda a la clínica, implementar, utilizar y administrar el sistema de acuerdo al manual de usuario.
- Por último, se recomienda una revisión periódica cada cierto tiempo lo cual ayudará a la eficiencia y un funcionamiento adecuado del sistema.

Bibliografía

- Acedo, J. I. (04 de 05 de 2015). *Apuntes de Programación*. Obtenido de <http://programacion.jias.es/2015/05/web-%C2%BFque-es-el-framework-bootstrap-ventajas-desventajas/>
- Alvarez, M. A. (17 de 10 de 2014). *DesarrolloWeb.com*. Obtenido de <https://desarrolloweb.com/articulos/composer-gestor-dependencias-para-php.html>
- Andrés. (10 de 02 de 2014). *Guiadev*. Obtenido de <https://guiadev.com/que-framework-php-me-conviene-utilizar/>
- Auladell, G. (23 de 12 de 2016). *¿Qué es MariaDB?* Obtenido de Drauta agencia digital: <https://www.drauta.com/que-es-mariadb>
- Calderon, S. M. (2017). *PORTAL WEB PARA LA GESTION DE CONSULTAS MEDICAS CON APLICACION DE TECNOLOGIAS MOVILES*. El Alto, Bolivia.
- Carlos Miguez López Leon, G. D. (2016). *Sistema Informatico de Adiministración de Pacientes y Control de Citas e Inventario para la Clínica ISTA*.
- CONDE, W. C. (2015). *AUDITORIA MEDICA, COMO INSTRUMENTO DE LA MEJORA DE LA CALIDAD EN LA ATENCION A LA MUJER GESTANTE Y AL RECIEN NACIDO, EN EL SERVICIO DE ALTO RIESGO OBSTETRICO DEL HOSPITAL DE LA MUJER*. La Paz, Bolivia.
- Condori, E. A. (2015). *IMPLEMENTACION DEL SISTEMA WEB PARA LA GESTION Y CONTYROL DE HHISTORIALÑ CLINICO DE PACIENTES*. El Alto, Bolivia.
- Conejos, J. (27 de 11 de 2017). *Blog de José Conejos*. Obtenido de <https://joseconejow.wordpress.com/2017/11/27/cambia-el-idioma-del-editor-atom/>
- DAGNINO, M. A. (2013). *DESARROLLO DE UN SISTEMA WEB DE CONTROL DE CITAS, PARA UN HOSPITAL DEL DÍA*. Quito, Ecuador.
- Flores, R. L. (2014). *Sistema de administración y control de historiales clínicos para los consultorios de la UMSA*. La Paz.
- GDR Soluciones. (2017). *Preguntas Frecuentes - FAQ*. Obtenido de <https://facturacion.gdrsoluciones.com/knowledgebase.php?action=displayarticle&id=26&language=spanish>
- LLipe, D. A. (2014). *Sistema web de registro y seguimiento de pacientes, elaboración y emisión de análisis efectuados, registro y guía médica, farmacéutica y hospitalaria*. La Paz.
- Nina, M. A. (2013). *Sistema de gestión hospitalario modulo consulta externa*. La Paz.

Pereira, M. A. (2018). *SISTEMA DE RECORDATORIO PREVIO PARA INCREMENTAR LA ASISTENCIA DE PACIENTES A CONSULTORIOS EXTERNOS DE ESPECIALIDADES MÉDICAS EN EL HOSPITAL "CENTRAL DE EMERGENCIAS NUEVA ESPERANZA" DE LA CIUDAD DE EL ALTO EN LA GESTIÓN 2018*. La Paz, Bolivia.

The PHP Group. (2018). *¿Qué es PHP?* Obtenido de <http://php.net/manual/es/intro-what-is.php>

VALDIVIA, L. A. (2017). *Implementación del Sistema Web para la Gestión de Citas Horarias en el Hospital María Auxiliadora*. Lima, Peru.

Zepeda, R. (2015). *EL INFORMÁTICO.COM*. Obtenido de <http://blogdelinformatico-reizer.blogspot.com/2015/11/que-es-xampp.html>

ANEXOS

MANUAL DE USUARIO



SISTEMA WEB PARA EL SEGUIMIENTO DE
DIAGNÓSTICO Y TRATAMIENTO DE
PACIENTES

POR JULIO CESAR LECOÑA LAURA

Acceso al Sistema:

Para ingresar al sistema web, debe dirigirse a la dirección del sistema en internet: <https://sansimon.clinic> .

Una vez dentro de la dirección usted visualizará el inicio de acceso al sistema



Para iniciar sesión el medico como administrado deberá dirigirse al enlace "ingresar al sistema", donde se abrirá la siguiente pantalla, y para iniciar sesión deberá llenar los 2 campos obligatoriamente y al finalizar click en acceder.

The image shows a login form with a white background and rounded corners. It contains the following elements: a label 'Carnet De Identidad' above a text input field with a dropdown arrow on the right; a label 'Contraseña' above a text input field; a checkbox labeled 'Recuérdame'; and a dark grey button with the text 'Acceder' in white. There is a thin horizontal line below the button.

Una vez ingresado al sistema se desplegará un menú con distintas opciones dependiendo del tipo de usuario que ingresa.

- **SESION INICIADA CON ROL ADMINISTRADOR**

Donde se le mostrará la siguiente pantalla con todos los componentes que podrá administrar el administrador y que se detallan a continuación.



El usuario "Administrador" tiene todos los permisos sobre el sistema.

También el usuario administrador tiene un resumen general de toda la información en el panel principal.

LISTADO DE PACIENTES PARA EL ADMINISTRADOR

Para poder administrar a los pacientes el administrador debe dirigirse al menú y hacer Click en la opción pacientes, donde se visualizará la siguiente ventana.

PRINCIPAL

- Inicio

CONSULTAS MEDICAS

- Pacientes

ADMINISTRACION

- Medicos
- Enfermeras
- Especialidades

MI CUENTA

- Mi Perfil
- Cerrar sesión

Julio Cesar Cerrar sesión

Pacientes

Buscar

PDF Excel CSV Copiar Imprimir

#	Imagen	Nombres	Apellidos	Carnet	Nacimiento	Edad	Sexo	Telefono	Correo
1		Darwin	Callizaya Flores	9083573	2001-03-11	20	Masculino	76394633	darwind@gmail.com
2		Oliver	Callizaya Ramires	7853936	2006-02-02	15	Masculino	793457322	oliverc@mail.com
3		Carlos	Caique Ramirez	9687537	1993-02-01	28	Masculino	79354388	carlos33@gmail.com

LISTADO DE MEDICOS PARA EL ADMINISTRADOR

El usuario “Administrador” tiene todos los permisos sobre el sistema. También el usuario administrador tiene un resumen general de toda la información en el panel principal.

PRINCIPAL

- Inicio

CONSULTAS MEDICAS

- Pacientes

ADMINISTRACION

- Medicos
- Enfermeras
- Especialidades

MI CUENTA

- Mi Perfil
- Cerrar sesión

Julio Cesar Cerrar sesión

Medicos

Buscar

PDF Excel CSV Copiar Imprimir

+ Nuevo

#	Imagen	Nombres	Apellidos	Carnet	Nacimiento	Edad	Sexo	Telefono	Correo
1		Jhonny	Choque Mamani	90486011	1981-04-04	40	Masculino	76795433	jhonnycm@gmail.com
2		Alexander	Quispe Tonconi	9043284	1995-02-11	26	Masculino	76304922	alexcam@gmail.com
3		Carol	Gutierrez Calle	9034820	1991-12-22	29	Femenino	76204506	carol23@gmail.com
4		Carol	Gutierrez Calle	9034826	1991-12-22	29	Femenino	76204506	caroltt@gmail.com

LISTADO DE ENFERMEROS PARA EL ADMINISTRADOR

El usuario “Administrador” tiene habilitada todas las funciones disponibles, tales como: Nuevo, Actualizar, Ver y Desactivar.

The screenshot shows the 'Enfermeros/as' management page. It features a search bar, a '+ Nuevo' button, and a table with columns for #, Imagen, Nombres, Carnet, Nacimiento, Edad, Sexo, Telefono, and Acciones. The table contains four entries for nurses.

#	Imagen	Nombres	Carnet	Nacimiento	Edad	Sexo	Telefono	Acciones
1		Rayner Condori	5777777777	2015-02-11	6	Masculino	456789000	
2		Rolando Calle Quispe	9473045	1987-07-03	34	Masculino	763984977	
3		Neymar Callizaya Ramos	9064570	1995-03-03	26	Masculino	76938599	
4		Carmen Aquino Gutierrez	9037491	1985-03-01	36	Femenino	76394944	

LISTADO DE ESPECIALIDADES PARA ADMINISTRADOR

De igual forma el usuario “Administrador” tiene habilitada todas las funciones disponibles, tales como: Nuevo, Actualizar, Ver y Desactivar.

The screenshot shows the 'Especialidades' management page. It features a search bar, a '+ Nuevo' button, and a table with columns for #, Imagen, Nombre, Condicion, and Acciones. The table contains four entries for specialties, all marked as 'Activo'.

#	Imagen	Nombre	Condicion	Acciones
1		Medicina General	Activo	
2		Traumatologia	Activo	
3		Ginecologia	Activo	
4		Dentista	Activo	

- **SESION INICIADA COMO DOCTOR**

PRINCIPAL

- Inicio
- CONSULTAS MEDICAS
- + Nueva consulta
- MI CUENTA
- Mi Perfil
- Cerrar sesión

Consultas

+ Nuevo paciente

Buscar

PDF Excel CSV Copiar Imprimir

#	Carnet	Nombre paciente	Genero	Edad	Acciones
1	9083573	Darwin Callizaya Flores	Masculono	20	
2	7853936	Oliver Callizaya Ramires	Masculono	15	
3	9687537	Carlos Colque Ramirez	Masculono	28	
4	9034916	Teodoro Villegas Mamani	Masculono	34	
5	2542452	Jeannette Inofuentes Gonzales	Femenino	35	

1 Muestra botón de opción de registrar nuevo paciente y hacer llenado del siguiente formulario que se muestra a continuación.

Nombres

Apellidos

Carnet de identidad

Fecha de nacimiento

Sexo

Telefono/Celular

Correo electronico

Datos vitales

<input checked="" type="checkbox"/> Edad (años) <input type="text" value="Ingrese el nombre complet"/>	<input checked="" type="checkbox"/> Talla (cm) <input type="text" value="Ingrese el nombre complet"/>	<input checked="" type="checkbox"/> Peso (Kg) <input type="text" value="Ingrese el nombre complet"/>	<input checked="" type="checkbox"/> Temperatura (°C) <input type="text" value="Ingrese el nombre complet"/>
<input checked="" type="checkbox"/> Frecuencia cardiaca (x min.) <input type="text" value="Ingrese el nombre complet"/>	<input checked="" type="checkbox"/> Presion arterial <input type="text" value="Ingrese el nombre completo"/>	<input checked="" type="checkbox"/> Frecuencia respiratoria (%) <input type="text" value="Ingrese el nombre complet"/>	<input checked="" type="checkbox"/> Enfermero/a <input type="text" value="Rayner Condori"/>

2 Opción de generar reportes de todos los campos que se muestran en la tabla en todos los formatos que se visualizan en el sistema.

3 Buscador de pacientes

4 Opción de ver pacientes

Pacientes ← atrás

Datos clínicos

Estevan Callisaya Lopez

Edad:	28 años	Peso:	56 kg
Talla:	160 Cm	Temperatura:	23 ° C
Frecuencia cardiaca:	34 Ppm	Presion arterial:	33 mmHg
Frecuencia respiratoria:	43 Frec. Resp.		

Actualizado el 2021-11-12 18:27:09

Nueva medicion
Ver historial

Datos personales

Historial clinico Historial + Nuevo

2021-11-15 15:16

No tiene historias clinicas

Registre una nueva historia haciendo click en el boton Nuevo

El botón **atrás** tiene la funcionalidad de retroceder a la anterior página

El botón **nueva medición** tiene la funcionalidad de medir los signos vitales del paciente

El Botón **ver historial** tiene la funcionalidad de imprimir el reporte del paciente junto a sus signos vitales como se puede ver en la siguiente imagen.

Historial signos vitales de: Darwin Callizaya Flores ← Atras

Buscar

PDF Excel CSV Copiar Imprimir

#	Fecha	Edad	Talla	Peso	Temperatura	Frecuencia cardiaca	Presion arterial	Frecuencia respiratoria	Enfermero
1	2021-11-19 17:31:08	20 años	170 cm	45 kg	34 °C	80 Ppm	81 mmHg	70 Frec. Resp.	Rolando Calle Quispe

Mostrando 1 al 1 de 1 registros Previous 1 Next

El botón de **Historial** muestra el tratamiento del paciente.

El botón **nuevo** crea una nueva consulta como se puede apreciar en esta imagen.

Fecha: 2021-11-24 Título (SUBJETIVO): Ingrese los subjetivo Medico: Carol Gutierrez Calle

Tipo de atencion:

Consulta general Plan de tratamiento Vacuna Examen

Sintomas (OBJETIVO):

Diagnostico (ANALISIS):

Tratamiento (PLAN DE ACCION):

Exámenes (COMPLEMENTARIOS):

[Registrar consulta](#)

- **SESION INICIADA CON ROL DE ENFERMEROS**

Listado de pacientes correspondiente a cada enfermero

 Neymar [Cerrar sesión](#)

Pacientes [+ Nuevo](#)

Buscar:

[PDF](#) [Excel](#) [CSV](#) [Copiar](#) [Imprimir](#)

#	Imagen	Nombres	Apellidos	Carnet	Nacimiento	Edad	Sexo	Telefono	Correo	Acciones
1		Darwin	Callizaya Flores	9083573	2001-03-11	20	Masculino	78394633	darwind@gmail.com	Ver Editar
2		Oliver	Callizaya Ramires	7853936	2006-02-02	15	Masculino	793457322	oliverc@mail.com	Ver Editar
3		Carlos	Colque Ramirez	9887537	1993-02-01	28	Masculino	79354388	carlos33@gmail.com	Ver Editar
4		Teodoro	Villegas Mamaní	9034916	1987-08-01	34	Masculino	74376173	teo123@gmail.com	Ver Editar
5		Jeannette	Inofuentes Gonzales	2542452	1985-12-02	35	Femenino	61100564	inofuentes@mail.com	Ver Editar

Aquí se hace el registro de un nuevo paciente

Nombres
 Ingrese el nombre completo

Apellidos
 Ingrese los apellidos

Carnet de identidad
 Ingrese el numero de carnet

Fecha de nacimiento
 dd / mm / aaaa

Sexo
 Masculino

Telefono/Celular
 Ingrese numero de telefono o celular

Correo electronico
 Ingrese el correo electronico

Datos vitales

✓ **Edad (años)**
 Ingrese el nombre completo

✓ **Talla (cm)**
 Ingrese el nombre completo

✓ **Peso (Kg)**
 Ingrese el nombre completo

✓ **Temperatura (°C)**
 Ingrese el nombre completo

✓ **Frecuencia cardiaca (x min.)**
 Ingrese el nombre completo

✓ **Presion arterial**
 Ingrese el nombre completo

✓ **Frecuencia respiratoria (%)**
 Ingrese el nombre completo

♀ **Enfermero/a**
 Neymar Callizaya Ramos

Aquí el enfermero tiene el acceso limitado en cuanto al tratamiento a los pacientes.

Pacientes ← atras

Datos clinicos

Darwin Callizaya Flores			
Edad:	20 años	Peso:	45 kg
Talla:	170 Cm	Temperatura:	34 ° C
Frecuencia cardiaca:	80 Ppm	Presion arterial:	81 mmHg
Frecuencia respiratoria:	70 Frec. Resp.		
Actualizado el 2021-11-19 17:31:08			
Nueva medicion		Ver historial	

Datos personales

Aquí se hace la nueva medición de los datos vitales al paciente cuando viene más de una vez a la clínica

Datos vitales

Edad
 Talla
 Peso
 Temperatura

Frecuencia cardiaca
 Presion arterial
 Frecuencia respiratoria
 Enfermero/a

Registrar medicion

Aquí se ve el historial del paciente

Buscar

PDF Excel CSV Copiar Imprimir

#	Fecha	Edad	Talla	Peso	Temperatura	Frecuencia cardiaca	Presion arterial	Frecuencia respiratoria	Enfermero
1	2021-11-19 17:31:08	20 años	170 cm	45 kg	34 °C	80 Ppm	81 mmHg	70 Frec. Resp.	Rolando Calle Quispe

Mostrando 1 al 1 de 1 registros

Previous 1 Next

Editar datos de los pacientes

Nombres
 Apellidos

Carnet de identidad
 Fecha de nacimiento

Sexo
 Telefono/Celular

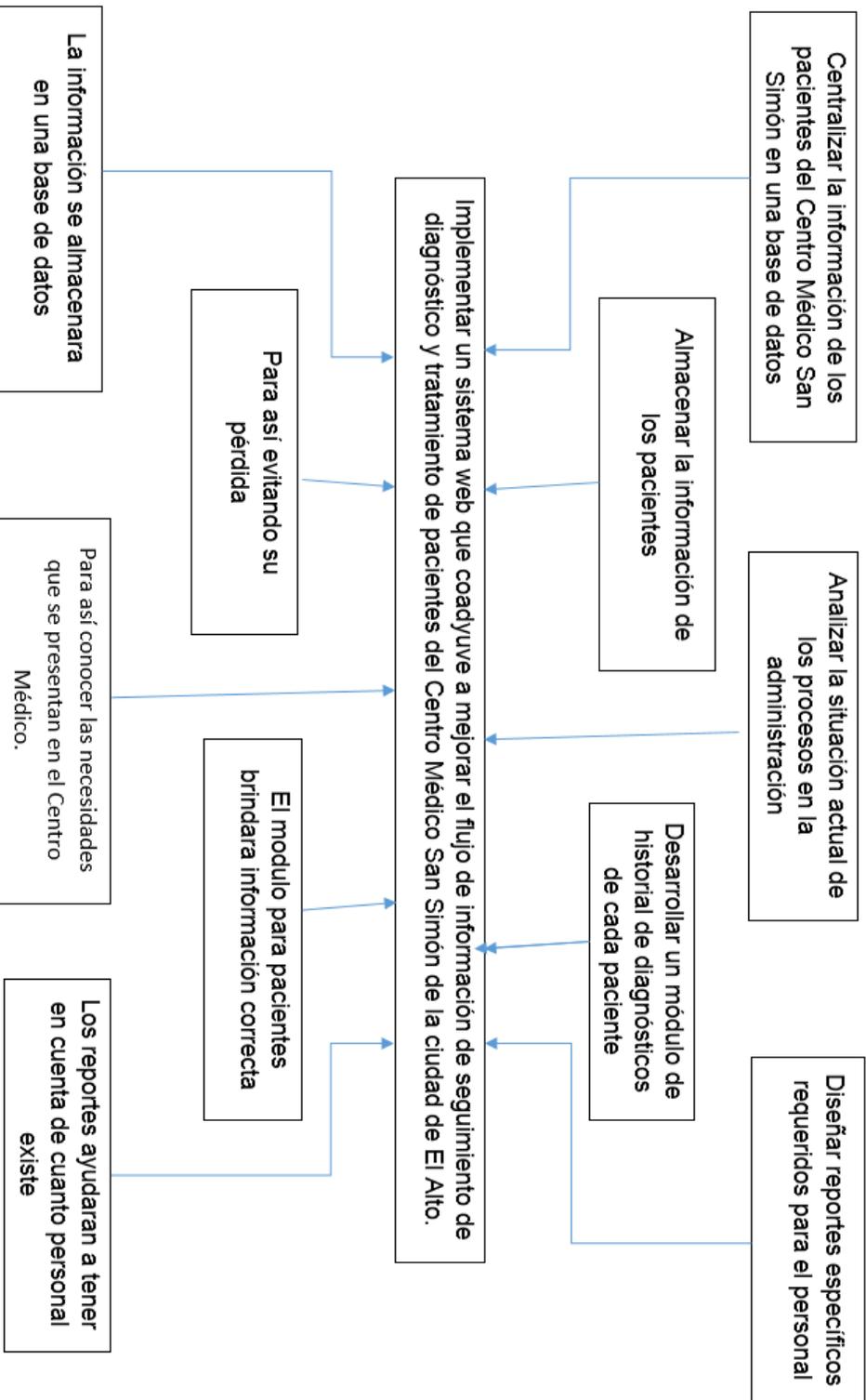
Correo electronico

Datos vitales

Edad (años)
 Talla (cm)
 Peso (Kg)
 Temperatura (°C)

Frecuencia cardiaca (x min.)
 Presion arterial
 Frecuencia respiratoria (%)
 Enfermero/a

ARBOL DE OBJETIVOS



ARBOL DE PROBLEMAS

