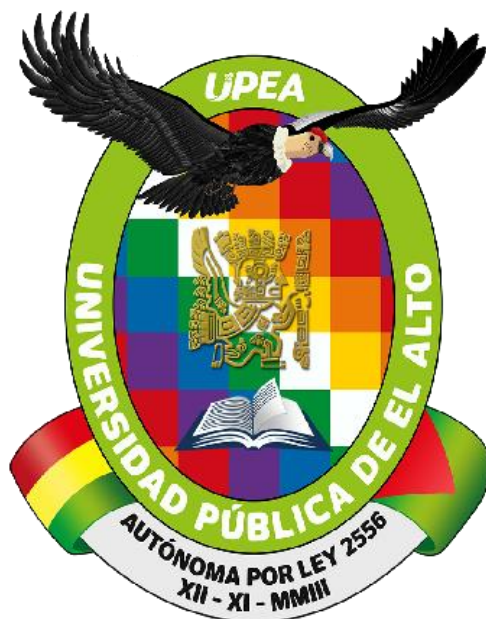


UNIVERSIDAD PÚBLICA DE EL ALTO

CARRERA INGENIERÍA DE SISTEMAS



PROYECTO DE GRADO

**SISTEMA DE ADMINISTRACIÓN Y SEGUIMIENTO DE HISTORIALES CLÍNICOS
ODONTOLÓGICOS**

“CASO: CENTRO ODONTOLÓGICO ZONADENTAL”

Para Optar al Título de Licenciatura en Ingeniería de Sistemas

MENCIÓN: INFORMÁTICA Y COMUNICACIONES

Postulante: Univ. Luis Pablo Alanoca Canaviri

Tutor Metodológico: Lic. Ing. Dionicio Henry Pacheco Ríos

**Tutor Revisor: M. Sc. Lic. Ing. Juan Fernando Chambi
Guachalla**

Tutor Especialista: Ing. Walter Emilio Paco Siles

EL ALTO – BOLIVIA

2023

DECLARACIÓN JURADA DE AUTENTICIDAD Y RESPONSABILIDAD

Yo **Luis Pablo Alanoca Canaviri** estudiante con C.I. 9212192 LP mediante la presente **declaro** de manera pública que la propuesta del **TRABAJO DE PROYECTO DE GRADO** titulado “**SISTEMA DE ADMINISTRACIÓN Y SEGUIMIENTO DE HISTORIALES CLÍNICOS ODONTOLÓGICOS CASO: CENTRO ODONTOLÓGICO ZONADENTAL**” es original, siendo resultado de mi trabajo personal y no constituye una copia o replica de trabajos similares elaborados.

Autorizo la publicación del resumen de mi propuesta en internet y me comprometo a responder a todos los cuestionamientos que se desprenden de su lectura.

Asimismo, me hago responsable ante la universidad o terceros, de cualquiera irregularidad o daño que pudiera ocasionar, por el incumplimiento de lo declarado.

De identificarse falsificación, plagio, fraude, o que el **PROYECTO DE GRADO** haya sido publicado anteriormente; asumo las consecuencias y sanciones que de mi acción se deriven, responsabilizándome por todas las cargas legales que se deriven de ello sometiéndome a las normas establecidas y vigentes de la Carrera de Ingeniería de Sistemas de la Universidad Pública de El Alto.

El Alto, noviembre del 2023.

.....
Univ. Luis Pablo Alanoca Canaviri

C. I. 9212192 LP

alancapl@gmial.com

DEDICATORIA

El presente proyecto de grado, está dedicada a mi familia y amigos, mismas que me motivaron para realizar un trabajo que pueda reflejar una problemática presente en nuestro entorno social, además de brindarnos su apoyo incondicional.

AGRADECIMIENTOS:

Está proyecto de grado es resultado de la colaboración y el esfuerzo generoso de muchas personas, quienes nos apoyaron con su guía en el desarrollo del presente trabajo. Por ello, se quiere reconocer y agradecer a:

A la Universidad Pública de El Alto, particularmente a la carrera de Ingeniería de Sistemas, donde recibí la formación profesional.

Finalmente, a todas aquellas personas que, de una manera u otra forma, colaboraron y ayudaron en la realización de este proyecto de grado.

Índice General

CAPÍTULO I	1
1 MARCO PRELIMINAR	1
1.1 INTRODUCCIÓN.....	1
1.2 ANTECEDENTES.....	2
1.2.1 Antecedentes de la institución	2
1.2.2 Proyectos similares.....	3
1.1.1.1 Proyectos similares internacionales.....	3
1.2.2.1 Proyectos similares locales	3
1.3 PLANTEAMIENTO DEL PROBLEMA.....	4
1.3.1 Problema general	4
1.3.2 Problemas específicos.....	5
1.4 FORMULACIÓN DEL PROBLEMA	5
1.5 OBJETIVOS.....	6
1.5.1 Objetivo general.....	6
1.5.2 Objetivos específicos	6
1.6 JUSTIFICACIÓN.....	6
1.6.1 Justificación técnica.....	6
1.6.2 Justificación económica.....	7
1.6.3 Justificación social	7
1.7 METODOLOGÍA.....	7
1.7.1 Método de ingeniería.....	7
1.7.1.1 Metodología de desarrollo	7
1.7.1.2 Métrica de calidad al software	8
1.7.1.3 Estimación de costo.....	8
1.7.1.4 Seguridad	8
1.7.1.5 Pruebas al software	9
1.8 HERRAMIENTAS	9
1.8.1 Gestor de base de datos MariaDB.....	9
1.8.2 Framework codeIgniter	10
1.8.3 HTML (lenguaje de marcas de hipertexto).....	10
1.8.4 Framework bootstrap.....	10

1.8.5	MagicDraw 16.0.....	10
1.8.6	JQuery	11
1.8.7	PHP	11
1.8.8	Servidor apache.....	12
1.9	LÍMITES Y ALCANCES	12
1.9.1	Limites	12
1.9.2	Alcances	12
1.10	APORTES	13
CAPÍTULO II		15
2	MARCO TEÓRICO	15
2.1	INTRODUCCIÓN.....	15
2.2	DATO.....	15
2.3	INFORMACIÓN	15
2.4	SISTEMA.....	16
2.4.1	Tipos de sistemas.....	17
2.4.1.1	Sistemas conceptuales o abstractos	17
2.4.1.2	Sistemas reales o materiales.....	17
2.5	SISTEMA WEB.....	17
2.6	SISTEMA DE CONTROL	18
2.7	CONTROL	19
2.8	HISTORIAL CLÍNICO	19
2.9	HISTORIA CLÍNICA ODONTOLÓGICA	19
2.10	ODONTOGRAMA.....	20
2.11	INGENIERÍA DE SOFTWARE.....	20
2.11.1	Proceso software	20
2.12	INGENIERIA WEB.....	21
2.12.1	Usabilidad	22
2.13	METODOLOGÍA OOHDM	23
2.14	ARQUITECTURA CLIENTE SERVIDOR.....	30
2.14.1	Características.....	30
2.14.2	Ventajas.....	31
2.14.3	Desventajas.....	32

2.14.4	Seguridad de modelo cliente /servidor.....	32
2.14.5	Arquitectura de dos y tres niveles cliente /servidor.....	32
2.14.5.1	Arquitectura dos niveles	32
2.14.5.2	Arquitectura tres niveles	33
2.14.6	Comparación entre ambos tipos de arquitecturas	34
2.15	SEGURIDAD DE APLICACIONES WEB.....	36
2.16	MODELO VISTA CONTROLADOR (MVC).....	37
2.16.1	Modelo.....	37
2.16.2	Vista.....	37
2.16.3	Controlador	37
2.16.4	Características del modelo vista controlador	37
2.17	MÉTRICAS DE CALIDAD AL SOFTWARE	38
2.17.1	Métricas de calidad del modelo de ISO-9126.....	39
2.17.1.1	Funcionalidad	40
2.17.1.2	Fiabilidad	40
2.17.1.3	Usabilidad.....	41
2.17.1.4	Portabilidad.....	41
2.17.1.5	Mantenibilidad.....	42
2.17.1.6	Eficiencia	42
2.18	PRUEBAS AL SOFTWARE	43
2.18.1	Pruebas de caja negra.....	43
2.18.2	Pruebas de caja blanca	44
2.18.3	Pruebas de estrés.....	44
2.19	ESTIMACIÓN DE COSTOS	45
2.19.1	Cocomo II	45
2.19.1.1	Modelo básico.....	45
2.19.1.2	Modelo intermedio	45
2.19.1.3	Modelo detallado	46
2.20	HERRAMIENTAS	47
2.20.1	Herramientas de desarrollo	47
2.20.1.1	Gestor de base de datos Mariadb.....	47

2.20.1.2	Lenguaje de programación PHP	48
2.20.1.3	Framework Codeigniter	49
2.20.1.4	La estructura de codeigniter	49
2.20.2	Herramientas de diseño.....	50
2.20.2.1	Framework bootstrap.....	50
2.20.2.2	CSS (Cascading Style Sheets).....	50
2.20.2.3	HTML (Hypertext Markup Language)	51
2.20.2.4	Ajax.....	51
2.20.2.5	JavaScript.....	51
2.20.2.6	Jquery.....	52
CAPÍTULO III		54
3	MARCO APLICATIVO	54
3.1	INTRODUCCIÓN.....	54
3.2	ESQUEMA DEL SISTEMA	54
3.3	DESARROLLO DE LA METODOLOGÍA OOHDM.....	54
3.3.1	Obtención de requerimientos.....	55
3.3.1.1	Identificación de actores y tareas	55
3.3.1.2	Especificación de las tareas	56
3.3.1.3	Especificación de casos de uso.....	57
3.3.1.3.1	Caso de uso: general.....	58
3.3.1.3.2	Caso de Uso: Usuarios Administración	58
3.3.1.3.3	Caso de Uso: Usuarios Medico	61
3.3.1.3.4	Caso de Uso: Usuarios secretaria	63
3.3.1.4	Especificación de UIDs.....	64
3.3.2	Diseño conceptual	65
3.3.3	Diseño navegacional	67
3.3.3.1	Diseño de navegacional: Usuarios Administración	68
3.3.3.2	Diseño de navegacional: Usuarios Medico	68
3.3.3.3	Diseño de navegacional: Usuarios Secretaria	69
3.3.4	Diseño de interfaz abstracta	69
3.3.5	Implementación	71
CAPÍTULO IV		81
4	MÉTRICAS DE CALIDAD, COSTO Y SEGURIDAD	82
4.1	INTRODUCCIÓN.....	82

4.2	METRICAS DE CALIDAD.....	82
4.2.1	Funcionalidad	82
4.2.2	Confiabilidad.....	88
4.2.3	Mantenibilidad.....	89
4.2.4	Usabilidad	90
4.2.5	Portabilidad.....	91
4.3	ESTIMACIÓN DE COSTO DEL SOFTWARE.....	92
4.3.1	Método de estimación COCOMO II	92
4.3.2	Justificación de los valores	93
4.3.2.1	Atributos de software	93
4.3.2.2	Atributos de hardware.....	93
4.3.2.3	Atributos del personal	93
4.3.2.4	Atributos del proyecto	94
4.4	SEGURIDAD	96
4.4.1	Seguridad física	96
4.4.2	Seguridad lógica.....	97
4.5	PRUEBAS DE CAJA NEGRA Y BLANCA.....	97
4.5.1	Pruebas de Caja Negra	97
4.5.2	Prueba de Caja Blanca.....	99
4.5.3	Pruebas de estrés.....	102
CAPÍTULO V		106
5	CONCLUSIONES Y RECOMENDACIONES.....	106
5.1	CONCLUSIONES	106
5.2	RECOMENDACIONES.....	107
Bibliografía		108

Indicé de Tablas

Tabla 1	<i>Especificación de casos de uso</i>	26
Tabla 2	<i>Comparación de Arquitecturas</i>	35
Tabla 3	<i>Características de Calidad – Modelo ISO/IEC 9126</i>	39
Tabla 4	<i>Coeficiente de Modelo Cocomo II</i>	46
Tabla 5	<i>Ecuaciones del Modelo de Cocomo II</i>	47
Tabla 6	<i>Especificaciones de las tareas</i>	56
Tabla 7	<i>Descripción de los actores del sistema</i>	57
Tabla 8	<i>Caso de Uso: Usuarios Administración</i>	59
Tabla 9	<i>Caso de Uso: Usuarios Medico</i>	62
Tabla 10	<i>Caso de Uso: Usuarios secretaria</i>	63
Tabla 11	<i>Número de Entradas de Usuario</i>	83
Tabla 12	<i>Número de Salidas Usuario</i>	83
Tabla 13	<i>Número de Peticiones del Usuario</i>	84
Tabla 14	<i>Numero de Archivos</i>	85
Tabla 15	<i>Numero de Interfaces Externas</i>	85
Tabla 16	<i>Factores de Ponderación</i>	86
Tabla 17	<i>Valores de Ajuste de Complejidad</i>	86
Tabla 18	<i>Ajuste de Preguntas</i>	90
Tabla 19	<i>Coeficientes del Modelo COCOMO II</i>	92
Tabla 20	<i>Cálculo de Atributos FAE de valores de coste COCOMO II</i>	94
Tabla 21	<i>Valor de determina dos casos de prueba</i>	101
Tabla 22	<i>Resultados Prueba de estrés</i>	104

Indicé de Figuras

Figura 1	<i>Sistema</i>	16
Figura 2	<i>Fases Metodología OOHDM</i>	23
Figura 3	<i>Clasificación de roles</i>	25
Figura 4	<i>Definición de roles y tareas</i>	26
Figura 5	<i>Diseño Conceptual</i>	28
Figura 6	<i>Diseño Navegacional</i>	29
Figura 7	<i>Arquitectura en dos niveles</i>	33
Figura 8	<i>Arquitectura Tres Niveles</i>	34
Figura 9	<i>Esquema del Sistema</i>	54
Figura 10	<i>Identificación de actores</i>	55
Figura 11	<i>Caso de Uso: Usuarios Administración</i>	59
Figura 12	<i>Caso de Uso: Usuarios Medico</i>	61
Figura 13	<i>Caso de Uso: Usuarios secretaria</i>	63
Figura 14	<i>UID: Administración de usuarios</i>	64
Figura 15	<i>UID: Autenticación de usuario</i>	64
Figura 16	<i>Diagrama de clases</i>	66
Figura 17	<i>Modelo de Flujo de Proceso de Autenticar Usuarios al Sistema</i>	67
Figura 18	<i>Diseño de navegacional: Usuarios Administración</i>	68
Figura 19	<i>Diseño de navegacional: Usuarios Medico</i>	68
Figura 20	<i>Diseño de navegacional: Usuarios Secretaria</i>	69
Figura 21	<i>Diseño de interfaz abstracta login</i>	70
Figura 22	<i>Diseño de interfaz abstracta administración</i>	71
Figura 23	<i>Diseño de interfaz de Inicio de Sesión.</i>	73
Figura 24	<i>Diseño de interfaz de inicio de sistema</i>	74
Figura 25	<i>Diseño de interfaz de módulo de usuario</i>	74
Figura 26	<i>Diseño de interfaz de módulo de privilegios</i>	75
Figura 27	<i>Diseño de interfaz de módulo de sucursales</i>	75
Figura 28	<i>Diseño de interfaz de módulo de administración de personal de clínica</i> ..	76
Figura 29	<i>Diseño de interfaz de formulario de registro de paciente</i>	76
Figura 30	<i>Inicio de sesión al sistema</i>	98

Figura 31 <i>Interfaz de inicio del sistema</i>	99
Figura 32 <i>Creación de nuevo usuario flujo drama.</i>	100
Figura 33 <i>Formulario de creación de usuario</i>	100
Figura 34 <i>Listar usuario</i>	101
Figura 35 <i>Software Jmeter</i>	102
Figura 36 <i>Ejecución del programa</i>	103
Figura 37 <i>Resultados de las pruebas</i>	103

RESUMEN

Hoy en día la información debería ser considerada como uno de los más valiosos recursos de una organización y el sistema de información es el encargado de que esta sea gestionada siguiendo criterios de eficacia y eficiencia

En la era digital actual, la eficiencia y precisión son elementos clave para cualquier organización, y los centros odontológicos no son la excepción. La gestión adecuada de los historiales clínicos es fundamental para brindar un cuidado de calidad a los pacientes y garantizar un flujo de trabajo óptimo en el Centro Odontológico Zonadental.

El objetivo es Desarrollar un sistema de administración y seguimiento de historiales clínicos odontológicos multisucursal para el Centro Odontológico Zonadental, con el fin de mejorar la eficiencia y la calidad de la gestión de los registros de los pacientes y facilitar el acceso a la información.

La metodología aplicada en este proyecto es la metodología OOHDM. La metodología OOHDM (Método de Diseño Hipermedia Objeto Orientado) Propone el desarrollo de aplicaciones Web hipermedia a través de un proceso de 5 fases que se combinan con notación UML.

Con base en el análisis del caso del Centro Odontológico ZonaDental y su implementación exitosa de un Sistema de Administración y Seguimiento de Historiales Clínicos Odontológicos, podemos extraer varias conclusiones.

ABSTRACT

Nowadays, information should be considered one of the most valuable resources of an organization and the information system is responsible for ensuring that it is managed following criteria of effectiveness and efficiency.

In today's digital age, efficiency and precision are key elements for any organization, and dental centers are no exception. Proper management of clinical records is essential to provide quality care to patients and ensure optimal workflow at the Zonadental Dental Center.

The objective is to develop a multi-branch dental clinical history administration and monitoring system for the Zonadental Dental Center, in order to improve the efficiency and quality of patient records management and facilitate access to information.

The methodology applied in this project is the OOHDM methodology. The OOHDM methodology (Object Oriented Hypermedia Design Method) proposes the development of hypermedia Web applications through a 5-phase process that is combined with UML notation.

Based on the analysis of the case of the ZonaDental Dental Center and its successful implementation of a System for Administration and Monitoring of Dental Clinical Records, we can draw several conclusions.

LISTADO DE ABREVIATURAS

Software: estos son los programas informáticos que hacen posible la ejecución de tareas específicas dentro de un computador.

Sistema: Conjunto ordenado de normas y procedimientos que regulan el funcionamiento de un grupo o colectividad.

Información: En la Informática, como información se denomina el conjunto de datos organizados y procesados que funcionan como mensajes, instrucciones y operaciones o cualquier otro tipo de actividad que tenga lugar en una computadora.

ISO/IEC 9126: La Norma ISO/IEC 9126 (1991): Evaluación del producto software. Características de calidad y guías para su uso, que fue desarrollada para dar apoyo a estas necesidades, define seis características de calidad y describe un modelo de proceso de evaluación del producto software.

Caja blanca: En programación, se denomina cajas blancas a un tipo de pruebas de software que se realiza sobre las funciones internas de un módulo

Caja negra: Caja negra es un término técnico que se aplica a la forma en que se estudia un sistema principalmente en términos de sus características de entrada y salida.

Seguridad Lógica: La seguridad lógica informática es una referencia a la protección por el uso de software en una organización, e incluye identificación de usuarios y contraseñas de acceso, autenticación, derechos de acceso y niveles de autoridad.

Seguridad Física: La seguridad física nos referimos a todos aquellos mecanismos -- generalmente de prevención y detección-- destinados a proteger físicamente cualquier recurso del sistema.

Framework: Un framework es un esquema o marco de trabajo que ofrece una estructura base para elaborar un proyecto con objetivos específicos, una especie de plantilla que sirve como punto de partida para la organización y desarrollo de software.

CAPÍTULO I
MARCO
PRELIMINAR

CAPÍTULO I

1 MARCO PRELIMINAR

1.1 INTRODUCCIÓN

Hoy en día la información debería ser considerada como uno de los más valiosos recursos de una organización y el sistema de información es el encargado de que esta sea gestionada siguiendo criterios de eficacia y eficiencia (Gómez y Suarez, 2007)

Si tuviéramos que resumir con una sola frase el principal cometido de un sistema de información dentro una organización podríamos afirmar que este se encarga de entregar la información oportuna precisa, con la presentación y el formato adecuado, a la persona que la necesita dentro la organización para tomar una decisión o realizar alguna operación y justo en el momento en que esta persona necesita disponer dicha información (Gómez y Suarez, 2007)

En la actualidad, para muchas organizaciones los sistemas de información basados en computadoras son el corazón de las actividades cotidianas y objeto de gran consideración en la toma de decisiones. La información juega un papel muy importante en las instituciones, es base para la toma de decisiones, debe estar disponible en cualquier momento de tal forma que sea la más oportuna, confiable y efectiva.

En la era digital actual, la eficiencia y precisión son elementos clave para cualquier organización, y los centros odontológicos no son la excepción. La gestión adecuada de los historiales clínicos es fundamental para brindar un cuidado de calidad a los pacientes y garantizar un flujo de trabajo óptimo en el Centro Odontológico Zonadental.

El nuevo sistema se basará en un software especializado diseñado específicamente para la gestión de historiales clínicos odontológicos. Esta solución tecnológica proporcionará una plataforma centralizada para almacenar, administrar y acceder a los historiales clínicos de los pacientes, eliminando la necesidad de archivos físicos y permitiendo un fácil intercambio de información entre los miembros del equipo dental.

Con el desarrollo de un sistema, se pretende sistematizar la información con un sistema de administración y seguimiento de historiales clínicos odontológicos que se

realizará para el consultorio clínico Centro Odontológico Zonadental que no existiera mal manejo de información ni gastos insulsos en materiales de escritorio ni pérdida tiempo y que se puedan agilizar la ejecución de muchos más proyectos que beneficien al consultorio.

Las herramientas con las que se trabajaran para la construcción del sistema son los siguientes; como lenguaje de programación utilizaremos PHP, como gestor base de datos MariaDB utilizaremos HTML5, CSS y JAVASCRIPT para el diseño del sistema y como metodología de ingeniería se trabajara con OOADM (Método de Diseño Hipermedia Objeto Orientado) el cual se adecua perfectamente al proyecto por su desarrollo, esto porque su principal objetivo es la productividad y dar prioridad a los trabajos que den un resultado directo.

1.2 ANTECEDENTES

1.2.1 Antecedentes de la institución

La clínica de Centro Odontológico Zonadental cuenta con dos sucursales actualmente ubicada, la clínica central en la Calle: Martín Lutero Nro. 100 zona Villa Mejillones, esquina avenida Juan Pablo II frente al Hospital de Norte, con número de Nit: 7643645011, con actividad principal: “86200 - Actividades de médicos y odontológicos”. La segunda sucursal ubicada en Av. Venancio Condori Nro. 502, Ex parada 8, distrito 14 El Alto, con una trascendencia de trabajo prestando el servicio a la población con más de seis años de servicios.

Actualmente la implementación de sistema de administración y seguimiento de historiales clínicos odontológicos se ha incrementado debido a la necesidad de tener un mejor control de la información de los pacientes, para así brindar una atención de calidad de manera eficiente. Así mismo podemos mencionar a algún software que actualmente se usa en diferentes clínicas, etc.

1.2.2 Proyectos similares

1.1.1.1 Proyectos similares internacionales

- “SOFTWARE PARA LA GESTIÓN DE CONTROL DE HISTORIAS CLINICAS ODONTOLOGICAS -VENEZUELA”.- La presente investigación tiene como objetivo desarrollar un sistema de gestión de control de historias clínicas, donde se pueda tener un control de cada consulta y evolución del paciente, duración de tratamientos.(Duque,2009).
- “ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE INFORMACIÓN INTEGRAL DE GESTIÓN HOSPITALARIA PARA UN ESTABLECIMIENTO DE SALUD PÚBLICO CASO: HOSPITAL NACIONAL CAYETANO HEREDIA DEL PERÚ”.- Presenta una propuesta que abarca la administración de la oficina de logística y la oficina de economía pertenecientes a la oficina ejecutiva de apoyo administrativo, admisión, archivo clínico, carga de servicios, citas y la oficina de estadística.
- “HISTORIA CLINICA ELECTRONICA EN UN DEPARTAMENTO DE OBSTETRICIA, GINECOLOGIA Y REPRODUCCION: DESARROLLO E IMPLEMENTACIÓN-BARCELONA”.- Presenta una propuesta para el centro sanitario, realiza el proceso de atención médica a través de asignación de consultas y el diagnóstico médico para la unidad de ginecología, obstetricia y reproducción.(Sabartés, 2013).

1.2.2.1 Proyectos similares locales

- “SISTEMA DE GESTION HOSPITALARIO MODULO CONSULTA EXTERNA CASO: SEGURO SOCIAL UNIVERSITARIO”. - Presenta una propuesta de Implementar un sistema de Consulta externa para el Seguro Social Universitario de La Paz, para brindar al médico la sistematización en la redacción de diagnósticos y tratamientos a los pacientes. (Avalos Nina, 2013).
- “SISTEMA DE GESTIÓN DE INFORMACIÓN CLÍNICA” CASO: CARRERA DE ENFERMERÍA”. - Presenta la propuesta de Desarrollar e implementar un Sistema de Gestión de Información Clínica, que permita gestionar las tareas

que se llevan a cabo en el consultorio de la Carrera de Enfermería. (Lara Ticona, 2018)

- “SISTEMA DE GESTION Y SEGUIMIENTO DE PACIENTES CASO: CENTRO DE SALUD JESUS OBRERO”.- El presente proyecto tiene el objetivo de Desarrollar un Sistema de Información Informático para gestión y seguimiento de pacientes para el Centro de Salud Jesús Obrero, que minimice el tiempo de respuesta de los procesos de modo que mejore el registro, atención y seguimiento de pacientes, además permita contar con información oportuna y confiable a las instancias superiores apoyando la toma de decisiones.(Chambi Alcon, 2014).

1.3 PLANTEAMIENTO DEL PROBLEMA

1.3.1 Problema general

El problema general del sistema de administración y seguimiento de historiales clínicos odontológicos en el caso del Centro Odontológico Zonadental, es la ineficiencia en la gestión y acceso a los historiales clínicos de los pacientes. Este problema se puede desglosar en los siguientes aspectos:

- Almacenamiento físico: El Centro Odontológico Zonadental enfrenta dificultades debido al uso de registros en papel para almacenar los historiales clínicos. Esto puede ocupar un espacio significativo en el centro y dificultar la organización y búsqueda de los expedientes de los pacientes. Además, los archivos físicos pueden estar expuestos a riesgos como pérdida, daño o deterioro.
- Acceso y compartición limitada: La falta de un sistema de administración digital dificulta el acceso rápido y eficiente a los historiales clínicos. El personal tiene las dificultades de localizar y compartir información relevante de manera oportuna, lo que afecta la eficiencia del flujo de trabajo y la calidad de la atención.
- Programación y seguimiento de citas: La gestión manual de las citas se generan confusiones y dificultades para el personal y los pacientes. La falta de un sistema automatizado de programación y recordatorio de citas generan un

retraso, de cancelaciones o pérdida de consultas, afectando la eficiencia y puntualidad del centro.

Estos problemas afectan la eficiencia operativa y la calidad de la atención en el Centro Odontológico Zonadental, por lo cual es necesario tener un sistema que sea capaz de ayudar en la administración y seguimiento de los historiales clínicos de los pacientes de acuerdo a las necesidades que tiene el centro odontológico en sus sucursales.

1.3.2 Problemas específicos

- Demora en el registro de datos del paciente por la elaboración de varios documentos como la transcripción en libros y formularios al mismo tiempo.
- Retardo en la búsqueda de historiales clínicos del paciente se entremezclan por la gran cantidad de archivos.
- Falta de control en la organización de los códigos de los pacientes extravió y duplicado del historial clínico.
- Falta de información acerca de pacientes no existe fácil acceso a los datos de un paciente.
- Los pagos de los tratamientos se realizan en recibos, implicando la posible pérdida de los mismos y el descontrol de los ingresos.
- Almacenamiento manual de historiales clínico, utilizando de varios gaveteros donde se guardan los sobres manillas con los formularios.

1.4 FORMULACIÓN DEL PROBLEMA

El problema específico del sistema de administración y seguimiento de historiales clínicos odontológicos en el caso del Centro Odontológico Zonadental puede ser formulado de la siguiente manera:

¿Cómo mejorar la gestión y accesibilidad de los historiales clínicos odontológicos en el Centro Odontológico Zonadental para optimizar la eficiencia operativa, garantizar la integridad de los datos y mejorar la calidad de la atención al paciente?

1.5 OBJETIVOS

1.5.1 Objetivo general

Desarrollar un sistema de administración y seguimiento de historiales clínicos odontológicos multisucursal para el Centro Odontológico Zonadental, con el fin de mejorar la eficiencia y la calidad de la gestión de los registros de los pacientes y facilitar el acceso a la información.

1.5.2 Objetivos específicos

- Analizar la situación actual de los procesos en la administración de historias clínicas odontológicas del centro médico.
- Diseñar una base de datos, acorde a los requerimientos realizados.
- Elaborar un sistema de fácil uso y acceso a la información con una interfaz que brinde una comodidad a los usuarios.
- Definir autenticación de usuarios donde se asignará roles y niveles de acceso para los usuarios, para resguardar la información.
- Realizar pruebas al sistema para su buen funcionamiento.
- Evaluar el sistema en términos de las métricas, costos y seguridad de la información.

1.6 JUSTIFICACIÓN

1.6.1 Justificación técnica

El sistema de administración y seguimiento de historiales clínicos odontológicos, se justifica porque las sucursales del centro cuentan con los recursos informáticos, suficientes como ser: Una laptop Core I5, una computadora de escritorio Core I3 y acceso a internet en ambas sucursales.

El gerente de las clínicas esta con la predisposición de alquilar un servidor web que tendrá un dominio con el cual podrán acceder desde cualquier navegador.

El sistema a desarrollar será multiplataforma puede funcionar tanto en GNU/Linux o Windows; escalable con capacidad de crecer en el tiempo.

1.6.2 Justificación económica

El desarrollo del Sistema reducirá gastos en material (formularios para historiales, sobres manillas, libros de consultas e historiales de pacientes), con lo que se ahorrará en costos. El acceso a la información y su calidad en tiempo reducido, mejorando de esta forma la atención médica.

1.6.3 Justificación social

El proyecto se justifica desde el punto de vista social, porque el desarrollo del presente proyecto de grado, beneficiará a la clínica odontológica con un sistema automatizado, que administre y Controle los Historiales Clínicos de los pacientes que acuden para su atención a los Consultorios de distintos sucursales, mejorando las condiciones de trabajo, evitando de esta manera las tareas manuales para el personal y al mismo tiempo brindando una información veraz, rápida y oportuna a los médicos y pacientes.

1.7 METODOLOGÍA

1.7.1 Método de ingeniería

1.7.1.1 Metodología de desarrollo

Metodología OOHDM: La metodología OOHDM (Método de Diseño Hipermedia Objeto Orientado) Propone el desarrollo de aplicaciones Web hipermedia atreves de un proceso de 5 fases que se combinan con notación UML. (Franco, 2012)

El modelo OOHDM propone 5 etapas:

- Obtención de requerimientos.
- Diseño conceptual.
- Diseño navegacional.
- Diseño de interfaz abstracta.
- Implementación.

1.7.1.2 Métrica de calidad al software

ISO/IEC 9126: El estándar ISO/IEC 9126: 2001 presenta un marco conceptual para el modelo de calidad y define un conjunto de características y sub características, las que debe cumplir todo producto software. (Verity, 2021)

El estándar ISO-9126 establece que cualquier componente de la calidad del software puede ser descrito en términos de una o más de las siete características básicas: funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad, portabilidad y satisfacción; cada una de ellas se detalla a través de un conjunto de sub características que permiten profundizar en la evaluación de la calidad de productos de software. (Verity, 2021)

1.7.1.3 Estimación de costo

COCOMO II: Este modelo permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de costo y de escala. Los factores de costo describen aspectos relacionados con la naturaleza del producto, hardware utilizado, personal involucrado, y características propias del proyecto. El conjunto de factores de escala explica las economías y des economías de escala producidas a medida que un proyecto de software incrementa su tamaño. (GOMEZ, 2010)

1.7.1.4 Seguridad

ISO 27001: Los requisitos de la Norma ISO 27001, norma que nos aportan un Sistema de Gestión de la Seguridad de la Información (SGSI), consistente en medidas orientadas a proteger la información, indistintamente del formato de la misma, contra cualquier amenaza, de forma que garanticemos en todo momento la continuidad de las actividades de la empresa. (Normas ISO, 2018).

Los Objetivos del SGSI son preservar la:

- Confidencialidad
- Integridad
- Disponibilidad de la Información

¿Por qué ISO 27001? Porque para el fin de preservar la información, se ha demostrado que no es suficiente la implantación de controles y procedimientos de seguridad realizados frecuentemente sin un criterio común establecido, en torno a la compra de productos técnicos y sin considerar toda la información esencial que se debe proteger. (Normas ISO, 2018)

1.7.1.5 Pruebas al software

- **Caja negra:** Las Pruebas de Caja Negra, es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software. (Terrera, 2017)
- En las pruebas de caja negra el Tester solo se centra en las entradas y salidas de la aplicación, sin preocuparse por el contenido interno. Lo que pase por dentro es indiferente, solo importa que, si se realiza cierta acción, la salida sea la indicada según los requerimientos. (Paulenko, 2019)
- **Caja blanca:** Las pruebas de caja blanca (también conocidas como pruebas de caja de cristal o pruebas estructurales) se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente, lo cual, esto significa que tenemos que realizar un “estudio”, o más bien, un análisis de nuestro código. (Kimbo Cesar, 2021)
- **Prueba de estrés (stress):** Consiste en probar los límites que un sistema puede soportar. En este tipo de pruebas se suele enviar más peticiones de las que el software podría atender normalmente para saber el comportamiento de la aplicación. (Daniel, 2018).

1.8 HERRAMIENTAS

Para el desarrollo del sistema web utilizaremos las siguientes herramientas que nos ayudara en el desarrollo del sistema web.

1.8.1 Gestor de base de datos MariaDB

Es un sistema gestor de bases de datos (SGBD), es decir, un conjunto de programas que permiten modificar, almacenar, y extraer información de una base de

datos. Disponiendo de otro tipo de funcionalidades como la administración de usuarios, y recuperación de la información si el sistema se corrompe, entre otras. (Michael Wedenius, 2009)

1.8.2 Framework codeIgniter

Es un entorno de desarrollo web escrito en PHP que presume de acelerar y optimizar el desarrollo de aplicaciones web gracias a un compacto diseño de software. El diseño orientado al rendimiento de este framework de desarrollo web se revela en su parca arquitectura, pues se basa en el patrón Modelo-Vista-Controlador (MVC). El principio fundamental que sustenta a la arquitectura de desarrollo MVC es la estricta separación entre el código y la presentación, gracias a una estructura modular de software y a la externalización del código PHP. (1y1 IONOS España S.L.U., 2020)

1.8.3 HTML (lenguaje de marcas de hipertexto)

Es un lenguaje de marcado para la elaboración de páginas web. Define una estructura básica y un código para la definición de contenido de una página web, como texto, imágenes, videos, entre otros (Tim Berners-Lee,2009)

1.8.4 Framework bootstrap

Bootstrap es un framework CSS y JavaScript diseñado para la creación de interfaces limpias y con un diseño responsivo. Además, ofrece un amplio abanico de herramientas y funciones, de manera que los usuarios pueden crear prácticamente cualquier tipo de sitio web haciendo uso de los mismos. Actualmente, Bootstrap es una de las alternativas más populares a la hora de desarrollar tanto sitios webs como aplicaciones. Una de las principales ventajas que ofrece es que permite la creación de sitios y apps 100% adaptables a cualquier tipo de dispositivo. (Jacob Thornton ,2019)

1.8.5 MagicDraw 16.0

MagicDraw, herramienta CASE desarrollada por No Magic. Es compatible con el estándar UML 2.3, desarrollo de código para diversos lenguajes de programación (Java, C++ y C#, entre otros) así como para modelar datos. Cuenta con capacidad para trabajar en equipo y es compatible con varios entornos de desarrollo (IDEs)

Diseñada para los analistas del negocio, los analistas del software, los programadores, los ingenieros y los escritores de la documentación.

Facilita el análisis y el diseño de los sistemas Orientado a Objetos (OO) y de las bases de datos orientados objeto (EcuRed, 2007).

Disponible en <http://www.pst.ifi.lmu.de/>, permite el diseño UML de las aplicaciones Web, de manera estandarizada.

1.8.6 JQuery

JQuery es un Framework JavaScript, nos ofrece una infraestructura con la que tendremos mucha mayor facilidad para la creación de aplicaciones complejas del lado del cliente. Por ejemplo, con JQuery obtendremos ayuda en la creación de interfaces de usuario, efectos dinámicos, aplicaciones que hacen uso de Ajax, etc. Cuando programemos JavaScript con JQuery tendremos a nuestra disposición una interfaz para programación que nos permitirá hacer cosas con el navegador que estemos seguros que funcionarán para todos nuestros visitantes. Simplemente debemos conocer las librerías del Framework y programar utilizando las clases y métodos para la consecución de nuestros objetivos (Miguel Angel, 2012).

1.8.7 PHP

PHP, es un lenguaje de programación interpretado de alto nivel al lado del servidor para internet, muy similar en su sintaxis al lenguaje C, con algunas diferencias, no compila como al igual que C, ya que es un intérprete, por lo tanto, cada vez que se debe ejecutar un programa, lo interpreta verificando toda su sintaxis.

PHP nos brinda la posibilidad de realizar tareas de forma automatizadas, mejorando la productividad de nuestro sitio web y dando la posibilidad de añadir gran cantidad de funcionalidades que con HTML no podemos hacerlo, ya que HTML no es un lenguaje de programación.

PHP5 mejorara los mecanismos de POO para solucionar las carencias de las anteriores versiones. Un paso necesario para Conseguir que PHP sea un lenguaje apto para todo tipo de aplicaciones y entornos, incluso los más exigentes (Alvarez, López, y Gutierrez, 2023).

1.8.8 Servidor apache

Apache es el Servidor Web hecho por excelencia por su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. La licencia es una descendiente de la licencias BSD(Berkeley Software Distribution), no es GPL(General Public License). Esta licencia permite hacer lo que quieras con el código fuente siempre que les reconozcas su trabajo.

Apache es una tecnología gratuita de código fuente abierta. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto.

Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor las características de Apache se pueden extender hasta donde nuestra imaginación y conocimientos lleguen (Cooper, 2004).

1.9 LÍMITES Y ALCANCES

1.9.1 Limites

Los límites del sistema de administración y seguimiento de historiales clínicos odontológicos para el consultorio clínico Centro Odontológico Zonadental:

- Las interfaces del sistema estarán restringidas a los tipos de usuarios autorizados del sistema.
- El proyecto realizara el análisis, diseño y desarrollo de los procesos que integran en función de los requerimientos y el estudio de la situación actual.

1.9.2 Alcances

El presente proyecto de grado contemplará los siguientes alcances: desarrollar un sistema de administración y seguimiento de historiales clínicos odontológicos multisucursal, en el ambiente geográfico en una red intranet, el cual comprenderá los siguientes módulos:

- Módulo administración de usuarios: En este módulo se realiza la validación de usuarios, de quienes podrán tener accesos a la administración del sistema web dependiendo al rol que tengan.
- Módulo registro de pacientes: En este módulo se registrará a los clientes

pacientes.

- Módulo de citas médicas: En este módulo se reservará la hora y día de la cita de con los pacientes eso para no generar choques de horarios.
- Módulo de historial clínico: Donde se verá todas las intervenciones que se le hizo a los pacientes y el estado de sus tratamientos.
- Módulo de odontograma: Donde representara gráficamente todas las intervenciones y tratamientos realizados antes y después de la visita.
- Diseñar el módulo de caja para el registro del costo de cada tratamiento que se realiza al paciente
- Módulo de reportes: En este módulo se realizan los reportes de todos los historiales médicos de los pacientes, costos de tratamientos, cantidad de pacientes atendidos, para tener un ajuste de costos.

1.10 APORTES

El aporte principal del proyecto de Grado es la aplicación de una intranet denominada sistema de administración y seguimiento de historiales clínicos odontológicos de multisucursales para el consultorio clínico Centro Odontológico Zonadental, utilizara este medio para centralizar la información que ingresa, debido a la demanda de la población, el cual va a difundir información confiable y oportuna de los pacientes, como también en la realización de informes estadísticos, cubriendo de esta forma las falencias que existen actualmente como es el empleo de varios formularios que deben llenarse manualmente.

CAPÍTULO II

MARCO TEÓRICO

CAPÍTULO II

2 MARCO TEÓRICO

2.1 INTRODUCCIÓN

En el presente capítulo se dará a conocer las definiciones y conceptos teóricos que son necesarios para el desarrollo e implementación del sistema web, todos estos conceptos se basan en las herramientas, metodología y métodos que serán aplicados para el desarrollo del sistema web, los conceptos teóricos constituyen en la base que sustentará el desarrollo e implementación del sistema web.

2.2 DATO

Un dato es la representación de una variable que puede ser cuantitativa o cualitativa que indica un valor que se le asigna a las cosas y se representa a través de una secuencia de símbolos, números o letras. (Suarez, 2020)

Un dato se puede definir como una representación de hechos, conceptos o instrucciones de una manera formal, que debe ser adecuada para la comunicación, interpretación o procesamiento por parte de una máquina humana o electrónica. Los datos son simplemente hechos o cifras, bits de información, pero no información en sí. Los datos pueden ser cualquier carácter, texto, palabra, número y, si no se ponen en contexto, significan poco o nada para un humano. (Suarez, 2020)

Un dato es el elemento principal de la información representado por letras, números y símbolos que reunidos pueden conformar un mensaje o información concreta.

2.3 INFORMACIÓN

La información es un conjunto de datos acerca de algún suceso, hecho, fenómeno o situación, que organizados en un contexto determinado tienen su significado, cuyo propósito puede ser el de reducir la incertidumbre o incrementar el conocimiento acerca de algo. (Thompson, 2023)

La información es el conjunto de datos que configuran un mensaje que emite un emisor y que se pretende llegue al receptor para que quede informado. Siempre se lleva a

cabo a través de un canal que es el que uno a ambos interlocutores ha elegido. (Peiró, 2020)

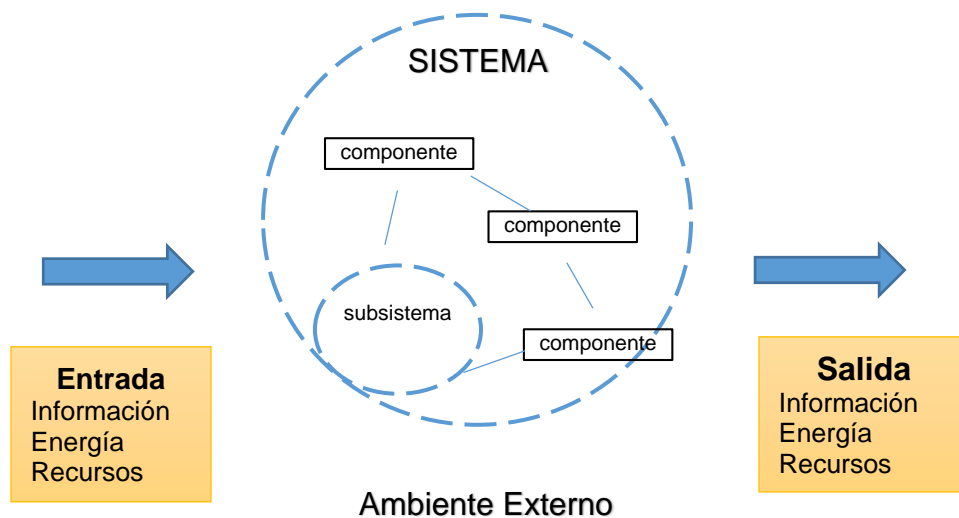
La información es un conjunto de datos con un significado, o sea, que reduce la incertidumbre o que aumenta el conocimiento de algo. En verdad, la información es un mensaje con significado en un determinado contexto, disponible para uso inmediato y que proporciona orientación a las acciones por el hecho de reducir el margen de incertidumbre con respecto a nuestras decisiones. (Chiavenato, 2006)

2.4 SISTEMA

Un sistema es un conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo. Los sistemas reciben datos, energía o materia del ambiente (entrada) y proveen información, energía o materia (salida). (Alegsa, 2023)

Figura 1

Sistema



NOTA: Información obtenida de (Alegsa, 2023)

Sistema es un objeto en el que variables de distintos tipos interactúan y producen señales observables. Las señales observables que nos son de interés se suelen denominar salidas. El sistema está afectado también por estímulos externos. Las señales externas que pueden ser manipuladas por el observador se denominan

entradas; las que no se pueden manipular se denominan perturbaciones y se dividen en aquellas que son directamente medibles y aquellas que son solo observables por su influencia sobre la salida. (Ljung, 2014)

2.4.1 Tipos de sistemas

2.4.1.1 Sistemas conceptuales o abstractos

Se trata de conjuntos ordenados e interrelacionados de conceptos e ideas, que pueden ser de cuatro tipos diferentes: individuos, predicados, conjuntos u operadores. Son de tipo abstracto, intangible. (Etecé, 2021)

2.4.1.2 Sistemas reales o materiales

Por el contrario, son tangibles, concretos, y están compuestos de componentes físicos, es decir, cosas con propiedades específicas, como energía, historia, posición, etc. (Etecé, 2021)

2.5 SISTEMA WEB

Los «sistemas Web» o también conocido como «aplicaciones Web» son aquellos que están creados e instalados no sobre una plataforma o sistemas operativos (Windows, Linux). Sino que se alojan en un servidor en Internet o sobre una intranet (red local). Su aspecto es muy similar a páginas Web que vemos normalmente, pero en realidad los ‘sistemas Web’ tienen funcionalidades muy potentes que brindan respuestas a casos particulares.

Los sistemas Web se pueden utilizar en cualquier navegador Web (chrome, firefox, Internet Explorer,etc) 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.

Las aplicaciones Web trabajan con bases de datos que permiten procesar y mostrar información de forma dinámica para el usuario.

Los sistemas desarrollados en plataformas Web, tienen marcadas diferencias con otros tipos de sistemas, lo que lo hacen muy beneficioso tanto para las empresas que lo utilizan, como para los usuarios que operan en el sistema.

Este tipo de diferencias se ven reflejada en los costos, en la rapidez de obtención de la información, en la optimización de las tareas por parte de los usuarios y en alcanzar una gestión estable. (DESARROLLO WEB, 2015)

Las aplicaciones Web trabajan con bases de datos que permite procesar y mostrar información de forma dinámica para el usuario. (Baez, 2012)

Los sistemas desarrollados en plataformas Web, tienen marcadas diferencias con otros tipos de sistemas, lo que lo hacen muy beneficioso tanto para las empresas que lo utilizan, como para los usuarios que operan en el sistema. (Baez, 2012)

2.6 SISTEMA DE CONTROL

Es un sistema compuesto por un grupo de elementos que busca ejercer control sobre otros sistemas. Tiene como objetivo, completar de manera efectiva las tareas y asignaciones para las cuales fue programado. Para ello, deben comportarse de manera estable ante los errores. Tienen como función principal, organizar y administrar de manera efectiva toda la información computarizada con la que funcionan las máquinas y los equipos automatizados.

Fue en la antigua Grecia donde se desarrolló el primer sistema de control. Para ese entonces, se realizaron unidades que sistematizaban los pasos necesarios para que una plataforma flotara sobre el agua. Así se dio a conocer en la historia el primer sistema de control automatizado.

Posteriormente se crea el primer regulador de presión inventado por Denis Papin. A partir de ese momento, se comienzan a desarrollar un conjunto de fenómenos relacionados con los sistemas de control, los cuales se han ido perfeccionando con el pasar de los años y sumando características contundentes; gracias a los nuevos avances tecnológicos. (García, 2022)

2.7 CONTROL

Control es el proceso de verificar el desempeño de distintas áreas o funciones de una organización. Usualmente implica una comparación entre un rendimiento esperado y un rendimiento observado, para verificar si se están cumpliendo los objetivos de forma eficiente y eficaz. El control permite tomar acciones correctivas cuando sea necesario. (Anzil, 2023)

El control es una de las principales actividades administrativas de las organizaciones. El control se relaciona con la planeación, porque el control busca que el desempeño se ajuste a los planes. El proceso administrativo, desde el punto de vista tradicional, es un proceso circular que se retroalimenta. (zonaeconomica.com, 2019)

2.8 HISTORIAL CLÍNICO

Se puede definir el historial clínico como un documento legal de la rama médica que surge del contacto entre un paciente y un profesional de la salud, donde se recoge toda la información relevante acerca de la salud del paciente, de modo que se le pueda ofrecer una atención correcta y personalizada.

En el grupo de los profesionales de la salud que tienen acceso al historial médico de un paciente se encuentran: médicos, enfermeros, fisioterapeutas, odontólogos, psicólogos, podólogos, cirujanos, oftalmólogos, y profesionales de cualquier otra especialidad médica. (Marqués, 2023)

2.9 HISTORIA CLÍNICA ODONTOLÓGICA

La historia clínica odontológica es un documento que sirve para recolectar de forma ordenada los datos de identidad, síntomas, signos y otros elementos que permitan plantear un diagnóstico clínico odontológico.

Este documento se actualiza constantemente en el tiempo y además se respalda con anexos de resultados de laboratorio, radiografías, fotos, videos y notas de todo tipo.

Es decir, la historia clínica odontológica es todo el conjunto de documentos, evoluciones y procesos realizados a un paciente dental. (Dental, 2020)

2.10 ODONTOGRAMA

El Odontograma es un esquema de las arcas dentarias, diseñadas gráfica, anatómica y geométricamente. En él se encuentran las características anatómicas de los dientes, así como de las particularidades y modificaciones, con fines de identificación. Permiten asentar los datos del examen del paciente, tratamientos realizados y pendientes por realizar, además, forman una parte imprescindible de la historia clínica odontología del paciente.

El Odontograma es entonces un historial clínico que sirve para registrar la información dental de un paciente adulto o infantil, así como la identificación propia del paciente. (Gómez, 2017)

2.11 INGENIERÍA DE SOFTWARE

La ingeniería del software, según la definición de la IEEE en 1993, es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento del software. La ingeniería del software ofrece métodos o técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo, y trata áreas muy diversas de la informática y de las ciencias computacionales. (IEEE)

La ingeniería de software es una disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de programas informáticos, más conocidos como softwares.

Para nosotros, este servicio trasciende a la programación, que es la base para crear una aplicación. La ingeniería de software engloba toda la gestión de un proyecto. Desde el análisis previo de la situación, el planteamiento del diseño hasta su implementación, pasando por las pruebas recurrentes para su correcto funcionamiento. Podríamos decir que la ingeniería del software es el continente donde se aloja el contenido, que sería el software en sí. (Maida y Pacienza, 2015)

2.11.1 Proceso software

El proceso de ingeniería de software se define como «un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad». El proceso de desarrollo de

software «es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo». Concretamente «define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo» (Jacobson, 1998).

Hay cuatro actividades fundamentales comunes a todo proceso software:

- **Especificación:** usuarios e ingenieros definen el software a producir y las restricciones en su funcionalidad.
- **Desarrollo:** fase en la cual el software se diseña y se programa.
- **Validación:** el software debe ser probado para asegurar que cumple con las necesidades del cliente.
- **Evolución:** el software debe poder ser modificado para adaptarse a cambios en el mercado y en las necesidades de los usuarios.

2.12 INGENIERIA WEB

EL Web Engineering es una actividad que se enfoca en el uso de la ingeniería de software de nueva generación para obtener software eficiente y de alta calidad en aplicaciones y sistemas para la World Wide Web. Es un conjunto de conocimientos científicos, diseño gráfico, control de calidad de los procesos y fundamentos de ingeniería que llevan a un proceso de desarrollo de software de sitios web de alta calidad. (edX, 2020)

Existen diversas etapas que llevan al ciclo de vida de un proceso de Ingeniería web: la Formulación, la Planificación, el Análisis, la Ingeniería propiamente dicha, la Generación de páginas o webapps, las Pruebas y las Evaluaciones del cliente. (edX, 2020)

La ingeniería web se debe al crecimiento desenfrenado que está teniendo la Web está ocasionando un impacto en la sociedad y el nuevo manejo que se le está dando a la información en las diferentes áreas en que se presenta ha hecho que las personas tiendan a realizar todas sus actividades por esta vía. (Lopez y Perfil, 2015)

Categorías

Los sitios web pueden ser categorizados de la siguiente forma:

- Sólo estático que se enfoca en la organización de la estructura y el contenido, en la forma como se va a presentar la información y que sea fácil de manejar para cualquier usuario, pero debe tener en cuenta la eficiencia y la confiabilidad. (Lopez y Perfil, 2015)
- Sitio estático con formularios de entrada este sitio tiene las mismas características que el anterior, adicionándole que él les permite a los usuarios la interacción por medio de cuestionarios, comentario y sugerencias. (Lopez y Perfil, 2015)
- Sitio con acceso de datos dinámicos aquí, además de las características antes mencionadas, cuenta con bases de datos en las cuales el usuario puede realizar consultas y búsquedas. (Lopez y Perfil, 2015)
- Sitio creado dinámicamente en este sitio los requerimientos son parecidos, pero deben suplir con las necesidades de cada usuario; creando sitios dinámicos que sean compatibles con el entorno de navegación de cada usuario. (Lopez y Perfil, 2015)
- Aplicación de software basada en la Web este sitio puede tener todas las características antes mencionadas, pero logrando un parecido con una implementación cliente/servidor comúnmente conocido que a un sitio web estático. (Lopez y Perfil, 2015)

2.12.1 Usabilidad

Consiste de un conjunto de atributos que permiten evaluar el esfuerzo necesario que deberá invertir el usuario para utilizar el sistema. (Abud, 2012)

- **Comprensibilidad.** Se refiere al esfuerzo requerido por los usuarios para reconocer la estructura lógica del sistema y los conceptos relativos a la aplicación del software.
- **Facilidad de aprender.** Establece atributos del software relativos al esfuerzo que los usuarios deben hacer para aprender a usar la aplicación.

- **Operabilidad.** Agrupa los conceptos que evalúan la operación y el control del sistema.

2.13 METODOLOGÍA OOHDM

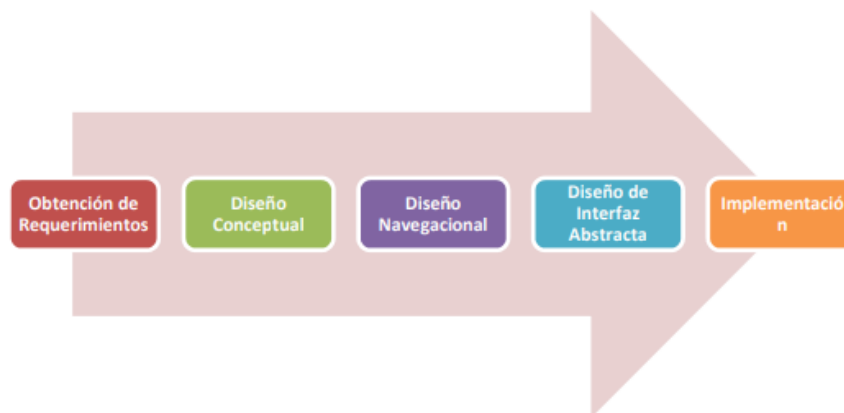
OOHDM es una metodología de desarrollo propuesta por Rossi y Schwabe (ROSSI 1996) para la elaboración de aplicaciones multimedia y tiene como objetivo simplificar y a la vez hacer más eficaz el diseño de aplicaciones hipermedia. OOHDM está basada en HDM, en el sentido de que toma muchas de las definiciones, sobre todo en los aspectos de navegación, planteadas en el modelo de HDM. Sin embargo, OOHDM supera con creces a su antecesor, ya que no es simplemente un lenguaje de modelado, sino que define unas pautas de trabajo, centrado principalmente en el diseño, para desarrollar aplicaciones multimedia de forma metodológica. (Ricardo, 2014)

Esta metodología plantea el diseño de una aplicación de este tipo a través de cinco fases que se desarrollan de un modo iterativo. Estas fases son:

Figura 2.1.

Figura 2

Fases Metodología OOHDM



NOTA: Información obtenida de (Ricardo, 2014)

Esta metodología está compuesta por cinco fases:

- Obtención de Requerimientos
- Diseño conceptual
- Diseño navegacional
- Diseño de interfaces abstractas
- Implementación

a) Fase I. Obtención de requerimientos

El primer paso es definir los requerimientos de los usuarios directamente involucrados y las tareas que ellos realizan, para poder capturarlos se tienen que utilizar los Casos de Uso.

Es así que OOHDM divide esta fase en cuatro sub-fases.

- Identificación de roles (actores) y tareas.
- Especificación de casos de uso.
- Especificación de UIDs.
- Validación de casos de uso y UIDs.

1. Identificación de roles (actores) y tareas

Esta sub fase tiene por objetivo identificar las necesidades reales de los usuarios, esto es posible a través de la interacción entre el desarrollador y el dominio de la aplicación, el desarrollar consigue esa interacción a través del análisis o revisión de documentos y entrevistas con los usuarios a fin de identificar los roles y tareas (Lamarca, 2006).

El modelo de entrevista está compuesto por cuatro pasos:

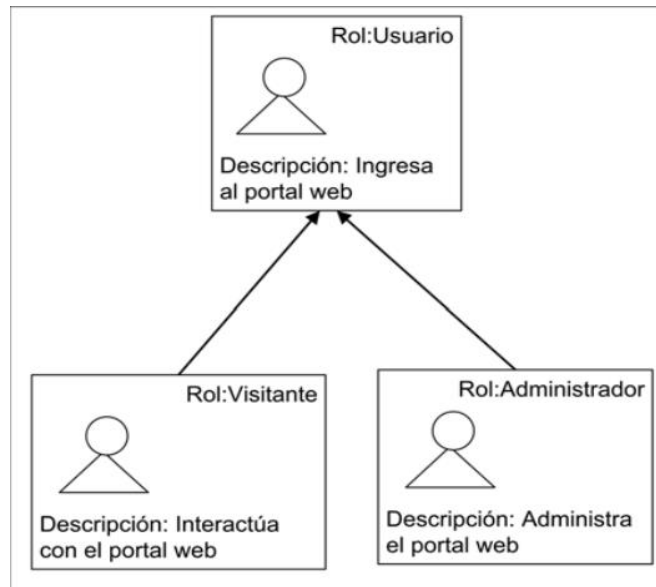
- Paso 1: identificar a los posibles entrevistados.
- Paso 2. Preparación de la entrevista.
- Paso 3. Efectuar la entrevista.
- Paso 4. Elaborar un informe de la entrevista.

Para la especificación de roles y tareas se plantea clasificar los tipos de actores de la aplicación web, ello implica identificar a las clases y sub clases de actores. Esta clasificación es importante ya que además de organizar la información, permite situarse mejor en el dominio de la aplicación y tener con claridad con que tipo de

usuarios se está trabajando, esto evita la confusión en los casos donde se tenían tareas compartidas por distintas clases de actores (Soto y Palma, 2004).

Figura 3

Clasificación de roles



NOTA: Información obtenida de (Soto y Palma, 2004)

En consecuencia, se hace el uso de una plantilla que permite definir las tareas correspondientes a cada rol. Esta plantilla consta de dos columnas, la primera donde está el “rol/actor” y la segunda donde se describen las “tareas”.

Figura 4

Definición de roles y tareas

Definición de tareas	
Rol	Tarea
Rol/Actor Usuario 1	*Tarea 1..... *Tarea 2..... *Tarea 3..... *Tarea 4.....
Rol/Actor Usuario 2	*Tarea 5..... *Tarea 6..... *Tarea 7..... *Tarea 8.....

NOTA: Información obtenida de (Soto y Palma, 2004)

2. Especificación de casos de uso

De acuerdo a la definición formal de UML un caso de uso es una secuencia de acciones que un sistema efectúa, para producir un resultado observable de una determinada tarea. Específicamente representa la interacción entre el usuario y el sistema (Boch, Rumbaugh, Jacobson, 2000)

Tabla 1

Especificación de casos de uso

Nombre:	Iniciar sesión	
Actor:	Usuario, administrador	
Propósito:	El actor se autentifica	
Resumen:	En este módulo se debe ingresar los datos para su verificación para una posterior autenticación.	
	Evento actor	Evento sistema
	1. El actor selecciona el link de inicio de sesión	3. El modulo muestra el formulario de inicio de sesión.
	2. El actor ingresa su	4. El modulo muestra el

usuario y contraseña.	formulario de inicio de sesión.
Pre condición	Estar en el formulario de inicio
Pos condición	Estar en el menú principal de la página de inicio con sesión

NOTA: Información obtenida de (Soto y Palma, 2004)

3. Especificación de UIs

La metodología OOHDM propone la utilización de la herramienta denominada UIs, cuyo objetivo es mejorar la interacción del usuario con el sistema (de forma gráfica), descrita de manera textual en la especificación de casos de uso, sin entrar en detalles relativos a interfaces del usuario. Para obtener un UIs desde un caso de uso la secuencia de información intercambiada entre el usuario y el sistema debe ser identificada y organizada en las interacciones. Identificar la información de intercambio es crucial ya que es la base para la definición de los UIs (Soto y Palma, 2004).

4. Validación de casos de uso y UIs.

En esta sub fase el desarrollador debe interactuar con cada usuario para validar UIs obtenidos, mostrando y explicando cada uno de ellos para ver si el o los usuarios están de acuerdo (Soto y Palma, 2004).

El usuario debe intervenir solo en aquellos casos de uso y UIs en los que participa. Cada sugerencia, problema e inconsistencia deben ser anotadas, el número de interacciones necesarias para llegar a un consenso dependerá del tiempo disponible del desarrollador como también de los usuarios (Soto y Palma, 2004).

Debido a la importancia que posee la validación de requisitos en todo proceso de desarrollo de software cualquiera que sea este, es que se hace necesario llevarlo a cabo de manera ordenada y entendible al usuario.

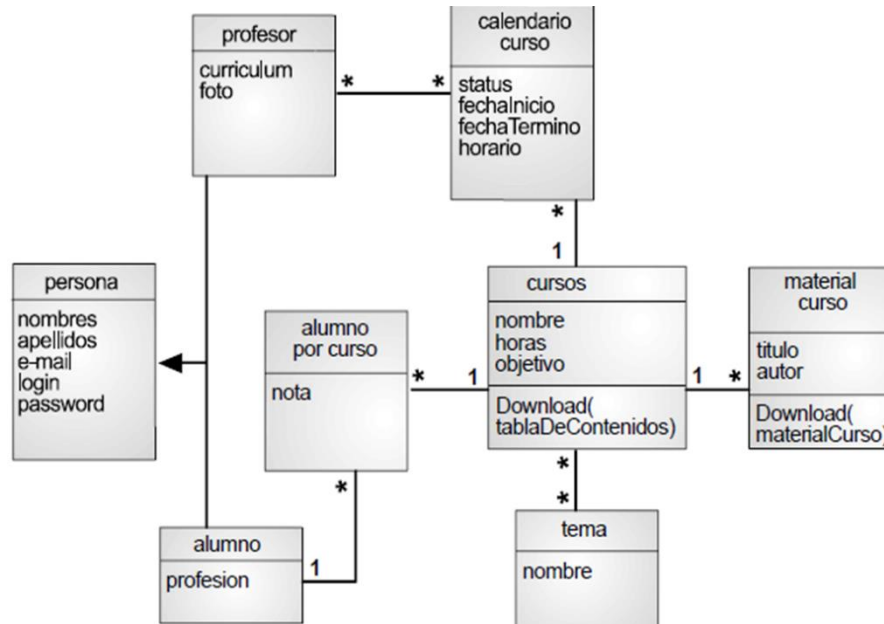
b) Fase II. Diseño conceptual

En esta fase se construye un modelo orientado a objetos que representa el dominio semántico de la aplicación, usando las técnicas propias de la orientación a objetos. Esta construcción es realizada mediante la definición de clases, subsistemas,

relaciones, jerarquías de agregación y especialización, la definición de los tipos de atributos de las clases y cardinalidades. (Soto y Palma, 2004).

Figura 5

Diseño Conceptual



NOTA: Información obtenida de (gestioninformacion y conocimiento, 2011)

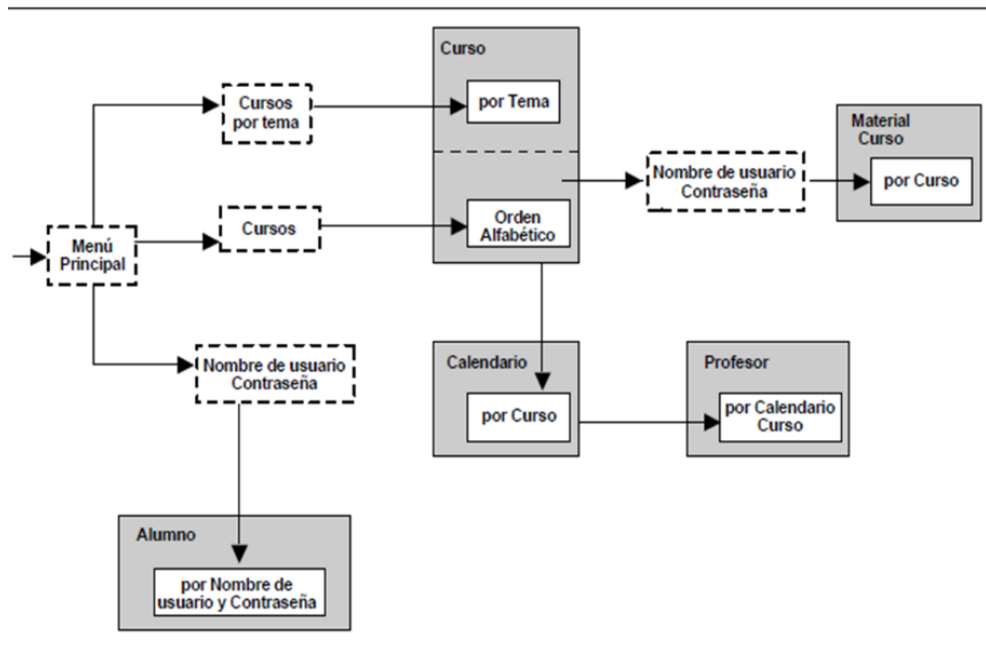
c) Fase III. Diseño navegacional

Los sitios Web y las aplicaciones hipermedia son aplicaciones proyectadas para la navegación a través de un espacio de información. La fase anterior propone la construcción de un modelo conceptual en el cual describe las clases de objetos y relaciones del dominio de la aplicación (Soto y Palma, 2004).

De este modelo conceptual se pueden originar varios de los modelos navegacionales es decir la información que será presentada a los usuarios y como se efectuará la navegación entre ellas (Soto y Palma, 2004).

Figura 6

Diseño Navegacional



NOTA: Información obtenida de (gestioninformacion y conocimiento, 2011)

d) Fase IV: Diseño de interfaces no abstractas

Finalizado el diseño navegacional es necesario especificar las diferentes interfaces de la aplicación. Esto significa definir de qué manera aparecerán los objetos navegacionales en la interfaz y cuales objetos activarán la navegación. Esta fase consiste en mostrar los prototipos de interfaz, de manera que el usuario pueda tener un concepto de las pantallas que vera una vez este implementado el trabajo.

e) Fase V. Implementación

En esta fase, el diseñador debe implementar el diseño. Hasta esta fase, todos los modelos fueron construidos en forma independiente de la plataforma de implementación; en esta fase se tiene en cuenta el entorno particular en el cual se va a correr la aplicación (Silva y Mercerat, 2003).

Al llegar a esta fase, el primer paso que debe realizar el diseñador es definir los ítems de información que son parte del dominio del problema. Debe identificar también, cómo son organizados los ítems de acuerdo con el perfil del usuario y su tarea; decidir qué

interfaz debería ver y cómo debería comportarse. A fin de implementar todo en un entorno web, el diseñador debe decidir además qué información debe ser almacenada (Silva y Mercerat, 2003).

2.14 ARQUITECTURA CLIENTE SERVIDOR

La arquitectura cliente-servidor consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

La red cliente-servidor es aquella red de comunicaciones en la que todos los clientes están conectados a un servidor, en el que se centralizan los diversos recursos y aplicaciones con que se cuenta; y que los pone a disposición de los clientes cada vez que estos son solicitados. Esto significa que todas las gestiones que se realizan se concentran en el servidor, de manera que en él se disponen los requerimientos provenientes de los clientes que tienen prioridad, los archivos que son de uso público y los que son de uso restringido, los archivos que son de sólo lectura y los que, por el contrario, pueden ser modificados, etc. Este tipo de red puede utilizarse conjuntamente en caso de que se esté utilizando en una red mixta.

2.14.1 Características

Un sistema Cliente/Servidor es un Sistema de Información distribuido basado en las siguientes características:

- Servicio: unidad básica de diseño. El servidor los proporciona y el cliente los utiliza.
- Recursos compartidos: Muchos clientes utilizan los mismos servidores y, a través de ellos, comparten tanto recursos lógicos como físicos.
- Protocolos asimétricos: Los clientes inician “conversaciones”. Los servidores esperan su establecimiento pasivamente.

- Transparencia de localización física de los servidores y clientes: El cliente no tiene por qué saber dónde se encuentra situado el recurso que desea utilizar.
- Independencia de la plataforma HW y SW que se emplee.
- Sistemas débilmente acoplados. Interacción basada en envío de mensajes.
- Encapsulamiento de servicios. Los detalles de la implementación de un servicio son transparentes al cliente.
- Escalabilidad horizontal (añadir clientes) y vertical (ampliar potencia de los servidores).

Integridad: Datos y programas centralizados en servidores facilitan su integridad y mantenimiento

2.14.2 Ventajas

- **Centralización del control:** Los accesos, recursos y la integridad de los datos son controlados por el servidor de forma que un programa cliente defectuoso o no autorizado no pueda dañar el sistema.
- **Escalabilidad:** Se puede aumentar la capacidad de clientes y servidores por separado. Cualquier elemento puede ser aumentado (o mejorado) en cualquier momento, o se pueden añadir nuevos nodos a la red (clientes y/o servidores).
- **Fácil mantenimiento:** Al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reemplazar, reparar, actualizar, o incluso trasladar un servidor, mientras que sus clientes no se verán afectados por ese cambio. Esta independencia de los cambios también se conoce como encapsulación.
- **Tecnología:** Existen suficientemente desarrolladas, diseñadas para el paradigma de C/S que aseguran la seguridad en las transacciones, la amigabilidad de la interfaz, y la facilidad de empleo.
- **Manejo de Capas:** Consiste en una capa de la Presentación, otra capa de la lógica de la aplicación y otra capa de la base de datos.

2.14.3 Desventajas

- Dificultad de la programación en comparación con modelo de dos capas ya que se comunican más dispositivos para realizar las peticiones de los clientes; además el incremento del tráfico en la red debido al flujo de los datos

2.14.4 Seguridad de modelo cliente /servidor

- Es de gran importancia por el valor intrínseco para los usuarios. Tiene tres componentes:
- Confidencialidad. - Protección contra individuos no autorizados.
- Integridad. - Protección contra la alteración o corrupción.
- Disponibilidad. - Protección contra la interferencia con los procedimientos de acceso a los recursos.

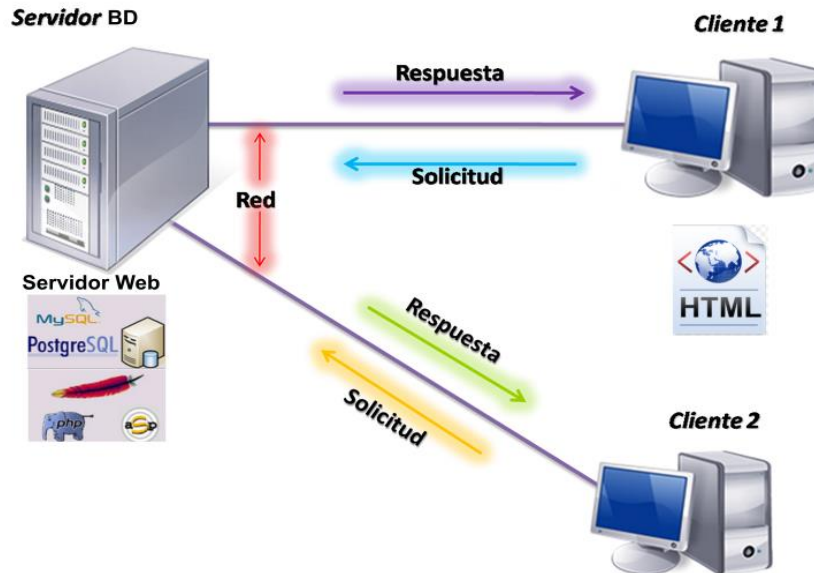
2.14.5 Arquitectura de dos y tres niveles cliente /servidor

2.14.5.1 Arquitectura dos niveles

La arquitectura en 2 niveles se utiliza para describir los sistemas cliente/servidor en donde el cliente solicita recursos y el servidor responde directamente a la solicitud, con sus propios recursos. Esto significa que el servidor no requiere otra aplicación para proporcionar parte del servicio.

Figura 7

Arquitectura en dos niveles



NOTA: Información obtenida de (Leal Castellano, Leal Molina, y Medina Castiblanco, 2011)

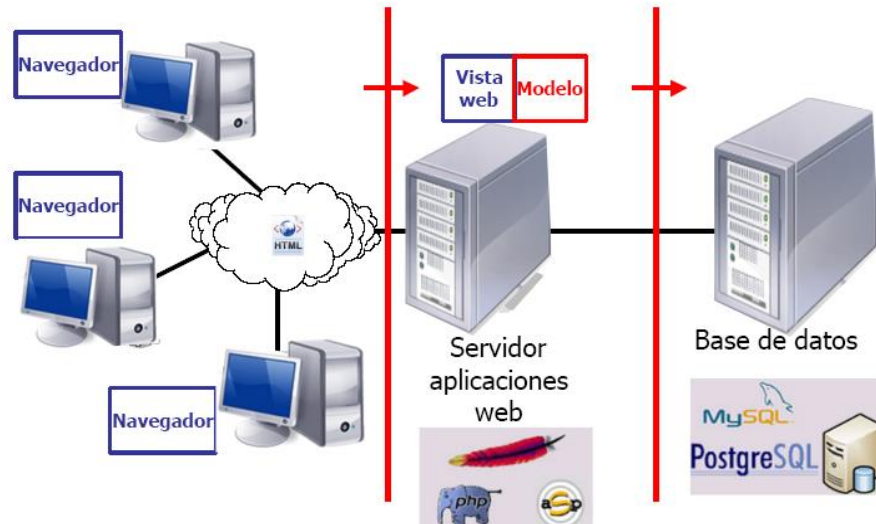
2.14.5.2 Arquitectura tres niveles

En la arquitectura en tres niveles, existe un nivel intermediario. Esto significa que la arquitectura generalmente está compartida por:

- Un cliente, es decir, el equipo que solicita los recursos, equipado con una interfaz de usuario (generalmente un navegador Web) para la presentación
- El servidor de aplicaciones (también denominado software intermedio), cuya tarea es proporcionar los recursos solicitados, pero que requiere de otro servidor para hacerlo
- El servidor de datos, que proporciona al servidor de aplicaciones los datos que requiere

Figura 8

Arquitectura Tres Niveles



NOTA: Información obtenida de: (Leal Castellano, Leal Molina, y Medina Castiblanco, 2011)

El uso masivo del término arquitectura en tres niveles también denota las siguientes arquitecturas:

- Aplicación compartida entre un cliente, un software intermedio y un servidor empresarial
- Aplicación compartida entre un cliente, un servidor de aplicaciones y un servidor de base de datos empresarial.

2.14.6 Comparación entre ambos tipos de arquitecturas

La arquitectura en dos niveles es, por lo tanto, una arquitectura cliente/servidor en la que el servidor es polivalente, es decir, puede responder directamente a todas las solicitudes de recursos del cliente.

Sin embargo, en la arquitectura en 3 niveles, las aplicaciones al nivel del servidor son descentralizadas de uno a otro, es decir, cada servidor se especializa en una determinada tarea, (por ejemplo: servidor web/servidor de bases de datos). La arquitectura en 3 niveles permite:

- Un mayor grado de flexibilidad
- Mayor seguridad, ya que la seguridad se puede definir independientemente para cada servicio y en cada nivel
- Mejor rendimiento, ya que las tareas se comparten entre servidores

Tabla 2

Comparación de Arquitecturas

Dos niveles	Tres niveles
Consiste en una capa de presentación y lógica de la aplicación; y la otra de la base de datos. Cuando se requiera poco procesamiento de datos en la organización.	Consiste en una capa de la Presentación, otra capa de la lógica de la aplicación y la capa de la base de datos. Cuando se requiera mucho procesamiento de datos en la aplicación.
Cuando se tiene una base de datos centralizada en un solo servidor.	En aplicaciones donde la funcionalidad este en constante cambio.
Cuando la base de datos es relativamente estática.	Cuando los procesos no están relativamente muy relacionados con los datos.
Cuando se requiere un mantenimiento mínimo.	Cuando se requiera aislar la tecnología de la base de datos para que sea fácil de cambiar.
	Cuando se requiera separar el código del cliente para que se facilite el mantenimiento.
	Está muy adecuada para utilizarla con la tecnología orientada a objetos.

NOTA: Información obtenida de (Leal Castellano, Leal Molina, y Medina Castiblanco, 2011)

2.15 SEGURIDAD DE APLICACIONES WEB

La seguridad de las aplicaciones Web es un tema crítico y complejo para los desarrolladores Web. Ya que requiere realizar estudios y tener entendimiento sobre los puntos de vulnerabilidad en cada aplicación Web.

Se puede clasificar los puntos más importantes:

Seguridad en el servidor

- Servidor Web
- Servidor de Base de Datos
- Lenguaje de programación

Seguridad en la aplicación

- Control de Acceso
- Validación de datos de entrada
- Pruebas de código

Seguridad en la comunicación

- Protocolos de seguridad
- Certificados de Comunicación

Copias de seguridad

- Backup de la Base de Datos
- Backup del control de Acceso

Las pruebas de seguridad están diseñadas para probar la vulnerabilidad en el ambiente del lado del cliente, las comunicaciones de red que ocurren mientras los datos pasan del cliente al servidor y de vuelta y el ambiente del lado del servidor. Cada uno de estos dominios puede recibir ataques y es labor de quien prueba la seguridad descubrir las debilidades que pueden explotar quienes tengan la intención de hacerlo (Pressman R. , 2005)

2.16 MODELO VISTA CONTROLADOR (MVC)

El modelo MVC es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

2.16.1 Modelo

Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permitiendo comparar un número de unidades negativo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o importes en un carrito de la compra.

2.16.2 Vista

Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

2.16.3 Controlador

Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

2.16.4 Características del modelo vista controlador

Aunque se pueden encontrar diferentes implementaciones de MVC, la característica del flujo que sigue el control generalmente es el siguiente:

- El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.)

- El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
- El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el. Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
- El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo. El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta dirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio.
- La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

2.17 MÉTRICAS DE CALIDAD AL SOFTWARE

Las Métricas de Calidad proporcionan una indicación de cómo se ajusta el software, a los requerimientos implícitos y explícitos del cliente. (Abud, 2012)

Hablar de calidad del software implica la necesidad de contar con parámetros que permitan establecer los niveles mínimos que un producto de este tipo debe alcanzar para que se considere de calidad. El problema es que la mayoría de las características que definen al software no se pueden cuantificar fácilmente; generalmente, se establecen de forma cualitativa, lo que dificulta su medición, ya que se requiere establecer métricas que permitan evaluar cuantitativamente cada característica dependiendo del tipo de software que se pretende calificar. (Abud, 2012)

El objetivo principal de la ingeniería del software es producir un producto de alta calidad. Para lograr este objetivo, los ingenieros del software deben utilizar mediciones que evalúen la calidad del análisis y los modelos de desafío, el código fuente, y los casos de prueba que se han creado al aplicar la ingeniería del software. Para lograr

esta evaluación de la calidad en tiempo real, el ingeniero debe utilizar medidas técnicas que evalúan la calidad con objetividad, no con subjetividad. (Abud, 2012)

2.17.1 Métricas de calidad del modelo de ISO-9126

El estándar ISO/IEC 9126 ha sido desarrollado en un intento de identificar los atributos clave de calidad para un producto de software (Pressman R. , 2010)

Este estándar es una simplificación del Modelo de McCall (Losavio, 2003), e identifica seis características básicas de calidad que pueden estar presentes en cualquier producto de software. El estándar provee una descomposición de las características en sub características, que se muestran en la Tabla 2.1

Tabla 3

Características de Calidad – Modelo ISO/IEC 9126

Características	Sub Características	
Funcionalidad	<ul style="list-style-type: none"> • Adecuacion • Exactitud • Interoperatividad 	<ul style="list-style-type: none"> • Seguridad • Conformidad
Fiabilidad	<ul style="list-style-type: none"> • Madurez • Tolerancia a fallas 	<ul style="list-style-type: none"> • Recuperabilidad • Conformidad
Usabilidad	<ul style="list-style-type: none"> • Comprensibilidad • Facilidad de Aprendizaje • Operabilidad 	<ul style="list-style-type: none"> • Atracción • Conformidad
Eficiencia	<ul style="list-style-type: none"> • Comportamiento Temporal • Utilización de recursos 	<ul style="list-style-type: none"> • Conformidad
Mantenibilidad	<ul style="list-style-type: none"> • analizabilidad • Cambiabilidad • Estabilidad 	<ul style="list-style-type: none"> • Capacidad de prueba • Conformidad
Portabilidad	<ul style="list-style-type: none"> • Adaptabilidad • Facilidad de Instalación • Coexistencia 	<ul style="list-style-type: none"> • Remplazabilidad • Conformidad

NOTA: Información obtenida de (Pressman R. , 2010)

2.17.1.1 Funcionalidad

Es la capacidad del producto de software de satisfacer los requisitos funcionales prescritos y las necesidades implícitas de los usuarios y tiene las siguientes sub características:

- **Adecuación:** La capacidad del producto de software para proveer un adecuado conjunto de funciones para las tareas y objetivos especificados por el usuario.
- **Exactitud:** La capacidad del producto de software para proveer los resultados o efectos acordados con un grado necesario de precisión.
- **Interoperabilidad:** La capacidad del producto de software de interactuar con uno o más sistemas especificados.
- **Seguridad:** La capacidad del producto de software para proteger la información y los datos de modo que las personas o los sistemas no autorizados no puedan leerlos o modificarlos.
- **Conformidad:** La capacidad del producto software para adaptarse a los estándares, convenciones o regulaciones en leyes y prescripciones relativas a la funcionalidad.

2.17.1.2 Fiabilidad

La capacidad del producto de software para mantener un nivel de desempeño bajo condiciones establecidas, por un periodo tiempo y contempla las siguientes sub características.

- **Madurez:** La capacidad del producto de software para evitar fallas como resultado de errores en el software.
- **Tolerancia a Errores:** La capacidad del producto de software para mantener un nivel especificado de funcionamiento en caso de errores del software o de incumplimiento de su interfaz especificada.

- **Recuperabilidad:** La capacidad del producto de software para restablecer un nivel especificado de funcionamiento y recuperar los datos afectados directamente en el caso de una falla.
- **Conformidad:** La capacidad del producto software para adaptarse a estándares, convenciones y regulaciones referidas a la fiabilidad.

2.17.1.3 Usabilidad

Es la capacidad del producto de software de ser entendido, aprendido, usado y atractivo al usuario, cuando es utilizado bajo las condiciones especificadas y se divide en las siguientes subcaracterísticas.

- **Entendibilidad:** La capacidad del producto de software para permitir al usuario entender si el software es adecuado, y cómo puede ser utilizado para las tareas y las condiciones particulares de la aplicación.
- **Capacidad de Aprendizaje:** La capacidad del producto de software para permitir al usuario aprender su aplicación. Un aspecto importante a considerar aquí es la documentación del software.
- **Operabilidad:** La capacidad del producto de software para permitir al usuario operarlo y controlarlo. Los aspectos de propiedad, de cambio, de adaptabilidad y de instalación pueden afectar la operabilidad.
- **Atracción:** Capacidad del producto para atraer al usuario.
- **Conformidad:** La capacidad del producto software para adaptarse a estándares, convenciones y regulaciones referidas a la usabilidad.

2.17.1.4 Portabilidad

La capacidad del software para ser trasladado de un entorno a otro. El entorno puede incluir entornos organizacionales, de hardware o de software.

- **Adaptabilidad:** La capacidad del producto de software para ser adaptado a diferentes entornos especificados sin aplicar acciones o medios diferentes de los previstos para el propósito del software considerado.

- **Facilidad de instalación:** La capacidad del producto de software para ser instalado en un ambiente especificado.
- **Reemplazabilidad:** La capacidad del producto de software para ser utilizado en lugar de otro producto de software, para el mismo propósito y en el mismo entorno.
- **Coexistencia:** La capacidad del producto para coexistir con otro software independiente en un ambiente común compartiendo recursos.
- **Conformidad:** La capacidad del producto software para adaptarse a estándares referidas a la portabilidad.

2.17.1.5 Mantenibilidad

Capacidad del producto de software para ser modificado. Las modificaciones pueden incluir correcciones, mejoras o adaptación del software a cambios en el entorno, y especificaciones de requerimientos funcionales.

- **Analizabilidad:** La capacidad del producto de software para atenerse a diagnósticos de deficiencias o causas de fallas en el software o la identificación de las partes a ser modificadas.
- **Cambiabilidad:** La capacidad del software para permitir que una determinada modificación sea implementada.
- **Estabilidad:** La capacidad del producto de software para evitar efectos inesperados debido a modificaciones del software.
- **Capacidad de prueba:** La capacidad del software para permitir que las modificaciones sean validadas.
- **Conformidad:** La capacidad del producto software de cumplir los estándares o convenciones relativas a la mantenibilidad.

2.17.1.6 Eficiencia

La capacidad del producto de software para proveer un desempeño adecuado, de acuerdo a la cantidad de recursos utilizados y bajo las condiciones planteadas.

- **Comportamiento temporal:** La capacidad del producto de software para proveer tiempos adecuados de respuesta y procesamiento, y ratios de rendimiento cuando realiza su función bajo las condiciones establecidas.
- **Utilización de recursos:** La capacidad del producto de software para utilizar cantidades y tipos adecuados de recursos cuando este funciona bajo las condiciones establecidas.
- **Conformidad:** La capacidad del producto software de cumplir los estándares o convenciones relacionadas a la eficiencia.

Cada factor de calidad (FC) se puede obtener como combinación de una o varias métricas:

$$FC = C_1 M_1 + \dots + C_n M_n \quad \text{Ecuación 2.1}$$

Dónde:

FC: Es el factor de calidad

Cn: Coeficiente de regresión.

Mn: Métricas que afectaran al factor de calidad.

La calidad del software se calculara en función de los factores considerados y para cada factor las métricas establecidas, promediando los porcentajes de los factores de calidad obtenidos. Siendo el factor de calidad:

$$QT = \sum \frac{FACCA}{N^{\circ} \text{ de } FC}$$

2.18 PRUEBAS AL SOFTWARE

2.18.1 Pruebas de caja negra

Las Pruebas de Caja Negra: También suelen ser llamadas funcionales y basadas en especificaciones. En ellas se pretende examinar el programa en busca de que cuente con las funcionalidades que debe tener y como lleva a cabo las mismas, analizando siempre los resultados que devuelve y probando todas las entradas en sus valores válidos e inválidos. (Lynnett, 2021)

Al ejecutar las pruebas de Caja Negra se desarrollan casos de prueba reales para cada condición o combinación de condiciones y se analizan los resultados que arroja el sistema para cada uno de los casos. En esta estrategia se verifica el programa considerándolo una caja negra. Las pruebas no se hacen en base al código, sino a la interfaz. No importa que se cubran todas las rutas dentro del programa, lo importante es probar todas las entradas en sus valores válidos e inválidos y lograr que el sistema tenga una interfaz amigable. (Lynnett, 2021)

2.18.2 Pruebas de caja blanca

Las pruebas de caja blanca son una metodología que se utiliza para asegurar y validar el marco interno, los mecanismos, los objetos y los componentes de una aplicación de software. Las pruebas de caja blanca verifican el código de acuerdo con las especificaciones de diseño y descubren las vulnerabilidades de las aplicaciones. (Lynnett, 2021)

Las pruebas de caja blanca es una técnica de prueba de software en la que se prueba la estructura interna, el diseño y la codificación del software para verificar el flujo de entrada y salida y para mejorar el diseño, la usabilidad y la seguridad. En una prueba de caja blanca, los probadores aparecen en el código, por lo que se denomina prueba de caja limpia, prueba de caja abierta, prueba de caja transparente, prueba basada en código y prueba de caja de vidrio. (Martínez, 2016)

Según Martínez, las características de las pruebas de Caja Blanca son:

- Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

2.18.3 Pruebas de estrés

Una Prueba de Estrés es el proceso en el cual se eligen las actividades a probar en un sitio para codificarlas y ejecutarlas en un tiempo determinado desde una ubicación remota. Nos permiten identificar y planear ante la posibilidad de fallas en el

funcionamiento de la plataforma, preferiblemente de manera preventiva. (Cinquegrani, 2020)

Lo que se hace es medir la capacidad de la infraestructura, los tiempos de respuesta a actividades específicas correspondientes a un número incremental de usuarios generados de manera remota y buscar el número límite de usuarios antes de la negación de servicio de la plataforma. En otras palabras, cuantos usuarios interactuando de manera simultánea con la plataforma puede tolerar la infraestructura de tu sitio. (Cinquegrani, 2020)

2.19 ESTIMACIÓN DE COSTOS

2.19.1 Cocomo II

Este modelo permite realizar estimaciones en función del tamaño del software, y de un conjunto de factores de costo y de escala. Los factores de costo describen aspectos relacionados con la naturaleza del producto, hardware utilizado, personal involucrado, y características propias del proyecto. El conjunto de factores de escala explica las economías y des economías de escala producidas a medida que un proyecto de software incrementa su tamaño. (Coronado, 2016)

Y por otro lado existen diferentes modelos que define COCOMO II:

2.19.1.1 Modelo básico

Se basa exclusivamente en el tamaño expresado en LDC y se utiliza para obtener una primera aproximación rápida del esfuerzo. (FERNANDEZ, 2016)

2.19.1.2 Modelo intermedio

Este añade al modelo básico quince modificadores opcionales para tener en cuenta en el entorno de trabajo, incrementando así la precisión de la estimación además del tamaño del programa incluye un conjunto de medidas subjetivas llamadas conductores de costes. (FERNANDEZ, 2016)

Tabla 4

Coefficiente de Modelo Cocomo II

Proyecto de Software	a	b	c	d
Orgánico	2.40	1.05	2.50	0.38
Semiacoplado	3.00	1.12	2.50	0.35
Empotrado	3.60	1.20	2.50	0.32

NOTA: Información obtenida de (FERNANDEZ, 2016)

2.19.1.3 Modelo detallado

Presenta principalmente dos mejoras respecto al anterior. Los factores correspondientes a los atributos son sensibles o dependientes de la fase sobre la que se realizan las estimaciones. Aspectos tales como la experiencia en la aplicación, utilización de herramientas de software, etc., tienen mayor influencia en unas fases que en otras, y además van variando de una etapa a otra.

Para realizar COCOMO II necesitaremos conocer el número de líneas de código, posteriormente para poder realizar los cálculos del método de estimación usaremos las siguientes ecuaciones.

Tabla 5*Ecuaciones del Modelo de Cocomo II*

Variable	Ecuación	Tipo de unidad
Esfuerzo requerido por el proyecto.	$E = a * (KLDC)^b$	Personas/Mes
	$* FAE$	
Tiempo requerido por el proyecto	$T = c * (E)^d$	Meses
Número de personas requeridas para el proyecto.	$NP = \frac{E}{T}$	Personas
Costo Total	CT = Sueldo Mes * NP * T	\$us

NOTA: Información obtenida de (ISO/IEC 27002, 2018)

2.20 HERRAMIENTAS

2.20.1 Herramientas de desarrollo

2.20.1.1 Gestor de base de datos Mariadb

MariaDB Server es uno de los servidores de bases de datos más populares del mundo. Está hecho por los desarrolladores originales de MySQL y se garantiza que permanecerá como código abierto.

MariaDB convierte los datos en información estructurada en una amplia gama de aplicaciones, desde banca hasta sitios web. Originalmente diseñado como un reemplazo directo mejorado para MySQL, MariaDB se usa porque es rápido, escalable y robusto, con un rico ecosistema de motores de almacenamiento, complementos y muchas otras herramientas que lo hacen muy versátil para una amplia variedad de casos de uso.

Características de MariaDB

MariaDB tiene características interesantes que lo han convertido en uno de los sistemas de gestión de bases de datos más utilizados. Podemos destacar entre los puntos fuertes que definen a este sistema de gestión de bases de datos:

Mejoras en la velocidad

MariaDB es muy rápida a la hora de realizar consultas complejas gracias al uso del motor aria, que utiliza el caché para almacenar las filas de datos, en lugar de escribir en disco. También se han eliminado conversiones innecesarias de juegos de caracteres que consiguió incrementar su velocidad entre el 1 y el 5 %. Con estas y otras características se puede decir que MariaDB es un sistema rápido.

Licencia GPL

MariaDB se distribuye bajo la licencia GPL, por lo que se trata de un sistema de software libre que puede utilizarse de manera gratuita en cualquier proyecto.

MariaDB se desarrolla como software de código abierto y como base de datos relacional proporciona una interfaz SQL para acceder a los datos. (Felipe, 2020)

2.20.1.2 Lenguaje de programación PHP

PHP es un lenguaje de programación destinado a desarrollar aplicaciones para la web y crear páginas web, favoreciendo la conexión entre los servidores y la interfaz de usuario.

PHP es un lenguaje de programación para desarrollar aplicaciones y crear sitios web que conquista cada día más seguidores. Fácil de usar y en constante perfeccionamiento es una opción segura para aquellos que desean trabajar en proyectos calificados y sin complicaciones. (Souza, 2020)

2.20.1.3 Framework Codeigniter

Codeigniter es preferido como framework para aplicaciones web por los desarrolladores que priorizan la velocidad a un abanico avanzado de funciones. El objetivo central que ha guiado el diseño de este entorno de desarrollo PHP de código abierto es, según la página web del proyecto, obtener el máximo en rendimiento y flexibilidad con un mínimo de código. (canao, 2022)

2.20.1.4 La estructura de codeigniter

El diseño orientado al rendimiento de este framework de desarrollo web se revela en su parca arquitectura, pues se basa en el patrón Modelo-Vista-Controlador (MVC). El principio fundamental que sustenta a la arquitectura de desarrollo MVC es la estricta separación entre el código y la presentación, gracias a una estructura modular de software y a la externalización del código PHP. Esta separación se realiza en estos tres grupos: el modelo (model), la vista (view) y el controlador (controller), que explicamos a continuación:

- **EL MODELO** representa la estructura de datos de una aplicación web desarrollada con Codeigniter. Para ello, en el código fuente se definen las denominadas clases (“model classes”), que contienen funciones especiales con las cuales se puede recibir, insertar o actualizar la información de la base de datos. (*Codeigniter, el peso pluma de los frameworks PHP, 2022*)
- **LA VISTA** es aquello que se le presenta al usuario final. Por lo general, se trata de un documento HTML en el cual se ha insertado contenido de forma dinámica con PHP, convirtiéndose en una especie de plantilla. Codeigniter también permite definir fragmentos de una página web como la cabecera y el pie de página o páginas RSS como vista. Normalmente las aplicaciones web utilizan varias vistas, que toman su contenido desde el mismo modelo, de tal forma que es posible presentar diversas características del programa en vistas diferentes. (*Codeigniter, 2022*)
- **EL CONTROLADOR** media entre el modelo, la vista y cualquier otro recurso necesario para procesar una petición HTTP o generar una página web de forma

dinámica. Este componente recibe las peticiones entrantes, valida la entrada, selecciona la vista deseada y le entrega el contenido que el modelo ha cargado desde una base de datos. (CodeIgniter, 2022)

2.20.2 Herramientas de diseño

2.20.2.1 Framework bootstrap

Bootstrap es un framework front-end utilizado para desarrollar aplicaciones web y sitios mobile first, o sea, con un layout que se adapta a la pantalla del dispositivo utilizado por el usuario.

Bootstrap es un framework CSS utilizado en aplicaciones front-end es decir, en la pantalla de interfaz con el usuario para desarrollar aplicaciones que se adaptan a cualquier dispositivo.

El framework combina CSS y JavaScript para estilizar los elementos de una página HTML. Permite mucho más que, simplemente, cambiar el color de los botones y los enlaces. (Arturo, 2020)

Bootstrap es un framework CSS de código abierto que favorece el desarrollo web de un modo más sencillo y rápido. Incluye plantillas de diseño basadas en HTML y CSS con la que es posible modificar tipografías, formularios, botones, tablas, navegaciones, menús desplegables, etc. También existe la posibilidad de utilizar extensiones de Javascript adicionales. (Urrutia, 2021)

2.20.2.2 CSS (Cascading Style Sheets)

La especificación CSS, junto con HTML y JavaScript es una tecnología usada por muchos sitios web para crear páginas con un diseño agradable. (Content, 2021)

El CSS (Cascading Style Sheets), en español Hojas de Estilo en Cascada, fue desarrollado por W3C en 1996. (Content, 2021)

Se le denomina Hojas de Estilos en Cascada porque las características se aplican de arriba a abajo mediante reglas que poseen un esquema prioritario. (Content, 2021)

Esta especificación es un lenguaje de diseño gráfico que se escribe dentro del código HTML del sitio web y, permite crear páginas de una manera más exacta y aplicarles estilos (colores, márgenes, formas, tipos de letras, etc.) por lo que se tiene mayor control de los resultados finales. (Content, 2021)

2.20.2.3 HTML (Hypertext Markup Language)

El HTML es parte fundamental de las normas web, en conjunto con otras tecnologías como el CSS o el JavaScript. Ya que este lenguaje es la base para construir páginas en Internet, es importante conocerlo para crear nuestras propias webs. (Mousinho, 2021)

HTML son las siglas en inglés para Hypertext Markup Language, que se traduce como lenguaje de marcación de hipertexto. (Mousinho, 2021)

A pesar de tener un nombre complicado, el HTML no es más que un lenguaje usado para crear páginas webs por medio de marcadores (tags) y atributos, que definen cómo el contenido va a ser presentado en un navegador web. (Mousinho, 2021)

2.20.2.4 Ajax

AJAX significa JavaScript asíncrono y XML (Asynchronous JavaScript and XML). Es un conjunto de técnicas de desarrollo web que permiten que las aplicaciones web funcionen de forma asíncrona, procesando cualquier solicitud al servidor en segundo plano. (Gustavo B., 2022)

2.20.2.5 JavaScript

JavaScript es un lenguaje de programación o de secuencias de comandos que te permite implementar funciones complejas en páginas web, cada vez que una página web hace algo más que sentarse allí y mostrar información estática para que la veas, muestra oportunas actualizaciones de contenido, mapas interactivos, animación de Gráficos 2D/3D, desplazamiento de máquinas reproductoras de vídeo, etc., puedes apostar que probablemente JavaScript está involucrado. Es la tercera capa del pastel

de las tecnologías web estándar, dos de las cuales (HTML y CSS) hemos cubierto con mucho más detalle en otras partes del Área de aprendizaje. (Gustavo B., 2022)

JavaScript es un lenguaje de programación muy conocido. Entre otras funciones, gestiona el contenido dinámico de un sitio web y permite la interacción dinámica del usuario. (Gustavo B., 2022)

2.20.2.6 Jquery

JQuery es una librería perteneciente al lenguaje de programación JavaScript, uno de los más usados en materia de desarrollo web, de código abierto y con la capacidad de mejorar la interactividad de una página web sin tener nociones de programación o un profundo conocimiento de este lenguaje. Es algo que facilita enormemente la tarea de desarrollo y diseño de páginas web. (NeoAttack, 2021)

CAPÍTULO III

MARCO

APLICATIVO

CAPÍTULO III

3 MARCO APLICATIVO

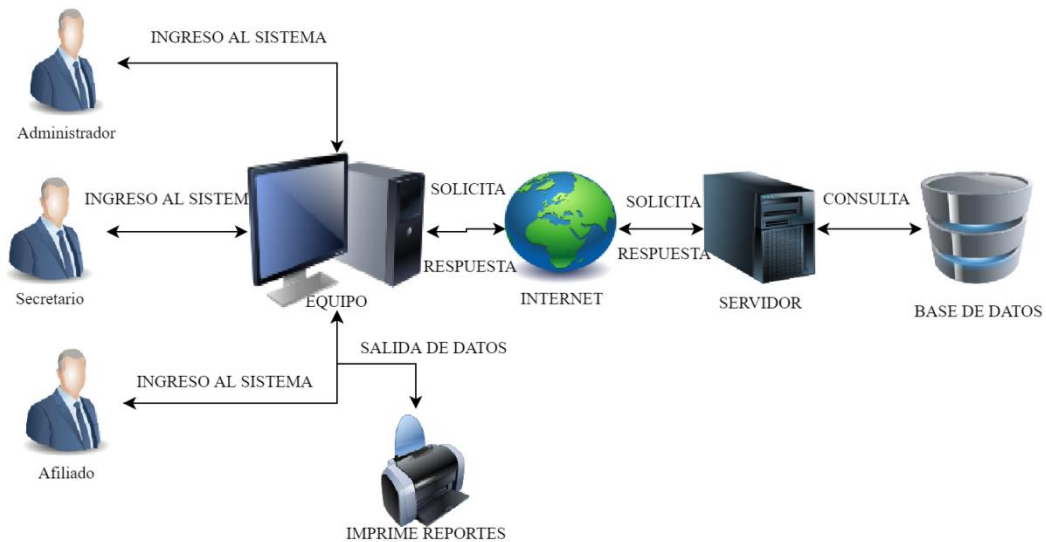
3.1 INTRODUCCIÓN

En el presente capítulo se aplicará la metodología OOHDM con los procesos de aplicación, análisis y diseño del desarrollo del sistema de administración y seguimiento de historiales clínicos odontológicos multisucursal para el Centro Odontológico Zonadental, utilizando los conceptos teóricos mencionados en el capítulo dos (marco teórico).

3.2 ESQUEMA DEL SISTEMA

Figura 9

Esquema del Sistema



3.3 DESARROLLO DE LA METODOLOGÍA OOHDM

Esta metodología está compuesta por cinco fases para su desarrollo:

- Obtención de Requerimientos
- Diseño conceptual
- Diseño navegacional

- Diseño de interfaces no abstractas
- Implementación

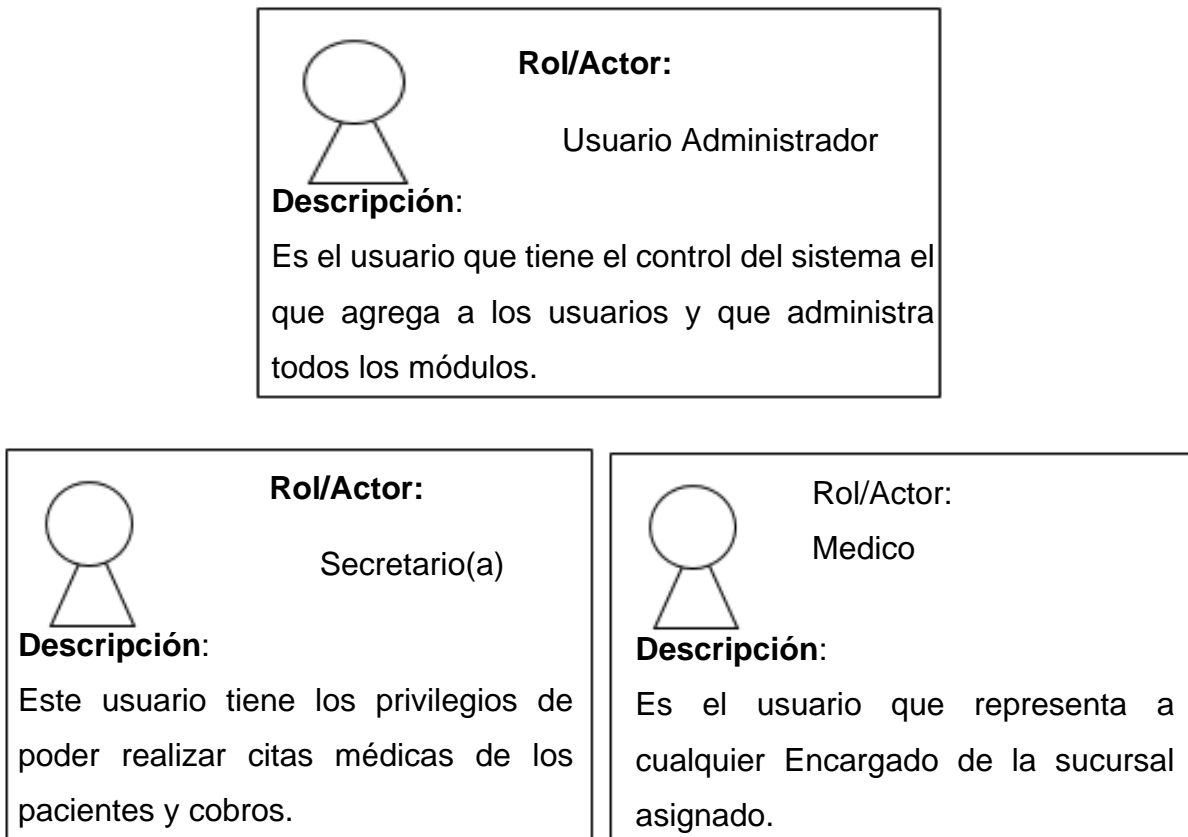
3.3.1 Obtención de requerimientos

El objetivo de esta fase es obtener todos los principales requerimientos del sistema web y participantes (actores) que interactuarán con el sistema, las necesidades que cubrirá el sistema.

3.3.1.1 Identificación de actores y tareas

Figura 10

Identificación de actores



3.3.1.2 Especificación de las tareas

Una vez identificado los roles que compondrán la administración del sistema, se describe las tareas que realiza cada usuario. Esta plantilla está formada por dos columnas: en la primera columna están los diferentes roles y en la segunda columna las tareas correspondientes a cada actor.

Tabla 6

Especificaciones de las tareas.

ROL/ACTOR	TAREAS
Rol/Actor: Usuario Administrador	T1. Administrar usuarios T2. Administrar privilegios T3. Administrar de sucursales T4. Administrar de personal clínica. T5. Administrar el módulo tipos tratamiento. T6. Administrar módulos pacientes. T7. Administrar el módulo citas. T8. Módulo historial de citas T9. Módulo historial pagos. T10. Módulo realizar pagos. T11. Módulo concepto de pagos. T12. Módulo otros ingresos. T13. Módulo egresos. T14. Reporte de ingresos. T15. Reporte de egresos T16. Reporte de ingresos y egresos
Rol/Actor: Secretario(a)	T6. Administrar módulos pacientes. T7. Administrar el módulo citas. T8. módulo historial de citas T10. Módulo realizar pagos.
Rol/Actor:	T6. Administrar módulos pacientes.


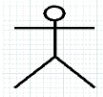
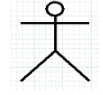
Medico	T7. Administrar el módulo citas. T8. Módulo historial de citas T10. Módulo realizar pagos. T11. Módulo concepto de pagos. T12. Módulo otros ingresos. T13. Módulo egresos.
--------	---

3.3.1.3 Especificación de casos de uso

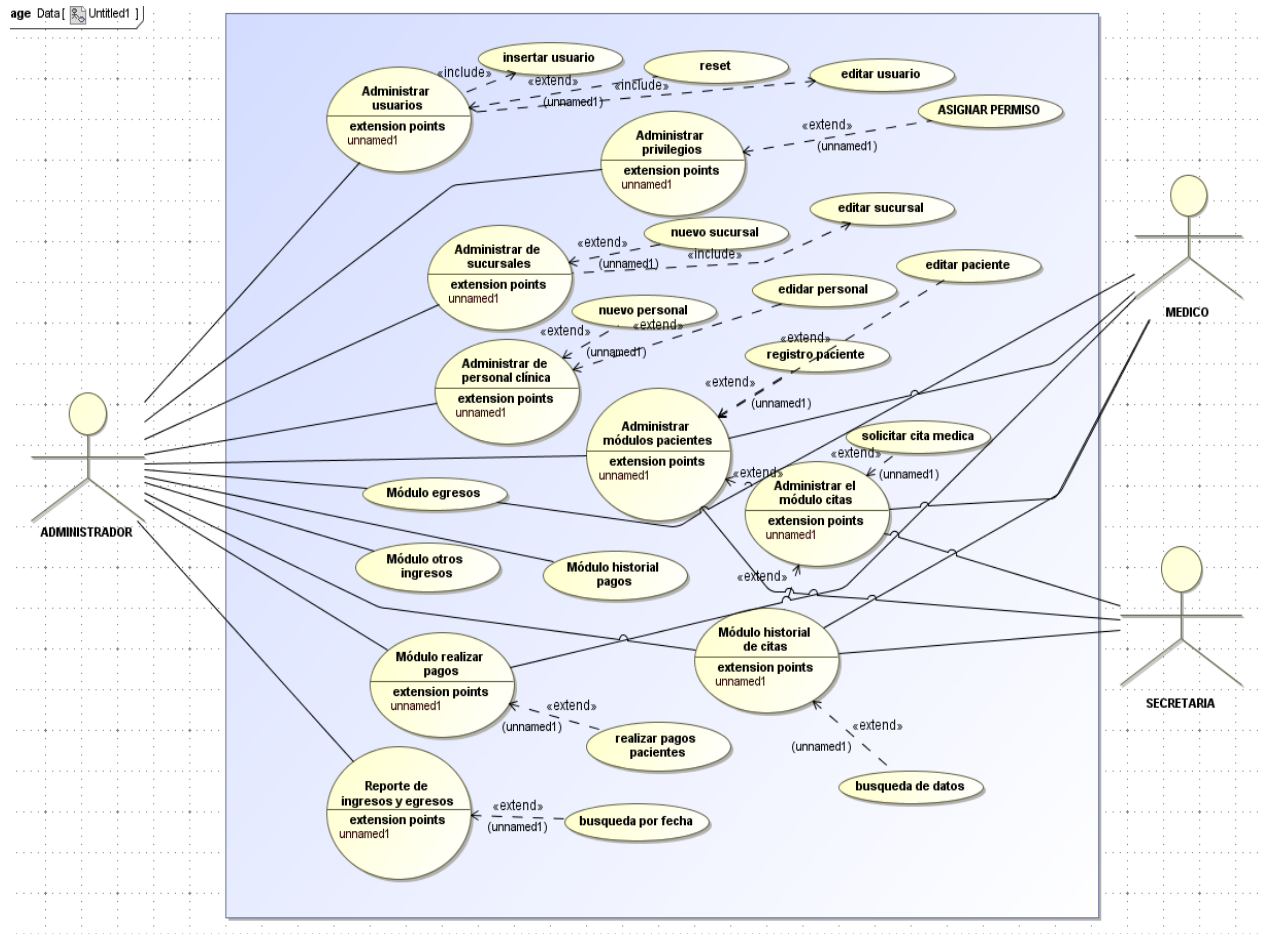
Luego de la identificación de actores y tareas que componen la elaboración de este sistema web se podrán obtener los casos de uso. En esta sub fase se detalla la relación que existe entre el usuario y las tareas que desempeña cada uno de ellos en el portal web a través de una plantilla de especificación de casos de uso para especificar los distintos escenarios, a continuación, se describe las características de los actores identificados.

Tabla 7

Descripción de los actores del sistema

ACTOR	DESCRIPCIÓN
	Usuario Administrador
	Usuario secretaria
	Usuario medico

3.3.1.3.1 Caso de uso: general



3.3.1.3.2 Caso de Uso: Usuarios Administración

Figura 11

Caso de Uso: Usuarios Administración

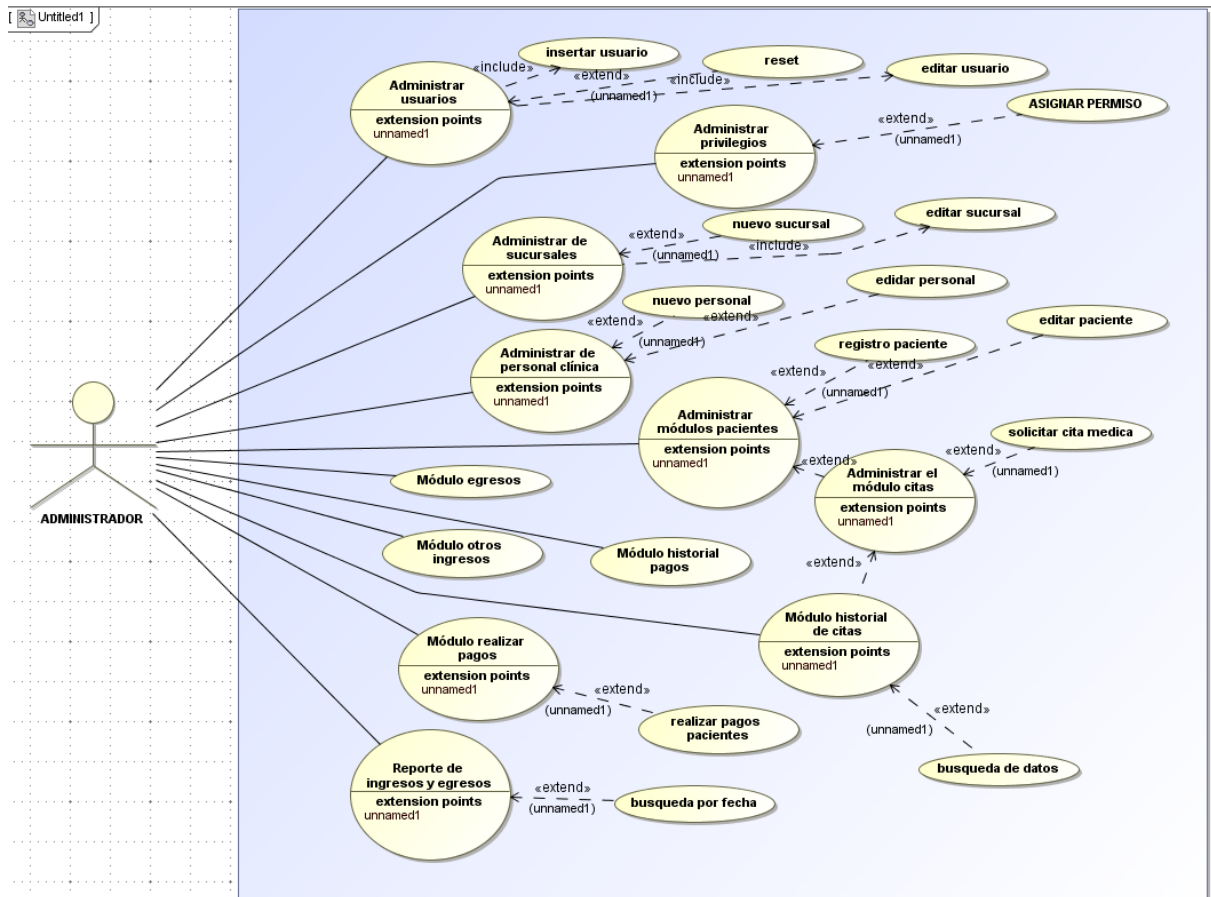


Tabla 8

Caso de Uso: Usuarios Administración

Caso de Uso: Usuarios Administración	
Nombre:	Usuarios Administración
Actor:	Administrador
Descripción:	El usuario administrador contempla todos los privilegios del sistema como ser los siguientes módulos que misionaremos a continuación.

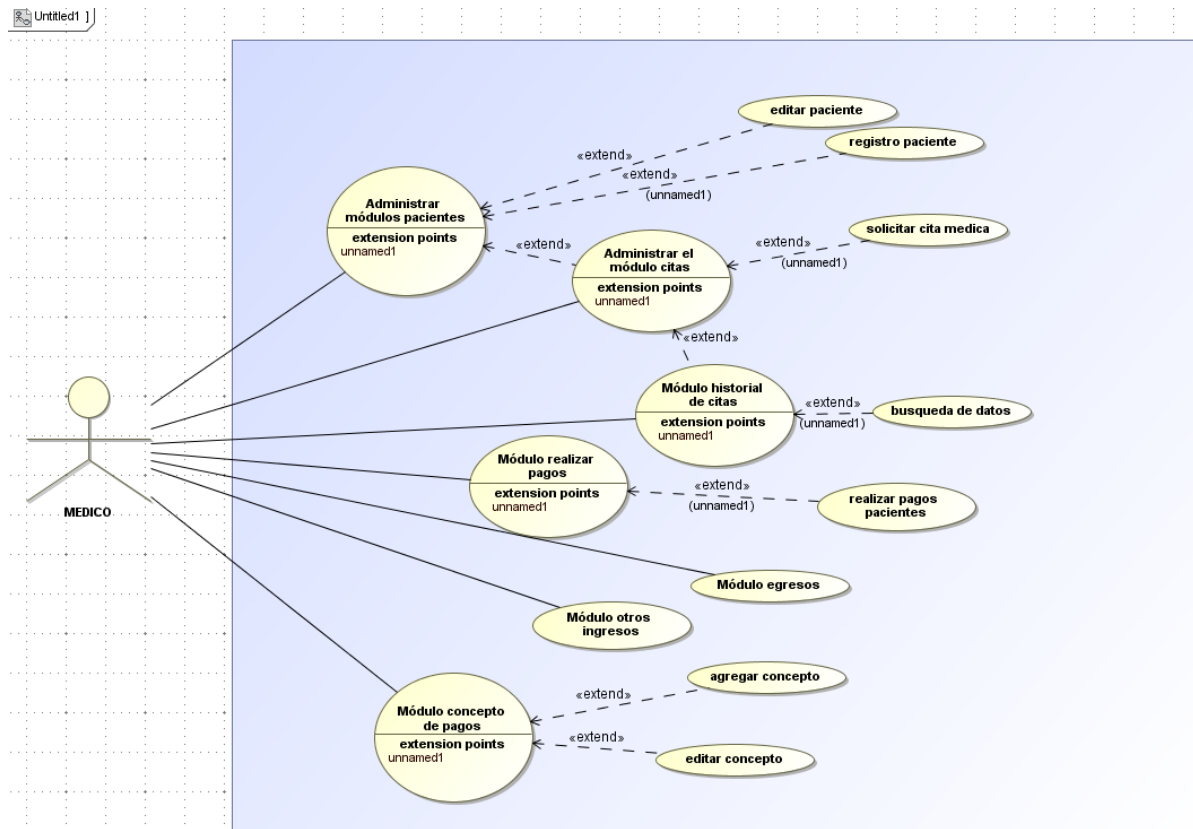
-
- Módulo de usuario: en este módulo permite registrar un nuevo usuario, editar usuario, cambiar estado y poder resetear su usuario y contraseña.
 - Módulo de privilegios: Este módulo nos permite admitir los privilegios a distintos roles de forma dinámica los menús.
 - Módulo de sucursales: en el presente modulo nos permite registra las sucursales, editar datos de sucursales y cambiar estado.
 - Módulo de administración de personal clínica: en este módulo nos permite realizar un nuevo registro de personal, editar datos y cambiar estado.
 - Módulo de paciente: en este módulo nos permite realizar un nuevo registro de paciente, editar datos de paciente y cambiar estado de paciente.
 - módulo citas: en este módulo nos permite realizar citas médicas.
 - Historial de citas médicas: en este módulo nos permite mostrar un historial de citas médicas del paciente.
 - historial pagos: el presente modulo nos permite presentar un historial de pagos de todos los pacientes de distintas sucursales.
 - realizar pagos: el presente modulo nos permite realizar cobros sobre los servicios realizados en la clínica.
 - Módulo de concepto de pagos: en el presente modulo se realiza sobre los conceptos de pagos como ser ingresos o egresos sobre la clínica.
 - Módulo de otros ingresos: en este módulo se realiza los registros de otros ingresos que contempla la clínica.
 - Módulo egresos: en este módulo se presenta los ingresos que tiene la clínica.
-

- Reporte de ingresos y egresos: en el presente modulo se realiza reportes por semana, mes y año.

3.3.1.3.3 Caso de Uso: Usuarios Medico

Figura 12

Caso de Uso: Usuarios Medico



NOTA: Información obtenida de (KOCH, 2000)

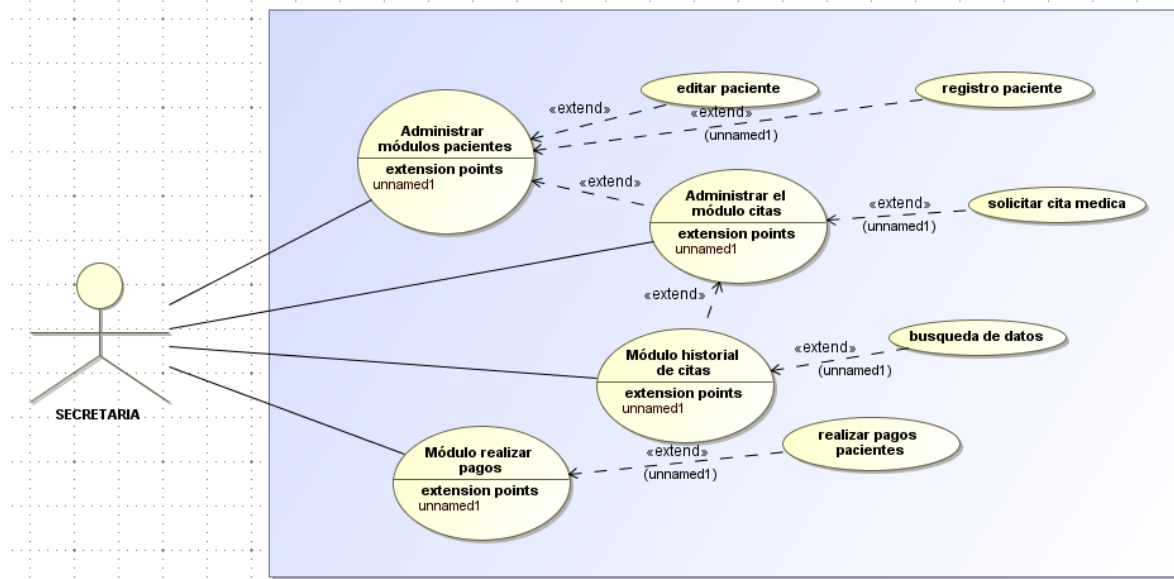
Tabla 9*Caso de Uso: Usuarios Medico*

Caso de Uso: Usuarios Medico	
Nombre:	Usuarios Medico
Actor:	Medico
Descripción:	<p>El usuario Medico contempla todos los privilegios del sistema como ser los siguientes módulos que misionaremos a continuación.</p> <ul style="list-style-type: none">• Módulo de paciente: en este módulo nos permite realizar un nuevo registro de paciente, editar datos de paciente y cambiar estado de paciente.• módulo citas: en este módulo nos permite realizar citas médicas.• Historial de citas médicas: en este módulo nos permite mostrar un historial de citas médicas del paciente.• Realizar pagos: el presente modulo nos permite realizar cobros sobre los servicios realizados en la clínica.• Módulo de concepto de pagos: en el presente modulo se realiza sobre los conceptos de pagos como ser ingresos o egresos sobre la clínica.• Módulo de otros ingresos: en este módulo se realiza los registros de otros ingresos que contempla la clínica.• Módulo egresos: en este módulo se presenta los ingresos que tiene la clínica.

3.3.1.3.4 Caso de Uso: Usuarios secretaria

Figura 13

Caso de Uso: Usuarios secretaria



NOTA: Información obtenida de (KOCH, 2000)

Tabla 10

Caso de Uso: Usuarios secretaria

Caso de Uso: Usuarios secretaria	
Nombre:	Usuarios secretaria
Actor:	secretaria
Descripción:	<p>El usuario Medico contempla todos los privilegios del sistema como ser los siguientes módulos que misionaremos a continuación.</p> <ul style="list-style-type: none"> Módulo de paciente: en este módulo nos permite realizar un nuevo registro de paciente, editar datos de paciente y cambiar estado de paciente.

-
- módulo citas: en este módulo nos permite realizar citas médicas.
 - Historial de citas médicas: en este módulo nos permite mostrar un historial de citas médicas del paciente.
 - Realizar pagos: el presente modulo nos permite realizar cobros sobre los servicios realizados en la clínica.
-

3.3.1.4 Especificación de UIs

En el diagrama de casos de uso se observa una interacción entre el usuario y el sistema durante la ejecución de tareas que se realizan cada uno de los actores. Con este tipo de diagramas podemos capturar los requisitos de la aplicación de manera independiente. Se define un diagrama de interacción de usuario para cada caso de uso.

Figura 14

UID: Administración de usuarios

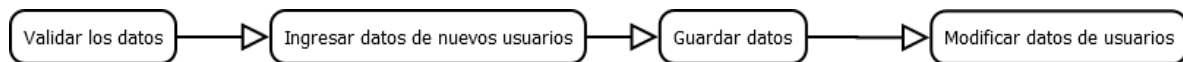
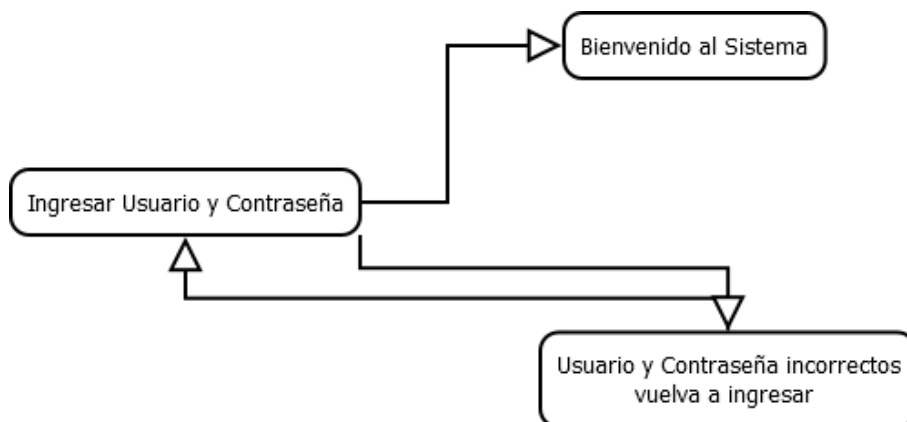


Figura 15

UID: Autenticación de usuario



3.3.2 Diseño conceptual

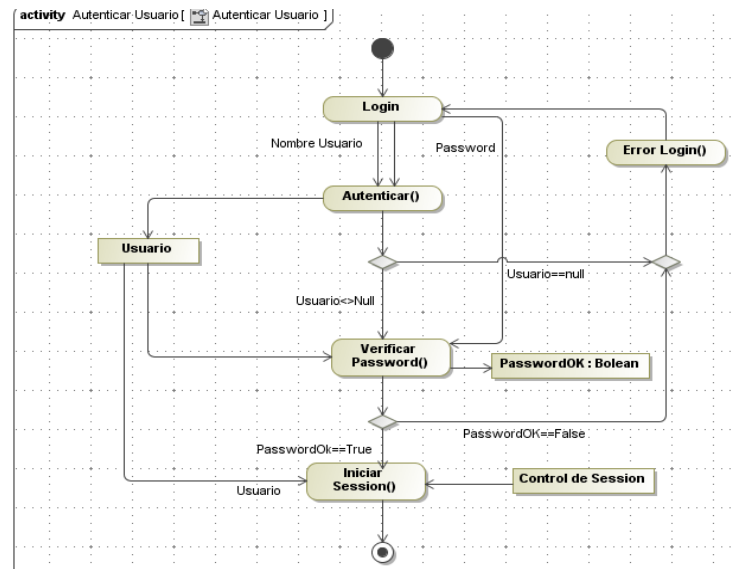
El diagrama de contenido tiene por propósito mostrar las relaciones entre las entidades y la estructura de los datos que se encuentran alojados en el sistema. En esta fase se construye un modelo orientado a objetos que representa el dominio semántico de la aplicación, usando las técnicas propias de la orientación a objetos. Esta construcción es realizada mediante la definición de clases, subsistemas, relaciones, jerarquías de agregación y especialización, la definición de los tipos de atributos de las clases y cardinalidades

3.3.3 Diseño navegacional

El diseño navegacional define la información que representa una posible navegación entre enlaces y nodos, se compone de componentes que contiene la especificación de su estructura interna de navegación, un punto de entrada, restricciones de accesos de usuario, operaciones y estructura de acceso asociada al nivel de cada usuario.

Figura 17

Modelo de Flujo de Proceso de Autenticar Usuarios al Sistema

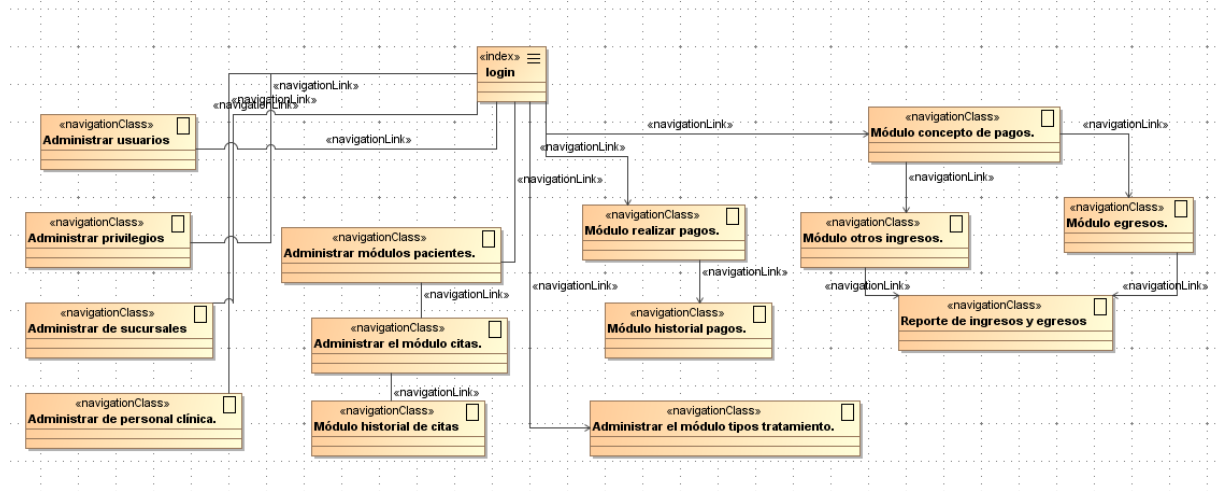


NOTA: Información obtenida de (KOCH, 2000)

3.3.3.1 Diseño de navegacional: Usuarios Administración

Figura 18

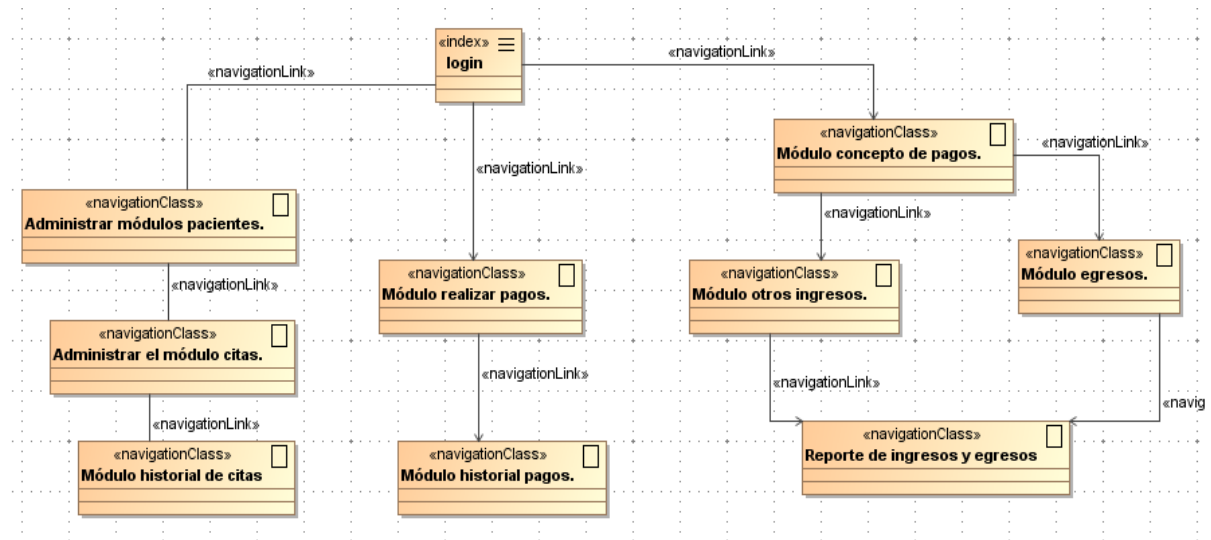
Diseño de navegacional: Usuarios Administración



3.3.3.2 Diseño de navegacional: Usuarios Medico

Figura 19

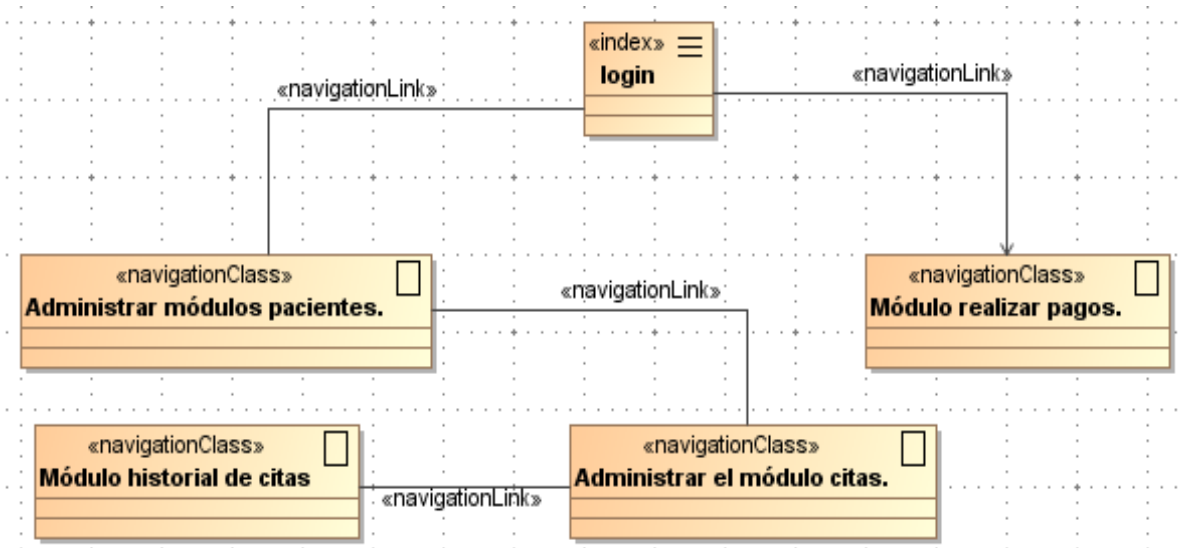
Diseño de navegacional: Usuarios Medico



3.3.3.3 Diseño de navegacional: Usuarios Secretaria

Figura 20

Diseño de navegacional: Usuarios Secretaria



3.3.4 Diseño de interfaz abstracta

Una vez observado el diseño navegacional y conociendo los nodos que se tiene que pasar para acceder a los distintos módulos del sistema y también conociendo el enfoque al diseño de interfaz abstracta, se utilizan elementos visuales no ligados a objetos del mundo real. Formas geométricas, colores y patrones creativos se emplean para transmitir información de manera conceptual y estilizada. También se define una interfaz gráfica que este compuesto de cuadros de texto, menús de selección, imágenes, etc. Para poder tener una fácil interacción entre el usuario y el sistema.

Figura 21

Diseño de interfaz abstracta login

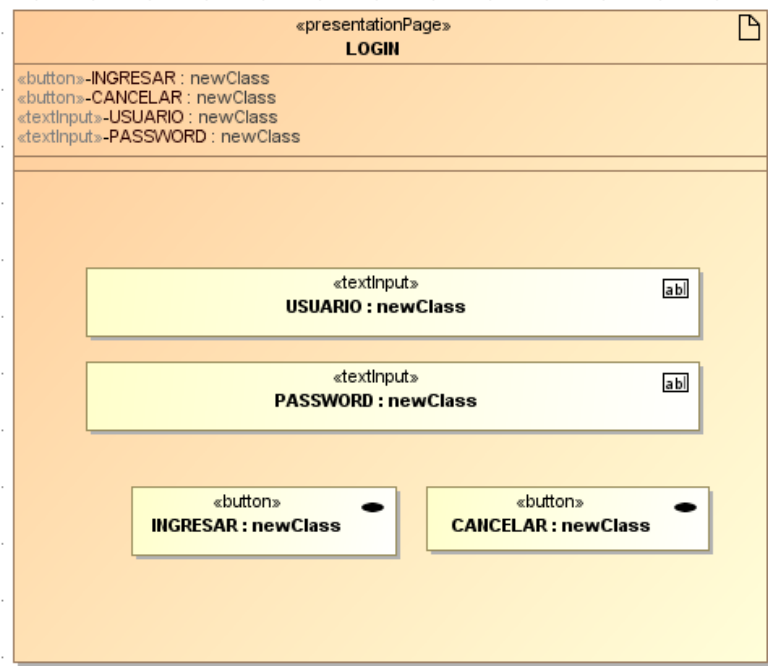
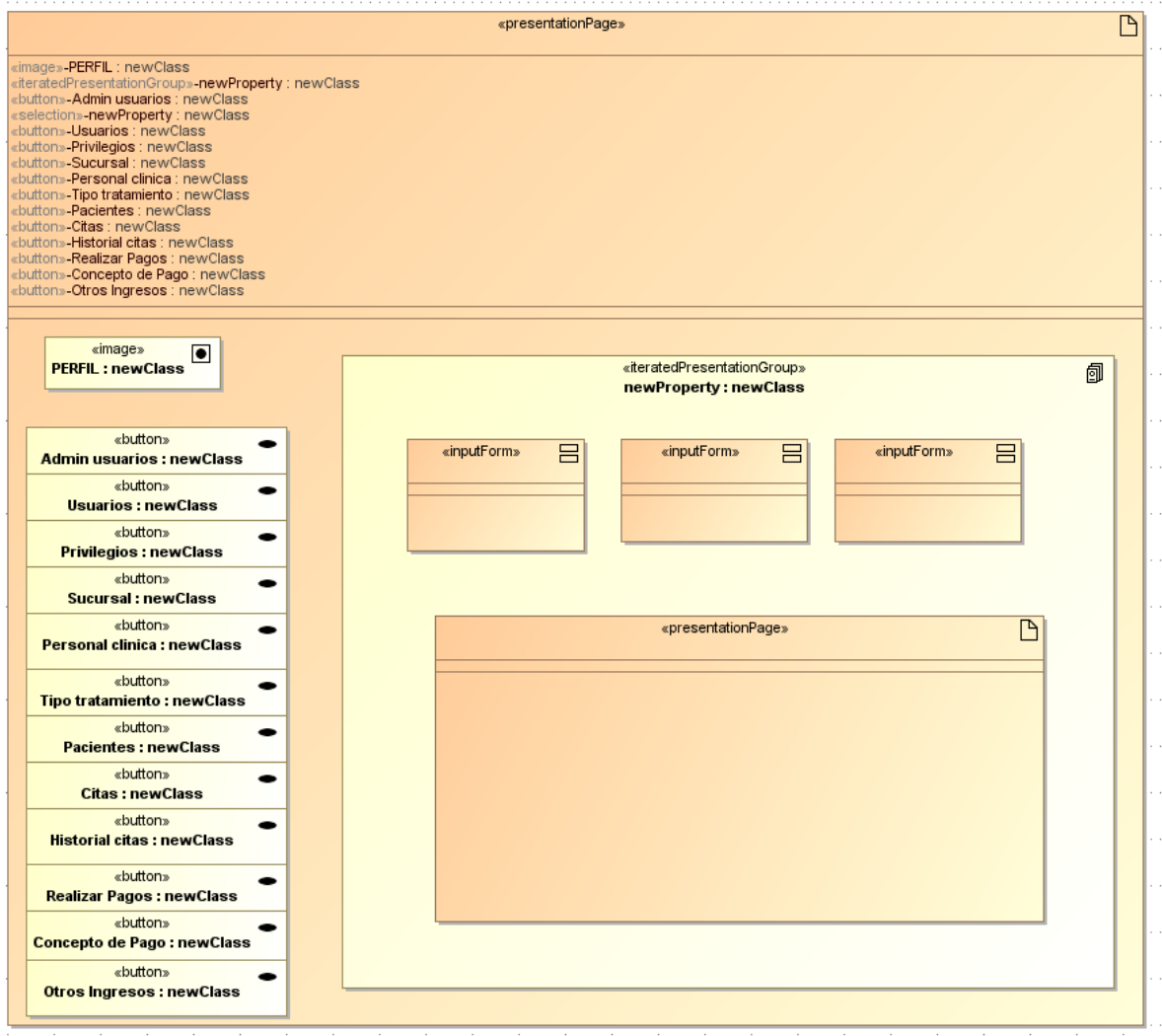


Figura 22

Diseño de interfaz abstracta administración



3.3.5 Implementación

En esta fase de implementación consiste en mostrar el desarrollo de la presentación del interfaz del sistema y sus elementos construidos a partir del diseño gráfico.

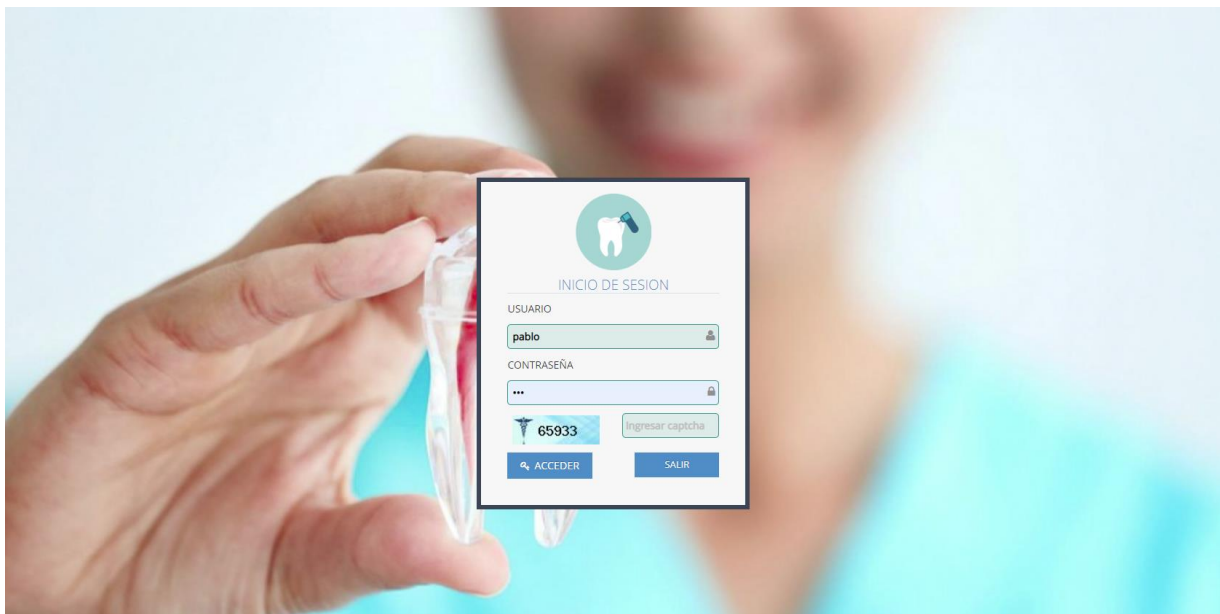
Figura 25
Codificación de Model

```

1 //php
2 class Modelo_sistema extends CF_Model
3 {
4     function __construct()
5     {
6         parent::__construct();
7         $this->load->database();
8         $this->default_timezone_set('America/LA_Paz');
9     }
10    public function eliminar_tabla_sys($tabla,$idtabla,$id)
11    {
12        $this->db->where($idtabla,$id);
13        $this->db->delete($tabla);
14    }
15    public function editar_tabla_sys($tabla,$sob,$idtabla,$id)
16    {
17        $this->db->where($idtabla,$id);
18        $this->db->update($tabla,$sob);
19    }
20    public function insertar_tabla_sys($tabla,$sob)
21    {
22        $this->db->insert($tabla,$sob);
23    }
24    public function tabla_row_sys($tabla,$idtabla,$id)
25    {
26        return $this->db->query("SELECT * FROM $tabla WHERE $idtabla='id' ")->row();
27    }
28    public function tabla_result_sys($tabla,$idtabla,$id)
29    {
30        return $this->db->query("SELECT * FROM $tabla WHERE $idtabla='id' ")->result();
31    }
32    public function listado_sucursales()
33    {
34        return $this->db->query(" SELECT * FROM sucursal WHERE su_estado='eliminar' ")->result();
35    }
36    public function listado_sucursales_activo()
37    {
38        return $this->db->query(" SELECT * FROM sucursal WHERE su_estado='activo' order by su_sucursal asc ")->result();
39    }
40    public function listado_especialidad_activo()
41    {
42        return $this->db->query(" SELECT * FROM especialidad WHERE esp_estado='activo' order by esp_nombre asc ")->result();
43    }
44    public function buscar_datos_exist_personal($rowid)
45    {
46        return $this->db->query("SELECT * FROM personal_clinico
47        INNER JOIN persona USING(idpersona)
48        WHERE persona.ci=".$codigo." AND personal_clinico.pe_estado='activo' ")->row();
49    }
50    public function listado_nuevopersonal()
51    {
52        return $this->db->query("SELECT * FROM personal_clinico
53        INNER JOIN persona USING(idpersona)
54        INNER JOIN sucursal USING(idsucursal)
55        INNER JOIN especialidad USING(idespecialidad)
56        WHERE persona.ci=".$codigo." AND personal_clinico.pe_estado='eliminar' ")->result();
57    }
58    public function listado_personal_activo()
59    {
60        return $this->db->query("SELECT
61        sucursal.su_sucursal,
62        persona.clinico.idpersonal_clinico,
63        persona.nombre,
64        persona.apellido,
65        persona.materno,
66        persona.ci,
67        idpersonal_clinico
68        INNER JOIN persona USING(idpersona)
69        INNER JOIN sucursal USING(idsucursal)
70        WHERE personal_clinico.pe_estado='activo' ")->result();
71    }
72    public function editarpersonal($idpersonal_clinico){
73    }
74 }
    
```

Figura 23

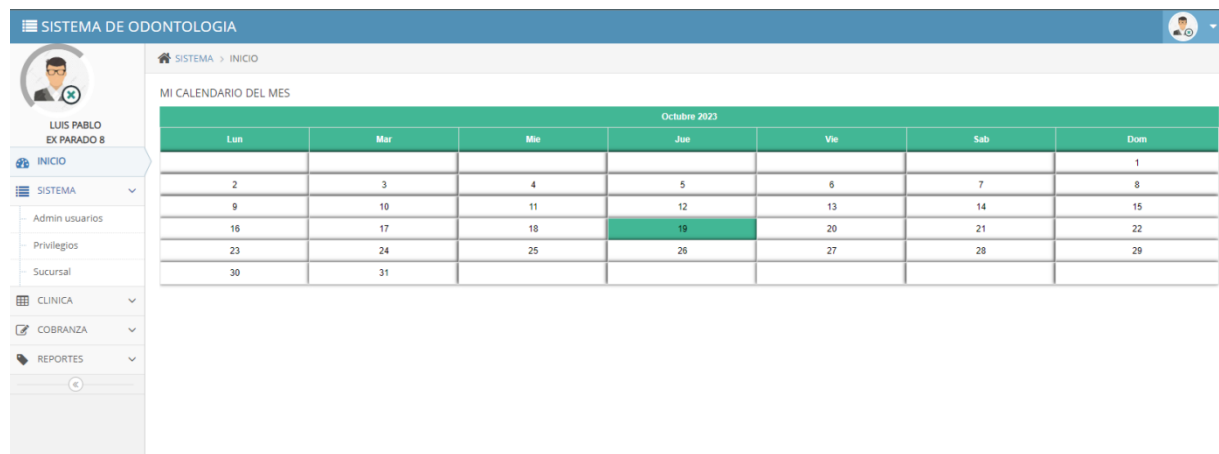
Diseño de interfaz de Inicio de Sesión.



Observamos la página de inicio de sesión del sistema donde debemos poner el usuario y la contraseña para poder acceder al sistema y realizar las peticiones y procesos que requerimos.

Figura 24

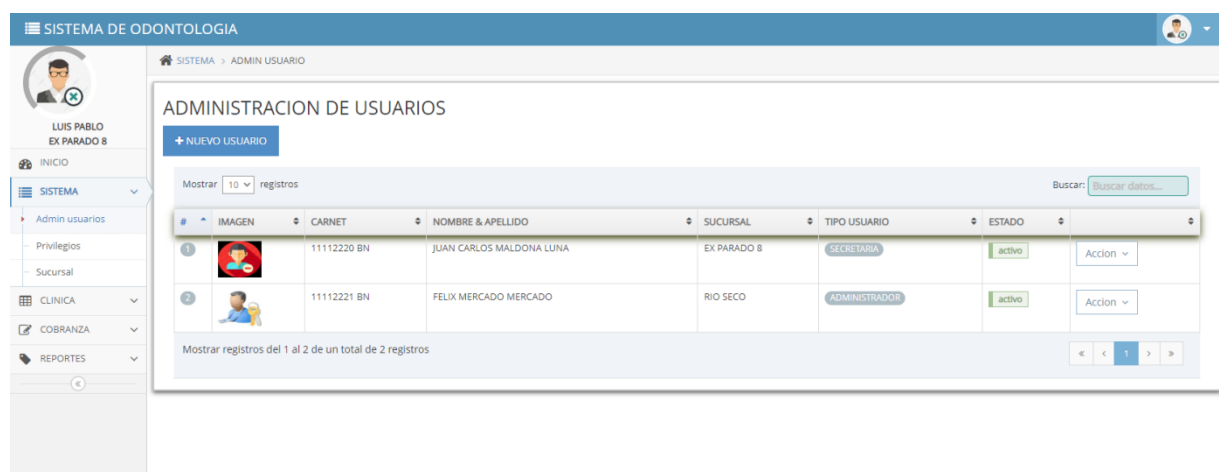
Diseño de interfaz de inicio de sistema



Una vez ingresado con un usuario y contraseña correcta nos permite mostrar el interfaz gráfico de inicio de sistemas.

Figura 25

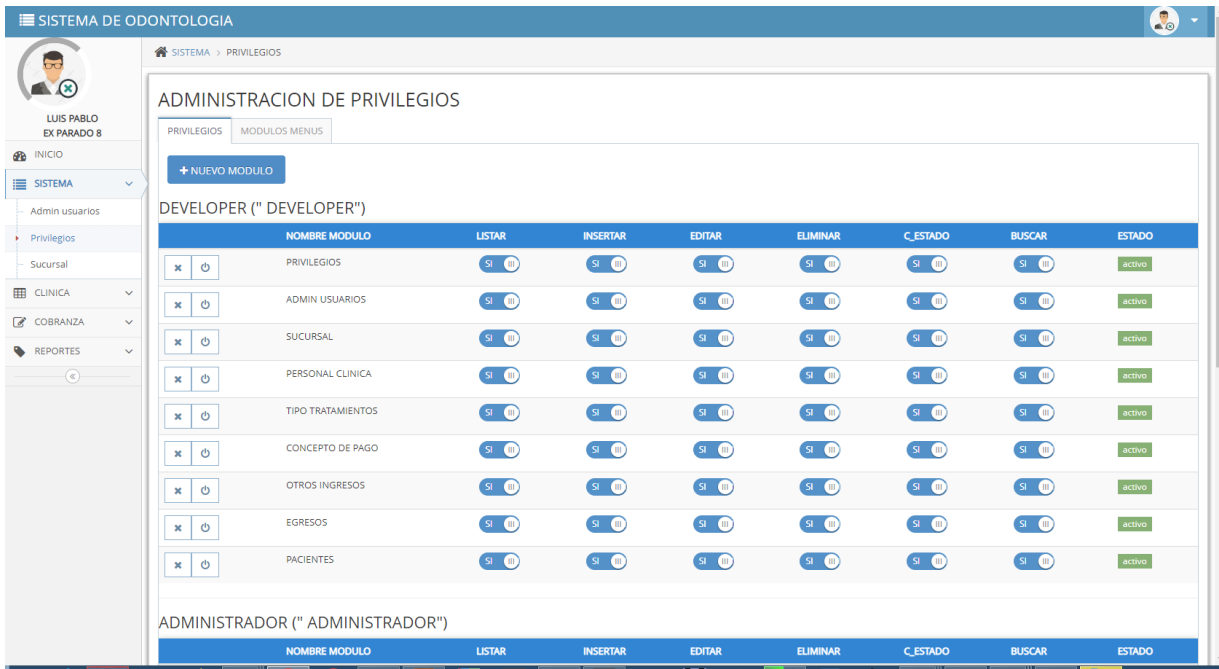
Diseño de interfaz de módulo de usuario



En este módulo de usuario se presenta el diseño de interfaz de administración de todos los usuarios que acceden al sistema.

Figura 26

Diseño de interfaz de módulo de privilegios



En este presente modulo se presenta una administración de control de privilegios a distintos roles de usuarios.

Figura 27

Diseño de interfaz de módulo de sucursales



En este módulo de administración de sucursales nos permite presentar un interfaz gráfico con los siguientes datos necesarios.

Figura 28

Diseño de interfaz de módulo de administración de personal de clínica



El presente modulo nos permite presentar a todo el personal de la clínica como ser doctores, secretarias y otros.

Figura 29

Diseño de interfaz de formulario de registro de paciente



El presente módulo nos permite mostrar un interfaz gráfico de un formulario de registro de un paciente.

Figura 30

Diseño de interfaz de odontograma de paciente

The screenshot displays a web-based interface for a dental clinic. At the top, there's a header with the system name 'SISTEMA DE ODONTOLOGIA' and a user profile for 'LUIS PABLO EX PARADO S'. A sidebar on the left contains navigation options like 'Inicio', 'Sistema', 'Clinica', 'Personal clinica', 'Tipo tratamiento', 'Datos Pacientes', 'Historial clinico', 'Citas', 'Historial citas', 'COBRANZA', and 'REPORTES'. The main area is titled 'SOLICITAR CITAS MEDICAS' and shows patient information: CARNET ID. 11112221 BN, NOMBRE FELIX, APELLIDOS MERCADO MERCADO, DIRECCION CALLE COIMBRA 3714 DOMICLIADO EN LA C/ 20 NO. 444 VILLA TUNARI DE LA CIUDAD DE EL ALTO, NOMBRE EL QUE BRINDA JUAN FERNANDO, APELLIDOS EL QUE BRINDA MALDONA FLORES, and CELULAR EL QUE BRINDA. Below this is a dental chart (odontograma) with teeth represented by icons and labeled with numbers like D18, D17, D16, D15, D14, D13, D12, D11, D21, D22, D23, D24, D25, D26, D27, D28, D35, D34, D33, D32, D31, D61, D62, D63, D64, D65, D85, D84, D83, D82, D81, D71, D72, D73, D74, D75, D48, D47, D46, D45, D44, D43, D42, D41, D31, D32, D33, D34, D35, D36, D37, D38. A 'DATOS DEL TRATAMIENTO' section includes fields for 'ESTADO' (DIENTE INTACTO -> 0.00), '# DIENTE TRATADO', 'CARA TRATADA', and 'DESCUENTO'. A 'DESCRIPCION HISTORIAL:' section has fields for 'FECHA REG.' (21/11/2023), 'HORA' (23:52), and 'SUBJETIVO'. Below this is a 'Guardar Tratamiento' table with 10 rows of treatment records. A 'Paraph' text editor is also visible.

#	MONTO	DESCUENTO	MONTO A PAGAR	TIPO TRATAMIENTO	NRO DIENTE	CARA TRATADA	DESCRIPCION	FECHA	
1	70.00	5.00	65	FRACTURA	D47	C5	NINGUNO	2023-11-08	Eliminar
2	50.00	5.00	45	CARIES	D46	C4	NINGUNO	2023-11-08	Eliminar
3	50.00	5.00	45	CARIES	D41	C5	CARIES SUPERFICIAL	2023-11-08	Eliminar
4	70.00	0.00	70	DIENTE POR EXTRAER	D44	C3	NINGUNO	2023-11-12	Eliminar
5	70.00	0.00	70	DIENTE AUSENTE	D14	C5	NINIGUNO	2023-11-12	Eliminar
6	50.00	0.00	50	CARIES	D18	C1	AAA	2023-11-13	Eliminar
7	70.00	0.00	70	DIENTE AUSENTE	D12	C5	4444	2023-11-13	Eliminar
8	70.00	0.00	70	DIENTE POR EXTRAER	D26	C5	ZSDASDAS	2023-11-13	Eliminar
9	70.00	20.00	50	AMALGAMA	D38	C3	SOLO UNA CARA	2023-11-14	Eliminar
10	70.00	10.00	60	AMALGAMA	D31	C1	TAPADO DE CARA 1	2023-11-14	Eliminar

#	FECHA / HORA	SUBJETIVO	OBJETIVO	ANALISIS	
1	2023-11-08 23:17	ninguno	ninguno	ninguno	Editar
2	2023-11-08 11:20	NINGUNO	NINGUNO	NINGUNO	Editar
3	2023-11-13 21:51	ninguno	ninguno786786	ninguno	Editar
4	2023-11-14 18:01	SOLO UNA CARA	SOLO UNA CARA	CURACION	Editar
5	2023-11-14 19:20	AMALGAMA	AMALGAMA	AMALGAMA	Editar

El presente módulo nos permite mostrar de manera gráfica los tratamientos que se realizó al paciente, en que pieza dental y cara se realizó el mismo.

Figura 31

Diseño de interfaz para realizar pagos



El presente módulo nos permite realizar pagos de los tratamientos dentales, previa búsqueda de los usuarios.

Figura 32

Diseño de interfaz para la administración de tratamientos.



El presente módulo nos permite administrar los tratamientos que se realizan en las clínicas en caso de realizar nuevos tratamientos se los puede adicionar.

Figura 33

Reporte de historial clínico-odontograma



HISTORIAL CLINICA ODONTOLOGICA

DATOS DEL PACIENTE :

MERCADO	MERCADO	FELIX	33	MASCULINO
APELLIDO PATERNO	APELLIDO MATERNO	NOMBRE	EDAD	SEXO
SANTIGO-LARECAJA	NINGUNO	CALLE COIMBRA 3714 DOMICILIADO EN LA C/ 20 NO. 444 VILLA...	74524545	
LUGAR Y FECHA DE NACIMIENTO	OCUPACION	DIRECCION	CELULAR	
T	SOLTERO	NINGUNO	NINGUNO	
GRADO DE INSTITUCION	ESTADO CIVIL	NACIONES ORIGINARIAS	IDIOMA O DIALECTO	

PERSONA QUE BRINDA LA INFORMACION :

MALDONA	FLORES	JUAN FERNANDO	CALLE COIMBRA 3714 DOMICILIADO EN ...	
APELLIDO PATERNO	APELLIDO MATERNO	NOMBRE	DIRECCION	CELULAR

ANTECEDENTES PATOLOGICOS FAMILIARES : NINGUNO

ANTECEDENTES PATOLOGICOS DEL PACIENTE :

Anemia ()	Cardiopatias ()	Asma (SI)	Diabetes Mel. ()	Enf. Dastricas ()
Epilepsia ()	Hepatitis ()	Hipertension ()	Tuberculosis (SI)	VIH ()
OTROS (NINGUNO)		ALERGIAS (NO)	EMBARAZO (NO)	EMBARAZO SEMANAS ()
¿ESTA EN TRATAMIENTO MEDICO? (NINGUNO)		¿ACTUALMENTE RECIBE ALGUN MEDICAMENTO? (NO)		
TUVO HEMORRAGIA DESPUES DE UNA EXTRACCION DENTAL (NO)		ESPECIFIQUE : INMEDIATA - MEDIATA (NO)		

EXAMEN EXTRA ORAL :

ATM (NINGUNO)	GANGLIOS LINFATICOS (NINGUNO)
RESPIRACION (NASAL)	OTROS (NINGUNO)

EXAMEN INTRA ORAL :

LABIOS (NINGUNO)	LENQUA ()
PALADAR ()	PISO DE LA BOCA ()
MUCOSA YUGAL ()	ENCIAS ()
	UTILIZA PROTESIS DENTAL (NO)

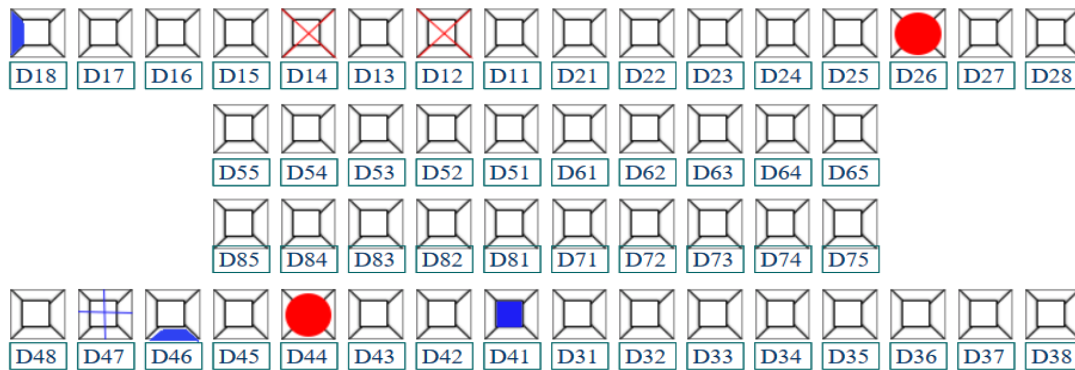
ANTECEDENTES BUCODENTALES :

FECHA DE LA ULTIMA VISITA AL ODONTOLOGO (22 de octubre de 2023)	HABITOS (OTROS)	OTROS (NINGUNO)
---	-----------------	-----------------

ANTECEDENTES DE HIGIENE ORAL :

UTILIZA CEPILLO DENTAL (SI)	UTILIZA HILO DENTAL (NO)	UTILIZA ENJUAGUE BUCAL (SI)
FRECUENCIA DEL CEPILLADO DENTAL (NINGUNO)	DURANTE EL CEPILLADO DENTAL LE SANGRAN LAS ENCIAS (NO)	HIGIENE BUCAL (BUENA)

ODONTOGRAMA :



	DIENTE AUSENTE
	FRACTURA
	CARIES
	ENDODONCIA
	DIENTE POR EXTRAER

ESTADO FERIODONTAL

0	1	2	3	4	X	T.	P.	CRITERIOS DE DIAGNOSTICO
sano	saagrado	calculos o tartaros	bolsa de 4 a 5 mm	bolsa de 6 mm o mas	sextante excluido	A	0	SANO
						B	1	CARIADO
						C	2	OBTURADO Y CARIADO
						D	3	OBTURADO SIN CARIES
						E	4	PERDIDO POR CARIES
						-	5	PERDIDO POR OTRA RAZON
						F	6	SOPORTE DE FUENTE CORONO ESPECIAL O FUNDA
						-	7	NO ERUPCIONADO
						G	8	TRAUMATISMOS
						H	9	NO REGISTRADO

HISTORIAL DE TRATAMIENTO

SUBJETIVO :	ninguno
OBJETIVO :	ninguno786786
ANALISIS :	ninguno
PLAN DE ACCION :	<p>Los países pueden nombrar a las especialidades médicas y odontológicas de distinta manera, aunque cumplan la misma función tienen nombres diferentes. También puede que tengan otras especialidades.</p> <p>Cada experto te ayudará a resolver problemas específicos. Las especialidades de odontología en España son nueve; son bastante reconocidas:</p> <ol style="list-style-type: none"> 1. Odontología general. 2. Odontopediatría. 3. Endodoncia. 4. Ortodoncia. 5. Prostodoncia. 6. Periodoncia.

MERCADO MERCADO FELIX PACIENTE		SELLO DEL ODONTOLOGO/A											
FECHA:	14/11/2023	HORA:	18:01	EDAD:	33	P.A.		F.C.		F.R.		Temp.	
SUBJETIVO :	SOLO UNA CARA												
OBJETIVO :	SOLO UNA CARA												
ANALISIS :	CURACION												
PLAN DE ACCION :	SOLO UNA CARA EL TRATAMIENTO SOLO IMPLICA UNA CARA												

MERCADO MERCADO FELIX PACIENTE		SELLO DEL ODONTOLOGO/A											
FECHA:	14/11/2023	HORA:	19:20	EDAD:	33	P.A.		F.C.		F.R.		Temp.	
SUBJETIVO :	AMALGAMA												
OBJETIVO :	AMALGAMA												
ANALISIS :	AMALGAMA												
PLAN DE ACCION :	AMALGAMA												

MERCADO MERCADO FELIX
PACIENTE

SELLO DEL ODONTOLOGO/A

Figura 33

Reporte egresos e ingresos de la clinica



REPORTES INGRESOS EGRESOS

FECHA :1 de septiembre de 2023 al 10 de noviembre de 2023

#	CODIGO	CONCEPTO PAGO	TIPO	FECHA	SUCURSAL	INGRESOS	EGRESO
1		10 PARACETAMOLE 800G	INGRESOS	2023-09-23	EX PARADO 8	10.00	0
2		20 PARACETAMOL DE 500G	INGRESOS	2023-09-24	EX PARADO 8	20.00	0
3	31029125112	PAGO MES ENERO	EGRESOS	2023-10-29	EX PARADO 8	0	400.00
4	C000010	POR CONCEPTO DE PAGO DE SERVICIO ODONTOLÓGICO ZONADENTAL AL PACIENTE JUAN CARLOS ALARCON PEREZ CON CARNET DE ID. 9212193 LP	INGRESOS	2023-11-08	EX PARADO 8	20.00	0
5	C000011	POR CONCEPTO DE PAGO DE SERVICIO ODONTOLÓGICO ZONADENTAL AL PACIENTE JUAN CARLOS ALARCON PEREZ CON CARNET DE ID. 9212193 LP	INGRESOS	2023-11-08	EX PARADO 8	10.00	0
6	C000012	POR CONCEPTO DE PAGO DE SERVICIO ODONTOLÓGICO ZONADENTAL AL PACIENTE JUAN CARLOS ALARCON PEREZ CON CARNET DE ID. 9212193 LP	INGRESOS	2023-11-08	EX PARADO 8	10.00	0
7	C000013	POR CONCEPTO DE PAGO DE SERVICIO ODONTOLÓGICO ZONADENTAL AL PACIENTE FELIX MERCADO MERCADO CON CARNET DE ID. 11112221 BN	INGRESOS	2023-11-08	EX PARADO 8	20.00	0
8	3110811011	PAGO MES SEPTIEMBRE	INGRESOS	2023-11-08	EX PARADO 8	400.00	0
9	31108112754	CONSULTA	INGRESOS	2023-11-08	EX PARADO 8	50.00	0
10	3110811139	PAGO MES NOV. ALQUIR	EGRESOS	2023-11-08	EX PARADO 8	0	1000.00
11	31108112846	ALQUILER	EGRESOS	2023-11-08	EX PARADO 8	0	100.00
12	3110993446	PAGO MES SEPTIEMBRE	EGRESOS	2023-11-09	EX PARADO 8	0	100.00
TOTAL:						540	1600

CAPÍTULO IV

METRICAS DE CALIDAD

COSTO Y SEGURIDAD

CAPÍTULO IV

4 MÉTRICAS DE CALIDAD, COSTO Y SEGURIDAD

4.1 INTRODUCCIÓN

En este presente capítulo se realiza el análisis y la obtención de las métricas de calidad, costo y seguridad del sistema de administración y seguimiento de historiales clínicos odontológicos multisucursal para el Centro Odontológico Zonadental, utilizando los conceptos teóricos mencionados en el capítulo dos (marco teórico).

4.2 METRICAS DE CALIDAD

Se realizará la métrica de calidad de software mediante la norma ISO 9126 con las características que son: la funcionalidad, confiabilidad, usabilidad, eficiencia, mantenibilidad y portabilidad.

4.2.1 Funcionalidad

La funcionalidad no se puede medir directamente por esta razón corresponde derivar medidas directas como es el punto función que cuantifica el tamaño y la complejidad del sistema en términos de las funciones del usuario. Determina las cinco características del dominio de información, teniendo en cuenta su cantidad y se definen de la siguiente forma:

- Número de Entradas de Usuario
- Número de Salidas de Usuario
- Numero de petición de Usuario
- Numero de archivos
- Numero de Interfaces externas

Número de Entradas de usuario, se cuenta cada entrada de usuario que proporciona diferentes datos orientados a la aplicación. Las entradas se deberían diferenciar de las peticiones, las cuales se cuentan de forma separada.

Tabla 11*Número de Entradas de Usuario*

Entradas de Usuario		
1	Administración de usuarios	5
2	Administración de privilegios	6
3	Administración de sucursales	5
4	Administración de personal clínica	5
5	Administración de tipos tratamiento	5
6	Administración de pacientes	8
7	Administración de citas	3
8	Administración de realizar pagos	3
9	Administración de concepto de pagos	4
10	Administración de otros ingresos	5
11	Administración de egresos	5
12	Reporte de ingresos y egresos	4
Total		58

Número de Salidas Usuario, se cuenta cada salida que proporciona al usuario información orientada a la aplicación. En este contexto la salida se refiere a informes, pantallas, mensajes de error, etc. Los elementos de datos particulares dentro de un informe no se cuentan de forma separada.

Tabla 12*Número de Salidas Usuario*

Salidas de Usuario		
1	Administración de usuarios	3
2	Administración de privilegios	2
3	Administración de sucursales	2
4	Administración de personal clínica	4
5	Administración de tipos tratamiento	4

6	Administración de pacientes	3
7	Administración de citas	4
8	Administración de realizar pagos	3
9	Administración de concepto de pagos	2
10	Administración de otros ingresos	4
11	Administración de egresos	4
12	Reporte de ingresos y egresos	3
Total		38

Número de Peticiones 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 separada.

Tabla 13

Número de Peticiones del Usuario

Peticiones de Usuario		
1	Administración de usuarios	5
2	Administración de privilegios	3
3	Administración de sucursales	3
4	Administración de personal clínica	5
5	Administración de tipos tratamiento	3
6	Administración de pacientes	5
7	Administración de citas	3
8	Administración de realizar pagos	3
9	Administración de concepto de pagos	3
10	Administración de otros ingresos	4
11	Administración de egresos	4
12	Reporte de ingresos y egresos	3
Total		44

Número de Archivos, se Cuenta archivo maestro Lógico

Tabla 14*Numero de Archivos*

Archivo		
1	Administración de usuarios	5
2	Administración de privilegios	4
3	Administración de sucursales	4
4	Administración de personal clínica	5
5	Administración de tipos tratamiento	3
6	Administración de pacientes	6
7	Administración de citas	3
8	Administración de realizar pagos	4
9	Administración de concepto de pagos	4
10	Administración de otros ingresos	4
11	Administración de egresos	4
12	Reporte de ingresos y egresos	3
Total		49

Numero de interfaces Externas, se cuenta todas las interfaces legibles por la máquina.

Tabla 15*Numero de Interfaces Externas*

Archivo		
1	Internet	1
2	Intranet	1
Total		2

Para realizar el cálculo de la cuenta Total con factores de ponderación se debe realizar con la siguiente Tabla.

Tabla 16*Factores de Ponderación*

	Parámetros de medida	Cuenta	Factores de Ponderación			Total
			Simple	Medio	Complejo	
1	Nro. de Entradas de Usuario	58	3	4	6	232
2	Nro. de Salidas de Usuario	38	4	5	7	190
3	Nro. de Peticiones de Usuario	44	3	4	6	176
4	Nro. de Archivos	49	7	10	15	490
5	Nro. de Interfaces Externas	2	5	7	10	14
Total						1102

Valores de ajuste de complejidad según las respuestas a las siguientes preguntas que se muestra en la siguiente tabla.

Tabla 17*Valores de Ajuste de Complejidad*

Nro	Factor de Complejidad	Sin Influencia	Incidental	Moderada	Medio	Significativa	Esencial	Fi
		0	1	2	3	4	5	
1	¿Requiere el sistema copia de seguridad y recuperación?						X	5
2	¿Requiere comunicación de datos?						X	5
3	¿Existen funciones de procesos distribuidos?				X			3
4	¿El rendimiento es critico					X		4
5	¿Sera ejecutado el sistema en entorno existente y fuertemente utilizado?						X	5
6	¿Entrada de datos EN LINEA?						X	5

7	¿Requiere la entrada de datos interactiva que las transiciones de entrada se llevan a cabo sobre múltiples pantallas o variadas opciones?	X	5
8	¿Se actualizan los archivos maestros de forma interactiva?	X	5
9	¿Son complejas de las entradas de salidas de archivos?	X	3
10	¿Lógica del proceso Interno Compleja?	X	3
11	¿Se diseña el código para ser reutilizable?	X	5
12	¿Están incluidas en el diseño conversiones de instalación?	X	2
13	¿instalaciones Múltiples?	X	2
14	¿Facilidad de Cambios?	X	5
Factor de Complejidad Total (FCT)			57

Para calcular los puntos función (PF), utilizaremos la relación siguiente

$$PF = \text{Cuenta Total} * (0.65 + 0.01 * \sum Fi)$$

Dónde:

Cuenta Total: Nivel de complejidad del sistema con respecto al usuario

$(0.65 + 0.01 * \sum Fi)$: Ajuste de complejidad según el dominio de la información.

0.01: Factor de conversión, es decir un error de 1%

0.65: Valor mínimo de ajuste

Calculando el punto función de según la ecuación:

$$PF = 1102 * [0.65 + 0.01 * 57]$$

$$PF = 1344.44$$

Si calculamos al 100% el nivel de confianza consideramos la sumatoria de $Fi=70$ como el máximo valor de ajuste de complejidad se tiene:

$$PF \max = \text{Cuenta Total} * [1 + 0.01 * \sum F_i]$$

$$PF \max = 1102 * [0,65 + 0.01 * 70]$$

$$PF = 1487.7$$

Con los máximos valores de ajuste de complejidad se tiene que la funcionalidad real es:

$$\text{Funcionalidad} = \frac{1344.44}{1487.7} = 0,90$$

$$\text{Funcionalidad} = 0,90 * 100 = 90\%$$

Entonces la funcionalidad del sistema es un 90% esto quiere decir que el sistema tiene un 90% que funcione sin riesgos de fallo y operatividad constante y 10% de colapso de sistema.

4.2.2 Confiabilidad

La confiabilidad del sistema se define como la probabilidad de operación libre de fallos de un programa de computadora en un entorno determinado y durante un tiempo específico.

Para calcular confiabilidad del sistema se toma en cuenta el periodo de tiempo en el cual se ejecuta y se obtiene muestras

$$F(t) = f * e^{(-\mu * t)}$$

En el inicio de ejecución $t_0 = 0$ lo que significa el tiempo inicial en el cual dará inicio el funcionamiento del sistema.

$$F(0) = f * e^{(-\mu * t_0)}$$

Se observa el trabajo del sistema hasta que produce una falla en el instante T, el cual se aproxima a una variable aleatoria continua.

Como se aproxima a variables aleatorias continuas, la confiabilidad a ser obtenida en términos probabilísticos.

Entonces el término en el cual el sistema trabaja sin falla está dado por la ecuación (2) y tiempo en el cual no falla el sistema está dado por ecuación (3).

$$P(T \leq t) = F(t) \quad (2) \text{ probabilidad de fallas}$$

$$P(T \leq t) = 1 - F(t) \quad (3) \text{ Probabilidad de trabajo sin falla}$$

En un periodo de 20 días como tiempo de prueba se define de cada 10 ejecuciones 1 falla

Conociendo la funcionalidad del 90% del sistema calculamos para el periodo establecido.

$$P(T \leq t) = 1 - F(t)$$

$$F(t) = 1 - 90 + e^{(-\frac{1}{10} + 20)}$$

$$F(t) = 1 - 0,122$$

$$F(t) = 0,88$$

La confiabilidad del sistema es del 88% en un periodo de 20 días como tiempo de prueba.

4.2.3 Mantenibilidad

El mantenimiento se desarrolla para mejorar el sistema en respuesta a los nuevos requerimientos que la universidad tenga y los reglamentos que está regida por la misma.

El estándar IEE94 sugiere un índice de madurez del software (IMS) que proporciona un indicador en la estabilidad de un producto, se lo determina con la siguiente fórmula.

$$IMS = \frac{[Mt - (Fa + Fc + Fd)]}{Mt}$$

Mt= Numero de módulos la Versión Actual

Fa= Numero de Módulos en la versión actual que se han añadido

Fc: Numero de Módulos en la versión actual que se han Cambiado.

Fd: Numero de Módulos en la versión anterior que se han borrado en la versión actual.

Calculado el IMS

$$IMS = [16 - (1 + 0 + 0)] / 16$$

$$IMS = 0,94$$

Con lo que podemos decir que el nuevo sistema tiene una estabilidad de 94% que es la facilidad de mantenimiento el 6% restante es el margen de error correspondiente a los cambios y modificaciones efectuados desde el prototipo de la versión actual

Puesto que es un sistema diseñado con los requerimientos actuales con el tiempo surgirán nuevos requerimientos los cuales cambiara el valor índice de madurez del software.

Mantenimiento Correctivo, el sistema presenta diseño modular y es por eso que tolera variaciones en su corrección.

Mantenimiento Adaptativo. Se realiza cuando en la organización se produce algún cambio haciendo que el sistema sufra modificaciones.

El sistema por su programación modular permitirá fácilmente hacer modificaciones en sus módulos o integrar nuevos al sistema.

4.2.4 Usabilidad

La usabilidad es lo mismo decir facilidad de uso, esta métrica nos muestra el costo de aprender a manejar el producto, lo cual se calcula con la siguiente formula:

$$FU = [(Sum(xi)/n) * 100]$$

Tabla 18

Ajuste de Preguntas

PREGUNTAS	Respuestas		Ponderación %
	SI	NO	

Puede Utilizar con facilidad el sistema?	5	1	83%
Puede Controlar operaciones que el sistema solicita?	5	1	83%
Las Respuestas del sistema son complicadas?	1	5	83%
El Sistema permitió la retroalimentación de información?	6	0	100%
El sistema cuenta con interface agradable a la vista?	6	0	100%
La respuesta del sistema es satisfactoria?	5	1	83%
Le parece complicada las funciones del sistema?	1	5	83%
Se hace difícil o dificultoso aprender a manejar el sistema?	1	5	83%
Los resultados que proporciona el sistema facilitan el trabajo?	6	0	100%
Durante el uso del sistema se produjo errores?	1	5	83%
USABILIDAD			88%

Existe un 88% de comprensión o entendimiento de los usuarios con respecto a la capacidad del sistema.

4.2.5 Portabilidad

Para la portabilidad del sistema se tomará en cuenta dos aspectos como ser a nivel aplicación, nivel hardware la portabilidad la dividimos en dos secciones portabilidad del lado del servidor y portabilidad del lado del cliente.

Portabilidad lado servidor

A nivel sistema de software El sistema de administración y seguimiento de historiales clínicos odontológicos multisucursal para el Centro Odontológico Zonadental es portable bajo los siguientes sistemas operativos de la familia Microsoft Windows 2000, Windows milenio, Windows XP, Windows Vista, Windows 7, Windows 8, Windows 10, Windows 11 y todo el sistema operativo libre de Linux.

A nivel de base de datos se utiliza base de datos creada en Mariadb, la portabilidad se muestra que la base de datos que la base de datos puede ser migrada SQL server, Oracle y PostgreSQL.

A nivel Hardware el sistema de administración y seguimiento de historiales clínicos odontológicos multisucursal para el Centro Odontológico Zonadental es portable bajo

las siguientes características mínimas de hardware microprocesador Pentium IV RAM de 256 Mb Como mínimo espacio en el disco duro 1Gb como mínimo, monitor SVGA.

4.3 ESTIMACIÓN DE COSTO DEL SOFTWARE

Existen distintos métodos para la estimación de costos del desarrollo de un software, estos costos no son más que establecer una relación matemática entre el esfuerzo y tiempo requeridos para dicho desarrollo.

4.3.1 Método de estimación COCOMO II

El método de estimación de costos COCOMO II considera como principal variable las KLDC (Kilo Líneas de código) como observaremos a continuación.

El sistema de web para el seguimiento y control de afiliados cuenta con 7595 líneas de código en el lenguaje PHP.

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

$$KLDC = \frac{7595}{1000}$$

$$KLDC = 7.595$$

Donde establecemos que el valor de KLDC es igual a 7.595, ya que en nuestro caso el tipo de proyecto de software será de orgánico ya que el número de líneas de código no supera las 50 KLDC, y además el proyecto no es muy complejo por esta razón los coeficientes que usaremos serán.

Dando como resultado un modelo Intermedio siendo así un sistema orgánico

Tabla

Tabla 19

Coefficientes del Modelo COCOMO II

Proyecto de Software	a	B	c	d
Orgánico	2.40	1.05	2.50	0.38
Semiacoplado	3.00	1.12	2.50	0.35

Empotrado	3.60	1.20	2.50	0.32
-----------	------	------	------	------

Para hallar los valores de FAE, se utilizará la tabla de atributos multiplicadores

4.3.2 Justificación de los valores

4.3.2.1 Atributos de software

- **Fiabilidad:** Requerida del software, si se produce un fallo en el sistema Web, puede ocasionar, que se lleve más tiempo en analizar la información, en lo requerimientos solicitados, etc... Puede ocasionar retraso en las funciones que desempeña la institución. (Valoración nominal).
- **Tamaño de la base de datos:** La base de datos de nuestro producto cumple con lo solicitado, requerido para que el sistema funcione de mejor manera. (Valoración Nominal).
- **Complejidad del producto:** El sistema Web no realizara cálculos complejos por esa razón su (Valoración Baja).

4.3.2.2 Atributos de hardware

- **Restricciones del tiempo de ejecución:** En los requerimientos se exige alto rendimiento para el buen funcionamiento del sistema Web (Valoración Alta).
- **Restricciones del almacenamiento principal:** No hay restricciones al respecto al almacenamiento de información por esa razón su (Valoración Nominal).
- **Volatilidad de la máquina virtual:** Se usarán sistemas de la “Familia Linux y Windows” ya que proporciona mayor seguridad (Valoración Nominal).
- **Tiempo de respuesta del ordenador:** Deberá ser interactivo con el usuario dependiendo a las peticiones realizada por el usuario (Valoración Alta).

4.3.2.3 Atributos del personal

- **Capacidad del analista:** Capacidad alta debido a la experiencia en análisis en proyecto similar (Valoración Alta)
- **Experiencia en la aplicación:** Se tiene muy alta experiencia en aplicaciones de esta similitud (Valoración muy alta).

- Capacidad de los programadores: Teóricamente deberá tenerse una capacidad muy alta por la experiencia en anteriores proyectos similares (Valoración muy alta).
- Experiencia en la maquinas virtual: Con Windows Professional la experiencia es a nivel usuario (Valoración Nominal).
- Experiencia en el lenguaje: Es relativamente alta, dada experiencia en otras similares aplicaciones (Valoración Alta).

4.3.2.4 Atributos del proyecto

- Técnicas actualizadas de programación: Se usarán prácticas de programación mayormente tónicas de programación actualizadas (Valoración alta).
- Utilización de herramientas software: Se usarán herramientas estándar, de las cuales se tiene cierta experiencia en su manejo (Valoración Alta).
- Restricciones de tiempo de desarrollo: Existen pocos límites de planificación del desarrollo lo cual establece una valoración (Valoración nominal).

Tabla 20

Cálculo de Atributos FAE de valores de coste COCOMO II

Atributos	Valor					
	Muy bajo	Bajo	Nominal	Alto	Muy alto	Extra alto
Atributos de software						
Fiabilidad	0,75	0,88	1,00	1,15	1,40	
Tamaño de Base de datos		0,94	1,00	1,08	1,16	
Complejidad	0,70	0,85	1,00	1,15	1,30	1,65
Atributos de hardware						
Restricciones de tiempo de ejecución			1,00	1,11	1,30	1,66
Restricciones de memoria virtual			1,00	1,06	1,21	1,56
Volatilidad de la máquina virtual		0,87	1,00	1,15	1,30	

Tiempo de respuesta	0,87	1,00	1,07	1,15
Atributos de personal				
Capacidad de análisis	1,46	1,19	1,00	0,86
Experiencia en la aplicación	1,29	1,13	1,00	0,91
Calidad de los programadores	1,42	1,17	1,00	0,86
Experiencia en la máquina virtual	1,21	1,10	1,00	0,90
Experiencia en el lenguaje	1,14	1,07	1,00	0,95
Atributos del proyecto				
Técnicas actualizadas de programación	1,24	1,10	1,00	0,91
Utilización de herramientas de software	1,24	1,10	1,00	0,91
Restricciones de tiempo de desarrollo	1,22	1,08	1,00	1,04
Total			0,68	

Nota: esta tabla muestra los valores asignados según el sistema realizado.

Por tanto, nuestro Factor de ajuste será

$$FAE=0,68$$

Aplicando y remplazando valores a la fórmula de esfuerzo, se tiene:

$$E = a * KLCD^b * FAE \text{ (Personas/Mes)}$$

$$E = 2,40 * 7,5^{1,05} * 0,68$$

$$E = 13,53 \text{ (Personas Mes)}$$

Cálculo del Tiempo

$$T = c * Esfuerzo^d \text{ (Meses)}$$

$$T = 2,50 * 13,53^{0,38}$$

$$T = 6,7 \text{ (Meses)}$$

Cálculo de la Productividad

$$PR = \frac{LCD}{Esfuerzo}(\text{Meses})$$

$$PR = \frac{7595}{13.53}(\text{Meses})$$

$$PR = 561.34 \text{ (LCD/personas Mes)}$$

Cálculo del personal requerido

$$P = \frac{E}{T}(\text{Personas})$$

$$P = \frac{13.53}{6.7}(\text{Personas})$$

$$P = 2.01(\text{Personas}) \text{ equivalente a 2 personas}$$

Costo Total del Proyecto

(Coste Mes) = P* Salario medio entre los programadores y analistas

Costo Persona Mes 340\$

$$\text{Coste Mes} = 2 * 340 = 680 \$us$$

$$\text{Costo Total} = 680 * 7 = 4760 \$us$$

En resumen, se requiere 2 personas estimando un trabajo de 7 meses y con costo total de 4760\$us Equivalente en bolivianos a 33,320 Bs.

4.4 SEGURIDAD

4.4.1 Seguridad física

Para la seguridad es importante el acceso físico a la información, es uno de los aspectos más olvidados a la hora del diseño de un sistema informático. Si bien algunos de los aspectos tratados a continuación se prevén, otros pueden intentar acceder físicamente a la sala donde opera el sistema web.

- Seguridad física en el entorno donde se aloja el sistema web.
- Se recomienda los back-up 2 a 3 copias en distintos discos de almacenamiento externo.

- Las copias de la base de datos (BD) deberá ser protegido por personal autorizado tenga el acceso al sistema web.

4.4.2 Seguridad lógica

La seguridad lógica se establecen procedimientos que resguarden el acceso a la información y solo se permita acceder a las personas autorizadas, entonces se debe:

- Los respaldos (Back-up) de la base de datos (BD) debe ser establecidos periódicamente cada semana y también respaldos manuales que solo pueda realizar el administrador del Sistema Web.
- Se aplica el paradigma de Modelo Vista Controlador para tener mayor seguridad en el código
- Los personales involucrados en el proceso del sistema deberán cambiar el Password del sistema periódicamente una vez al mes como mínimo para proporcionar seguridad a cada una de sus cuentas.
- Manejo de roles y tipos usuario en el gestor de Base de Datos
- El ingreso al sistema operativo donde se aloja el sistema web debe ser bajo un usuario y contraseña que solo tendrá el administrador.
- El ingreso al sistema implementado deberá ser con usuario y contraseña

4.5 PRUEBAS DE CAJA NEGRA Y BLANCA

4.5.1 Pruebas de Caja Negra

En la evaluación fundamenta que el sistema se observa aspectos de funcionalidad, operativa en su proceso que realiza.

Se procede a realizar una evaluación de acuerdo a los módulos para cotejar los resultados que entrega el sistema ya sea mediante reportes o interfaces de resultados.

Registro de Usuario

Datos de Ingreso: Se ingresa un usuario y una contraseña asignada por el administrador del sistema en momento otorgado los accesos.

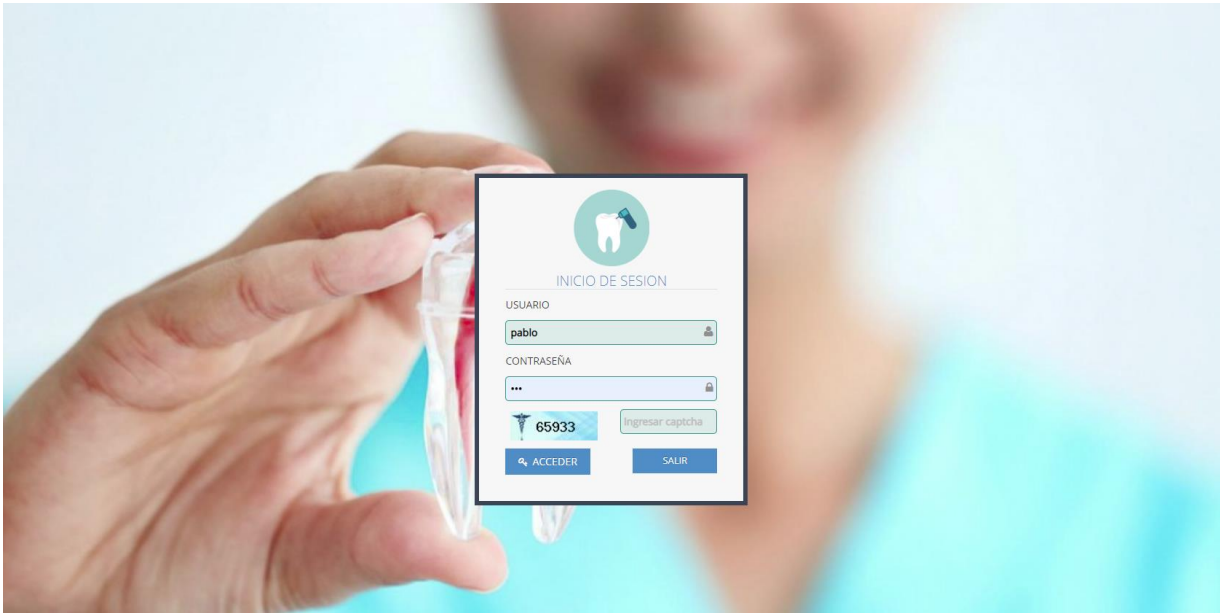
Prueba: Se espera que el sistema realice las respectivas validaciones para el ingreso de usuarios.

Resultado: El sistema no permite el ingreso al personal no autorizado por el administrador del sistema.

Inicio de sesión al sistema

Figura 30

Inicio de sesión al sistema



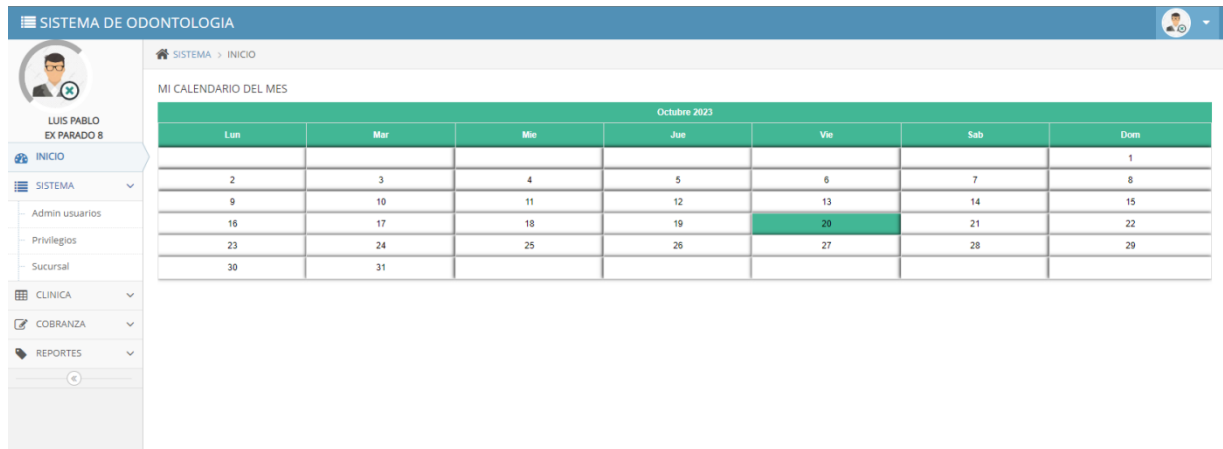
Datos de Ingreso: Es ingresa al módulo de inicio.

Prueba: Se espera que el sistema despliegue la información del sistema.

Resultado: El sistema despliega enlaces de todo el menú correspondiente a su privilegio o rol asignado.

Figura 31

Interfaz de inicio del sistema



4.5.2 Prueba de Caja Blanca

Para las pruebas de caja blanca se proporcionará una medición cuantitativa de la complejidad lógica del sistema. La Complejidad ciclo matica de un grado de flujo

$V(G)$ establece el número de caminos independientes:

- El número de regiones del gráfico de flujo.
- $V(G) = A - N + 2$, donde A es el número de aristas y N es el número de nodos.
- $V(G) = P + 1$, donde P es el número de nodos predicado.

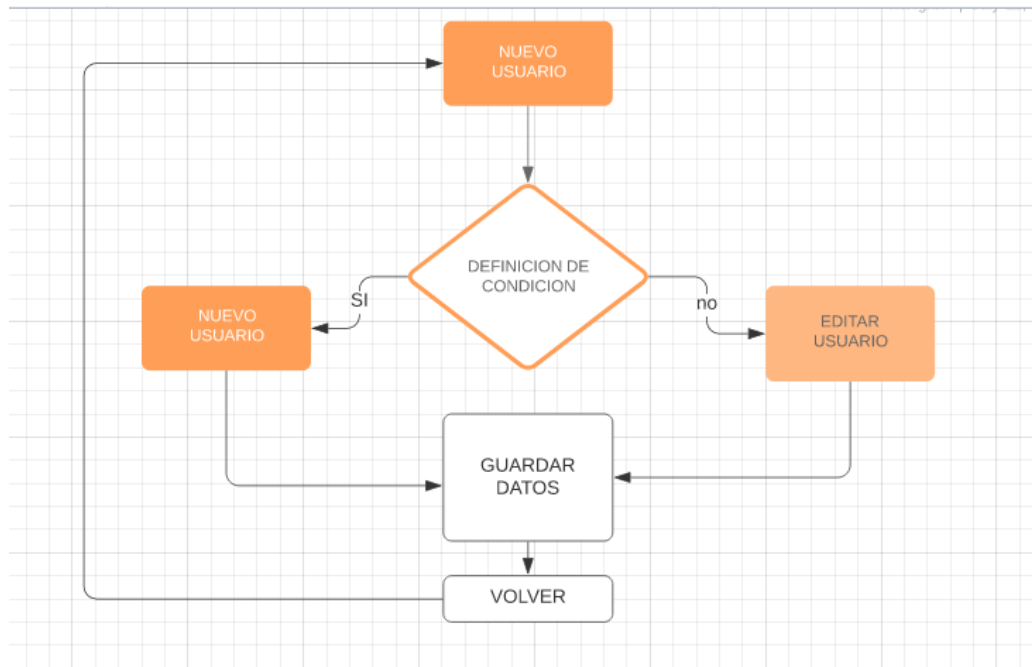
Módulo de Usuario

Se vio necesario realizar pruebas de caja blanca para el módulo de administración de usuario del sistema debido al flujo grama que presentaba.

Pruebas de Caja blanca para el registro de usuario en el sistema.

Figura 32

Creación de nuevo usuario flujo drama.



Para ilustrar el flujo grama se puede evidenciar el procedimiento en la figura anterior del módulo de administración de usuario.

Figura 33

Formulario de creación de usuario

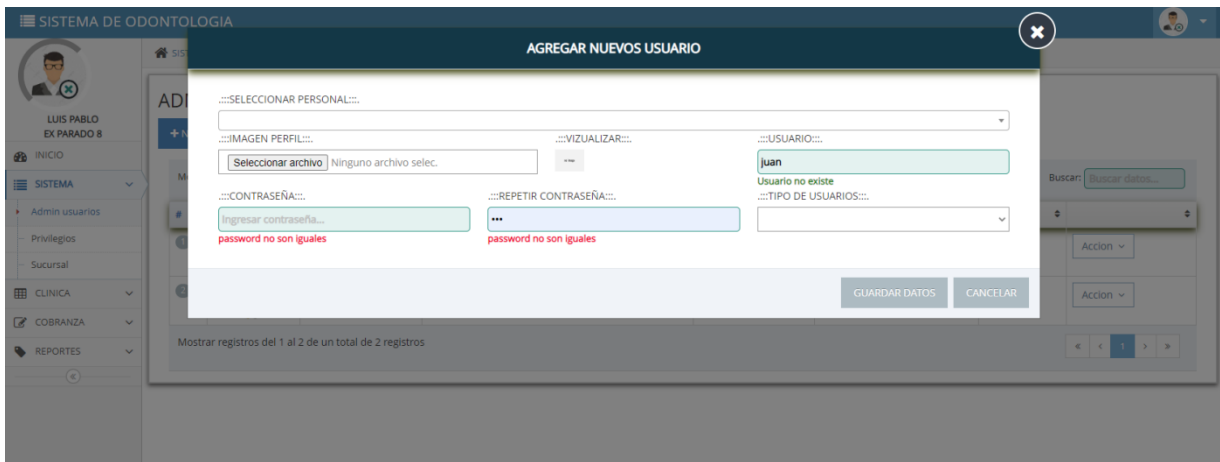
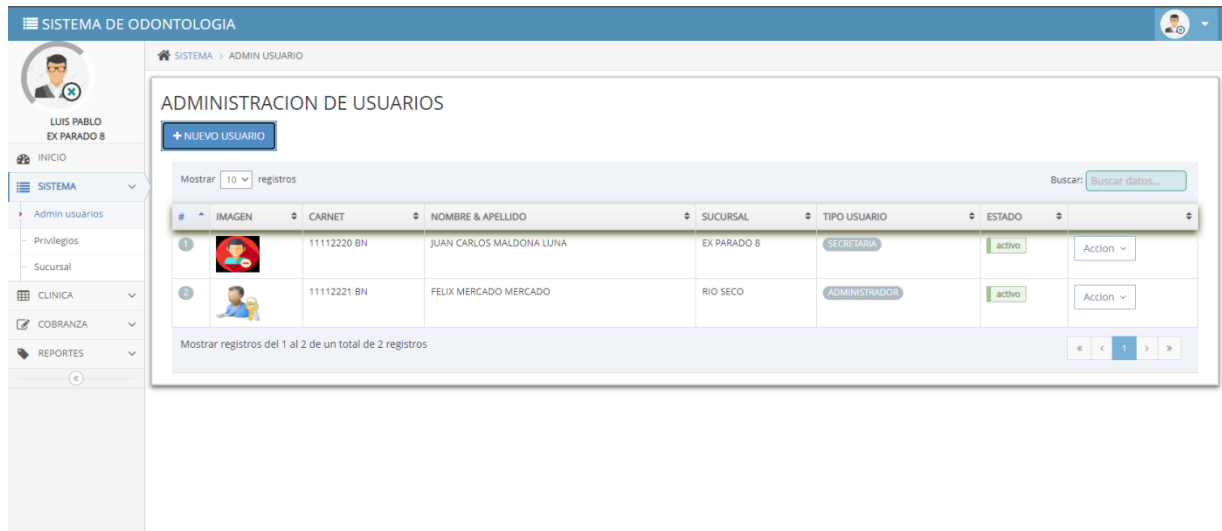


Figura 34

Listar usuario



La figura 5.9 Muestra la complejidad ciclo matica de un módulo de administración de usuario por lo que reemplazando el número de predicados en la segunda fórmula se obtiene:

$$V(G) = 1 + 1 = 2$$

Este valor determina dos caminos independientes, dos casos de prueba, para el módulo de administración de usuario. Por lo tanto, se realizará las pruebas necesarias para cada uno de los caminos como se muestra en la tabla.

Tabla 21

Valor de determina dos casos de prueba

CAMINO 1	CAMINO 2
<ul style="list-style-type: none"> • Ingresar al módulo de administración de usuario. • Vuelve a ingresar al botón de nuevo 	<ul style="list-style-type: none"> • Ingresar datos en el formulario de administración de usuario. • Verifica y valida si existen usuario activo incorrecto en el formulario. • Guarda información del usuario en el sistema.

usuario para el nuevo registro.

- Listar todos los usuarios.

4.5.3 Pruebas de estrés

Para realizar las pruebas de estrés al software realizaremos la aplicación de la herramienta JMETER que nos permite la carga de solicitudes al sistema para ver su capacidad de respuesta.

Se configuro todos los datos para la carga del sistema seguidamente se probó con un usuario en un segundo.

Figura 35

Software Jmeter

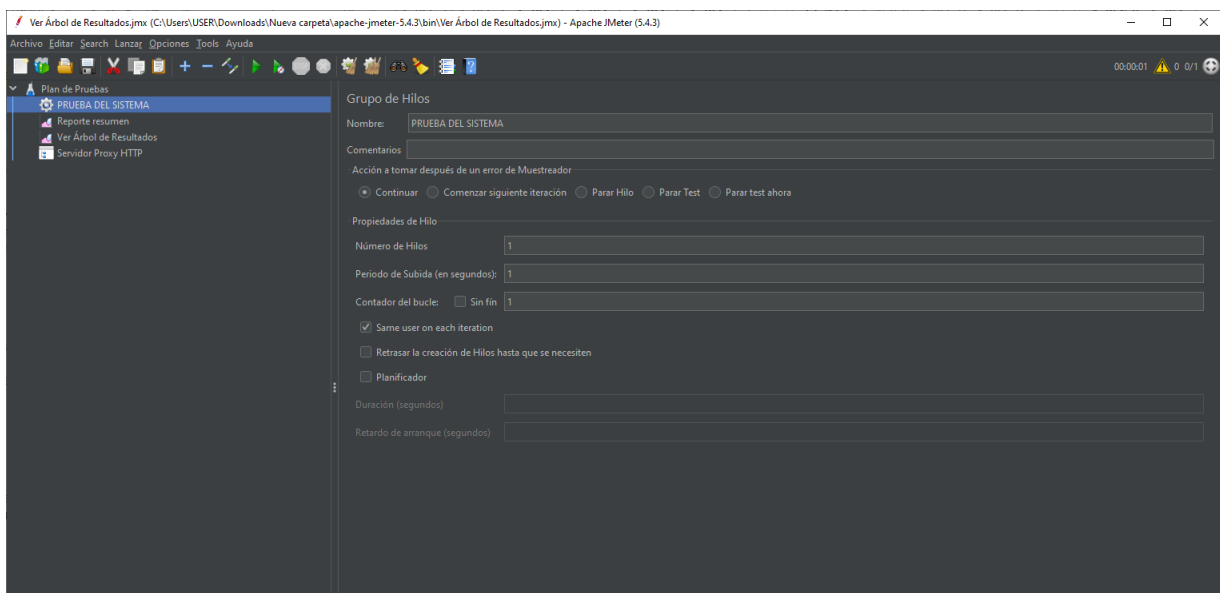


Figura 36

Ejecución del programa

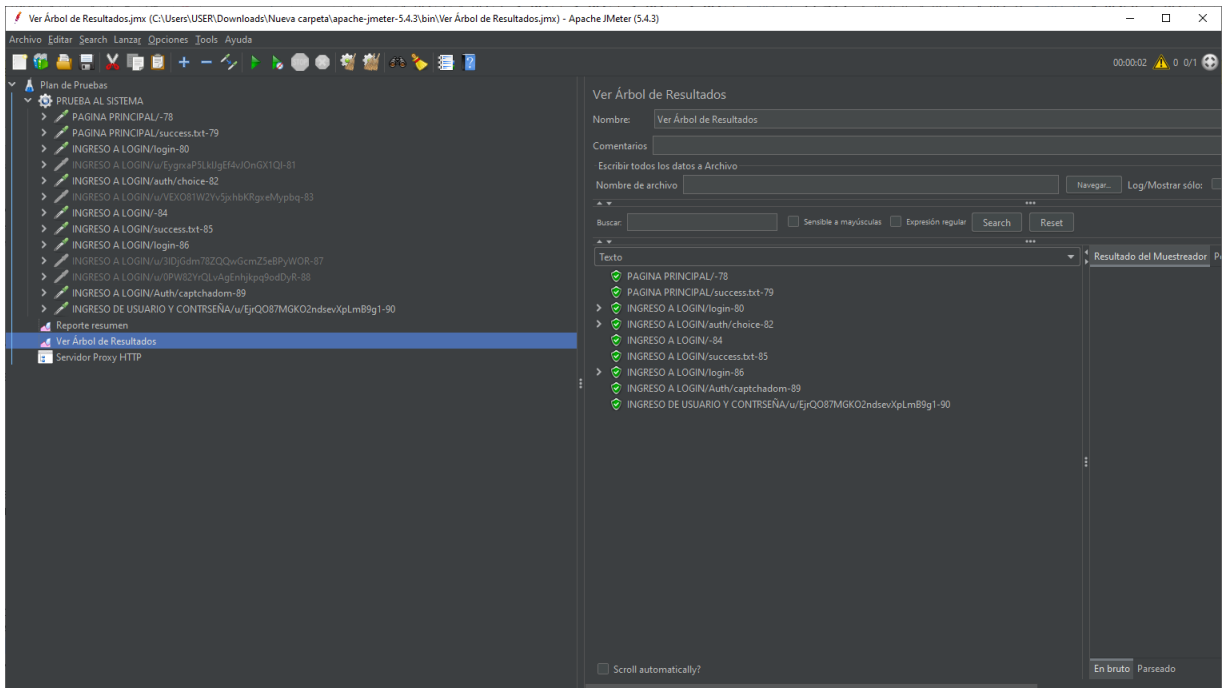


Figura 37

Resultados de las pruebas

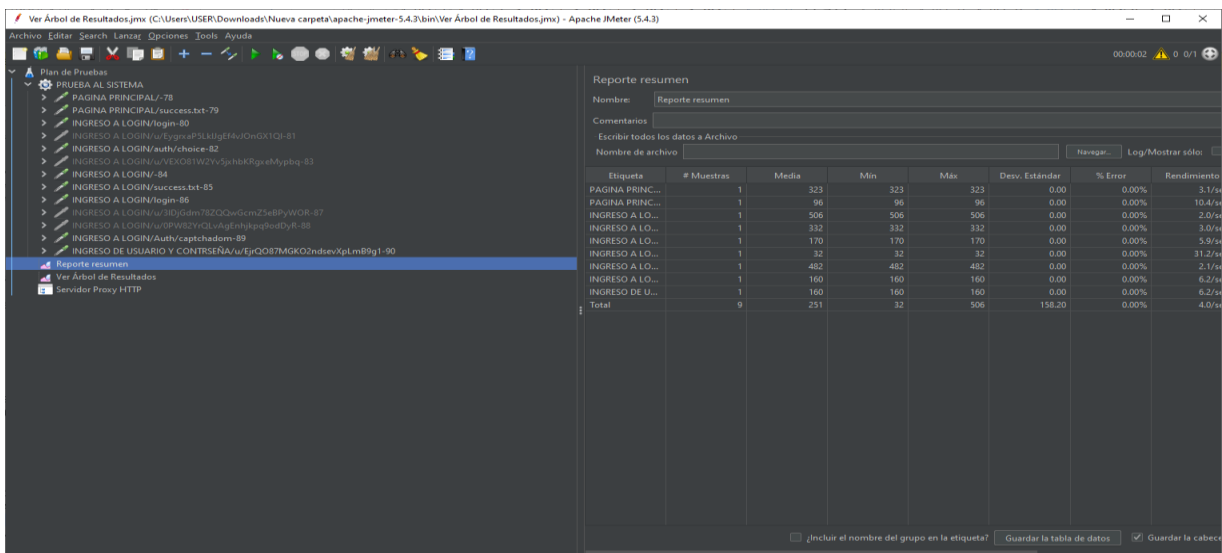


Tabla 22*Resultados Prueba de estrés*

CANTIDAD DE SOLICITUDES	TIEMPO	PORCENTAJE DE ERROR
1 usuarios	1 Segundo	0.00%
100 usuarios	1 Segundo	12.22%
200 usuarios	1 Segundo	26.32%
500 usuarios	1 Segundo	76.98%

CAPITULO V

CONCLUSIONES Y

RECOMENDACIONES

CAPÍTULO V

5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

Con base en el análisis del caso del Centro Odontológico ZonaDental y su implementación exitosa de un Sistema de Administración y Seguimiento de Historiales Clínicos Odontológicos, podemos extraer varias conclusiones clave:

- **Mejora en la eficiencia administrativa:** La introducción de un sistema de gestión digital de historiales clínicos ha simplificado y agilizado los procesos administrativos en el Centro Odontológico ZonaDental. Esto se traduce en una reducción de la carga de trabajo, una gestión más efectiva de citas, facturación y seguimiento de pacientes, lo que permite al personal centrarse en brindar una atención de mayor calidad.
- **Mayor precisión en la atención clínica:** La digitalización de historiales clínicos ha llevado a una mayor precisión en el diagnóstico y seguimiento de tratamientos. Los profesionales de la salud dental tienen acceso rápido y seguro a la información médica de los pacientes, lo que facilita la toma de decisiones informadas y contribuye a la mejora en los resultados clínicos.
- **Atención al paciente más personalizado:** El sistema ha permitido un seguimiento más cercano de los pacientes a lo largo del tiempo. Esto se traduce en una atención más personalizada y un mayor compromiso por parte de los pacientes en su propio cuidado dental. La comunicación efectiva y la coordinación entre el equipo de atención contribuyen a una experiencia más satisfactoria para los pacientes.
- **Seguridad y protección de datos:** La implementación de un sistema de gestión digital de historiales clínicos debe ir de la mano con sólidas medidas de seguridad y protección de datos. La integridad y la confidencialidad de la información de los pacientes son de suma importancia, y el caso de ZonaDental demuestra la importancia de implementar protocolos robustos en este sentido.

- Potencial de expansión y replicación: El éxito del Centro Odontológico ZonaDental en la implementación de este sistema demuestra su potencial para expandirse a otras clínicas y centros de salud dental. Esta experiencia puede servir como un modelo a seguir para otras instituciones interesadas en mejorar la gestión de historiales clínicos y la atención al paciente.

5.2 RECOMENDACIONES

Al igual que el avance de la tecnología los sistemas Web van en avance, en base al periodo de desarrollo de sistema de control y seguimiento de historiales clínicos odontológicos caso: Zonadental se realizan las siguientes recomendaciones.

- Se recomienda cambiar continuamente las contraseñas y usuario para la seguridad del sistema y proteger el acceso de personas ajenas al sistema Web.
- Se recomienda realizar mejoras en la usabilidad del sistema en función de los comentarios de los usuarios.
- Se debe implementar un programa de capacitación continua para el personal clínico, cubriendo aspectos clave del sistema.
- Para futuras actualizaciones del sistema se recomienda modificar y agregar algunos módulos y tablas a la base de datos.
- Se debe considerar la expansión del sistema a otras clínicas odontológicas, aprovechando la base de conocimiento adquirida en este proyecto.

Bibliografía

- Alegsa, L. (13 de Julio de 2023). *Definición de sistema*. Recuperado el 14 de Septiembre de 2023, de Definición de sistema: <https://www.alegsa.com.ar/Dic/sistema.php#gsc.tab=0>
- Alvarez, M., López, D., y Gutierrez, M. (17 de 01 de 2023). *Taller de PHP*. Obtenido de <http://www.desarrolloweb.com/manuales/6/>
- Anzil, F. (14 de octubre de 2023). *Concepto de Control*. Recuperado el 14 de Octubre de 2023, de Concepto de Control: <https://www.zonaeconomica.com/control>
- Arturo. (12 de abril de 2020). *Bootstrap: guía para principiantes de qué es, por qué y cómo usarlo*. Obtenido de Bootstrap: guía para principiantes de qué es, por qué y cómo usarlo: <https://rockcontent.com/es/blog/bootstrap/>
- cano, J. d. (17 de mayo de 2022). *codeigniter*. Obtenido de codeigniter: <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/codeigniter-framework-php-rapido-y-versatil/>
- Contraloría, G. d. (2013). *Sistema de Administración de Bienes y Servicios*. La Paz, Murillo, Bolivia.
- Cooper, R. (2004). *Apache Práctico* (1ra Edicion ed.). Madrid: Anaya Multimedia. Recuperado el 15 de Agosto de 2013, de http://linux.ciberaula.com/articulo/linux_apache_intro
- CUATRORIOS. (2011). *Norma ISO-9126 para análisis de software*. Recuperado el 10 de Marzo de 2023, de http://www.cuattrorios.org/index.php?option=com_content&view=article&id=163:norma-iso-9126-para-an%C3%A1lisis-de-software&catid=39:blogsfeeds
- Dental, D. (15 de noviembre de 2020). *Historia clínica odontológica* . Obtenido de Historia clínica odontológica : <https://dtdental.co/historia-clinica-odontologica/>
- DESARROLLO WEB, M. S. (21 de agosto de 2015). *¿Qué es un sistema web?* Recuperado el 10 de Septiembre de 2023, de *¿Qué es un sistema web?*: <https://www.addappto.com/que-es-un-sistema-web/>

- DGSGIF, D. d. (Septiembre de 2000). *SIGMA*. Recuperado el 12 de Marzo de 2023, de <http://www.sigma.gob.bo/php/index.php>
- DGSGIF, D. d. (2013). *Sistema de Contrataciones Estatales SICOES*. Recuperado el 12 de Marzo de 2023, de <http://www.sicoes.gob.bo/paginicio/inicio.php>
- DGSGIF, D. G. (09 de Mayo de 2013). *Registro Unico de Proveedores del Estado*. Recuperado el 16 de Marzo de 2023, de http://sigep.sigma.gob.bo/proveedores/faces/SPrvRegProLis?_afLoop=149741624685000&_afWindowMode=0&_adf.ctrl-state=cn57a2iy0_18
- EcuRed. (2007). *Características de MagicDraw*. Recuperado el 12 de Marzo de 2023, de <http://www.ecured.cu/index.php/MagicDraw>
- Engineering, I. f. (07 de 09 de 2012). *UWE - UML - BASED WEB ENGINEERING*. Recuperado el 18 de Marzo de 2023, de <http://uwe.pst.ifi.lmu.de/>
- Felipe. (12 de Diciembre de 2020). *Qué es MariaDB y cuáles son sus características*. Obtenido de *Qué es MariaDB y cuáles son sus características*: <https://www.hostingplus.pe/blog/que-es-mariadb-y-cuales-son-sus-caracteristicas/>
- García, P. G. (10 de marzo de 2022). *¿Qué es un sistema de control?* Recuperado el 10 de Septiembre de 2023, de *¿Qué es un sistema de control?*: <https://industriasgsl.com/blogs/automatizacion/sistema-de-control>
- Gómes Vieites, A., y Suarez Rey, C. (2007). *Sistemas de Información Practicas para la Gestión Empresarial* (2da ed.). Mexico: ALFAOMEGAGRUPO EDITOR S.A.
- Gómez, R. C. (14 de 12 de 2017). *PREVALENCIA DE CARIES DENTAL EN NIÑOS DE LA CLÍNICA ESTOMATOLÓGICA DE LA UNIVERSIDAD SEÑOR DE SIPÁN*. Obtenido de *REVISTA CIENTIFICA SALUD Y VIDA SIPANENSE*: <https://revistas.uss.edu.pe/index.php/SVS/article/view/696>
- KOCH, N. (2000). *Software Engineering for Adaptive Hypermedia Systems. Referece Modeling Techniques and Development Process*. Germany: Ludeing.Maximilians-Universität München.

- Leal Castellano, M. Y., Leal Molina, Y. C., y Medina Castiblanco, L. C. (2011). *Taller Cliente Servidor*. Recuperado el 14 de Diciembre de 2011, de <http://basesii.wikispaces.com/file/view/Caracter%C3%ADsticas+de+la+arquitectura+Cliente.pdf>
- Maida, E. G., y Pacienza. (Diciembre de 2015). *Ingeniería de software*. Obtenido de Ingeniería de software: <https://repositorio.uca.edu.ar/bitstream/123456789/522/1/metodologias-desarrollo-software.pdf>
- Marqués, F. L. (25 de Enero de 2023). *Definición de historia clínica*. Obtenido de Definición de historia clínica: <https://clinic-cloud.com/blog/la-historia-clinica-paciente-sirve/>
- Martinez Guerrero, R. (23 de Marzo de 2009). *Sobre PostgreSQL*. Recuperado el 05 de Marzo de 2023, de http://www.postgresql.org/es/sobre_postgresql
- Miguel Angel, A. (19 de Septiembre de 2012). *Manual de JQuery*. Recuperado el 05 de Marzo de 2023, de <http://www.desarrolloweb.com/manuales/manual-jquery.html>
- Pressman, R. (2005). *Ingeniería del Software*.
- Pressman, R. (2010). *Ingeniería del software un Enfoque Practico*. Mexico: Mc Graw Hill. Recuperado el 10 de Septiembre de 2023, de Ingeniería del software un Enfoque Practico.
- Pressman, R. S. (2010). *Ingeniería del software un Enfoque Practico* (7ma. ed.). Mexico: Mc Graw Hill.
- Ricardo, S. D. (01 de Marzo de 2014). *OOHDM (MÉTODO DE DISEÑO HIPERMEDIA OBJETO ORIENTADO)*. Recuperado el 14 de Octubre de 2023, de OOHDM (MÉTODO DE DISEÑO HIPERMEDIA OBJETO ORIENTADO): <https://darjelingsilva.files.wordpress.com/2018/05/5-metd-oohdm.pdf>
- Souza, I. (9 de Marzo de 2020). *Descubre qué es el lenguaje de programación PHP y en qué situaciones se hace útil*. Obtenido de Descubre qué es el lenguaje de

programación PHP y en qué situaciones se hace útil:
<https://rockcontent.com/es/blog/php/>

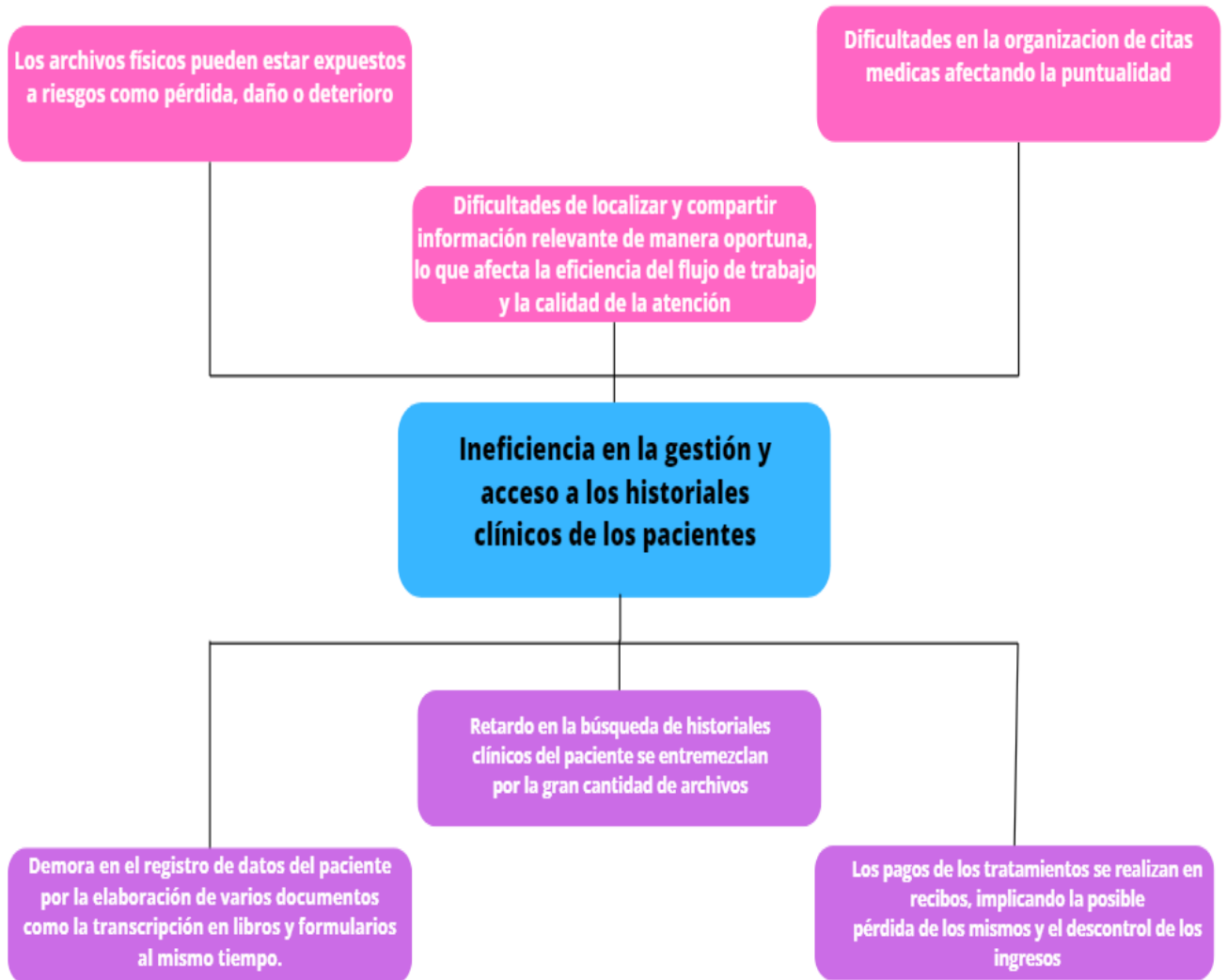
Suarez, J. (1 de octubre de 2020). *¿Qué es un dato?* Recuperado el 10 de Octubre de 2023, de *¿Qué es un dato?*: <https://concepto.de/dato/>

Thompson, I. (14 de septiembre de 2023). *¿Qué es Información?* Obtenido de *¿Qué es Información?*: <https://www.promonegocios.net/mercadotecnia/que-es-informacion.html>

UPEA. (19 de Octubre de 2012). *Universidad Pública de El Alto*. Recuperado el 3 de Marzo de 2022, de <http://www.upea.edu.bo/>

ANEXOS

Anexo A. Árbol de problemas



Anexo B. Árbol de Objetivos

