



**INGENIERIA DE SISTEMAS
PROYECTO DE GRADO**

**IMPLEMENTACIÓN DE UN SISTEMA
WEB PARA MEJORAR LA GESTIÓN Y
ORGANIZACIÓN DE LA INFORMACIÓN
EN LA CLÍNICA DERMATOLÓGICA
“DERMA - CAM” EN LA CIUDAD DE
COCHABAMBA EN LA GESTION 2024**

Silver Luis Aramayo Borges

Tutor: Lic. Alejandra Gómez Mendoza

**Proyecto de Grado para optar al grado de
Licenciatura en Ingeniería de Sistemas**

Cochabamba – Bolivia

2024

ABSTRACT

TITULO: "Implementación de un sistema web para mejorar la gestión y organización de la información en la clínica dermatológica “DERMA - CAM” en la ciudad de Cochabamba en la gestión 2024”

AUTOR (ES): Silver Luis Aramayo Borges

PROBLEMÁTICA

Dificultad en la gestión y organización de la información en la clínica dermatológica “DERMA - CAM”.

OBJETIVO GENERAL

Implementación de un sistema web para mejorar la gestión y organización de la información en la clínica dermatológica “DERMA - CAM”.

CONTENIDO

El proyecto se enfoca en desarrollar un sistema web para optimizar la administración de la clínica. Inicia con la identificación de problemas, establece objetivos y justifica la importancia del proyecto. El marco teórico aborda conceptos y revisiones de literatura relevantes. La metodología detalla el enfoque utilizado, fases del desarrollo y técnicas de recolección de datos. El análisis y diseño describen los requisitos y la arquitectura del sistema. La fase de desarrollo e implementación explica la codificación y configuración del sistema. Las pruebas aseguran la funcionalidad y aceptación mediante pruebas unitarias, de integración y de usuario.

CARRERA

: Ingeniería de Sistemas

TUTOR

: Lic. Alejandra Gómez Mendoza

DESCRIPTORES O TEMAS

: Desarrollo de Sistema Web

PERIODO DE INVESTIGACIÓN : MDG Gestión I/2024

E-MAIL DEL O LOS AUTORES : silveraramayo.b@gmail.com

DEDICATORIA

Este Proyecto de Grado está dedicado:

A mis padres, por su apoyo y amor incondicional, la educación que me brindaron, enseñarme buenos valores que ayudaron a forjar mi personalidad y formar a la persona que soy hoy en día.

A mi novia por el apoyo y amor que me brindo a lo largo de mi vida universitaria y por enseñarme a persistir y no darme por vencido por más difícil que sea el problema.

A mis hermanos por siempre estar a mi lado apoyándome, cuidándome, enseñándome y, sobre todo acompañándome a lo largo de mi vida.

AGRADECIMIENTOS

A: Mi tutora, Lic. Alejandra Gómez Mendoza, por su invaluable guía y apoyo durante todo el proceso de elaboración de este proyecto. Su experiencia y orientación fueron fundamentales para el desarrollo y éxito de esta iniciativa.

A: La Universidad Privada Domingo Savio, por brindarme una formación académica sólida y proporcionarme las herramientas necesarias para enfrentar y superar los desafíos de este proyecto.

A: La Dra. Carolina Achá Morató, dueña de la Clínica Dermatológica "DERMACAM", por su confianza y permitirme implementar este sistema que, sin duda, contribuirá a mejorar la gestión y organización de su clínica. Sin su colaboración y apoyo, este proyecto no habría sido posible.

ÍNDICE DE CONTENIDO

CAPÍTULO I	PÁG
INTRODUCCIÓN	
1.1 ANTECEDENTES.....	1
1.2 PLANTEAMIENTO DEL PROBLEMA	2
1.2.1. Situación Problemática Y/O Requerimiento De La Institución	2
1.2.2. Objeto De Estudio.....	3
1.2.3. Estudio De Soluciones.....	4
1.2.4. Pregunta De Investigación	5
1.3. OBJETIVOS DE LA INVESTIGACIÓN	5
1.3.1. Objetivo General	5
1.3.2. Objetivos Específicos.....	5
1.4. DEFINICIÓN DE VARIABLES	6
1.5. DELIMITACIÓN	8
1.5.1. Límite Temporal	8
1.5.2. Límite Geográfico.....	9
1.6. JUSTIFICACIÓN	9
1.6.1. Justificación Técnica	10
1.6.2. Justificación Económica.....	10
1.6.3. Justificación Social	10
1.7. TIPOLOGÍA DE PROYECTOS	11
1.8. TIPO Y ESTUDIO DE LA INVESTIGACIÓN	12
1.9. TÉCNICAS E INSTRUMENTOS DE INVESTIGACIÓN	13
1.8 POBLACIÓN Y MUESTRA	13
CAPÍTULO II	PÁG.
MARCO TEÓRICO	.
2.1. INTRODUCCIÓN	15
2.2. SISTEMA WEB	15
2.3. METODOLOGIA DE DESARROLLO	16
2.3.1. Modelo Tradicional O En Cascada.....	18
2.4. ARQUITECTURA MODELO VISTA CONTROLADOR	19
2.5. LENGUAJE DE PROGRAMACION.....	20

2.5.1.	PHP	22
2.6.	BASE DE DATOS.....	23
2.6.1.	Base De Datos SQL	24
2.6.2.	MySQL	25
2.6.3.	PHPMyAdmin	26
2.7.	FIGMA	27
2.8.	VISUAL STUDIO CODE	27
2.9.	TESTING	28
2.9.1.	PHPUnit.....	28
2.9.2.	Testim	29
2.10.	MÉTODO ALBRECHT.....	29

CAPÍTULO III . PÁG.
MARCO PRÁCTICO

3.1.	ANALISIS DE REQUERIMIENTOS	31
3.1.1.	Identificación De Actores	31
3.1.2.	Requerimientos Funcionales	32
3.1.3.	Requerimientos No Funcionales	35
3.1.4.	Definición De Procesos.....	38
3.2.	DISEÑO DEL SISTEMA	41
3.2.1.	Modelo De Casos De Uso	41
3.2.1.1.	Casos De Uso Usuario Administrador.....	42
3.2.1.2.	Casos De Uso Usuario Médico	43
3.2.1.3.	Casos De Uso Usuario Paciente.....	44
3.2.1.4.	Casos De Uso Usuario Asistente.....	45
3.2.2.	Diagrama De Secuencia.....	46
3.2.3.	Diagrama De Componentes.....	49
3.2.4.	Diagrama De Clases	49
3.2.5.	Diagrama De Entidad - Relación	51
3.2.6.	Interfaces De Usuario	52
3.2.6.1.	Landing Page	52
3.2.6.2.	Registro	54
3.2.6.3.	Inicio De Sesión	55

3.2.6.4. Home Administrador.....	56
3.2.6.5. Panel Pacientes	57
3.3. DESARROLLO DEL SISTEMA	58
3.3.1. Entorno De Desarrollo Integrado (Ide)	58
3.3.2. Desarrollo De La Base De Datos	60
3.3.2.1. Phpmyadmin	60
3.3.2.2. Desarrollo De La Base De Datos	61
3.3.3. Codificación Del Sistema Web.....	62
3.3.3.1. Registrarse E Iniciar Sesión.....	62
3.3.3.2. Panel De Administración.....	64
3.3.3.3. Gestión De Usuarios	64
3.3.3.4. Gestión De Pacientes.....	67
3.3.3.5. Gestión De Citas.....	70
3.3.3.6. Gestión De Productos	75
3.3.3.7. Gestión De Ventas.....	77
3.3.3.8. Vistas Para Pacientes	80
3.3.4. Adaptabilidad Del Sistema	82
3.4. PRUEBAS DE VALIDACIÓN	83
3.4.3. Pruebas Unitarias.....	83
3.4.4. Pruebas De Integración	84
3.4.5. Testim	85
3.4.5.1. Pruebas End To End (E2e)	88
3.4.5.2. Escenarios De Prueba	88
3.5. DESPLIEGUE DEL SISTEMA	91
3.5.5. Hostinger.....	91
CAPÍTULO IV	PÁG.
ANÁLISIS DE VIABILIDAD	.
4.1 VIABILIDAD TÉCNICA	95
4.2. VIABILIDAD ECONÓMICA	96
4.2.1. Estimación Del Costo Del Proyecto	96
4.2.1.1. Identificación Y Clasificación De Los Componentes.....	96
4.2.1.2. Puntos De Función Ajustado	97

4.2.1.3. Factor Productividad	99
4.2.1.4. Estimación Horas Hombre	99
4.2.2. Costos De Funcionamiento Del Proyecto	100
4.2.3. Conclusión	101
4.3. VIABILIDAD OPERATIVA	102

CAPÍTULO V **PÁG.**

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES	103
5.2. RECOMENDACIONES.....	104

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE TABLAS

TABLA N° 1: VALORACIÓN DE SOLUCIONES	4
TABLA N° 2: VARIABLES DE LA INVESTIGACIÓN	6
TABLA N° 3: DELIMITACIÓN DEL PROYECTO.....	8
TABLA N° 4: ALCANCES DE LA INVESTIGACIÓN	11
TABLA N° 5: DECLARACIÓN DE USO DE INSTRUMENTOS.....	13
TABLA N° 6: COMPARACIÓN DE METODOLOGIAS	17
TABLA N° 7: COMPARACIÓN DE LENGUAJES DE PROGRAMACIÓN.....	21
TABLA N° 8: COMPARACIÓN DE BASE DE DATOS	24
TABLA N° 9: LISTA DE ACTORES	32
TABLA N° 10: REQUERIMIENTOS FUNCIONALES DE USUARIO ADMINISTRADOR	33
TABLA N° 11: REQUERIMIENTOS FUNCIONALES DE USUARIO MÉDICO	34
TABLA N° 12: REQUERIMIENTOS FUNCIONALES DE USUARIO PACIENTE	34
TABLA N° 13: REQUERIMIENTOS FUNCIONALES DE USUARIO ASISTENTE	35
TABLA N° 14: REQUERIMIENTOS NO FUNCIONALES DE USUARIO ADMINISTRADOR	35
TABLA N° 15: REQUERIMIENTOS NO FUNCIONALES DE USUARIO MÉDICO	36
TABLA N° 16: REQUERIMIENTOS NO FUNCIONALES DE USUARIO PACIENTE.....	37
TABLA N° 17: REQUERIMIENTOS NO FUNCIONALES DE USUARIO ASISTENTE	37
TABLA N° 18: REQUERIMIENTOS NO FUNCIONALES DEL SISTEMA.....	38

TABLA N° 19: REQUERIMIENTOS TÉCNICOS DE HARDWARE Y SOFTWARE PARA EL USO DEL SISTEMA	95
TABLA N° 20: REQUERIMIENTOS TÉCNICOS DE HARDWARE Y SOFTWARE PARA EL FUNCIONAMIENTO DEL SISTEMA.....	95
TABLA N° 21: PUNTOS DE FUNCION SISTEMA WEB	97
TABLA N° 22: ESCALA DE INFLUECIA FACTOR AJUSTE	98
TABLA N° 23: AJUSTE DE LOS PUNTOS DE FUNCIÓN.....	98
TABLA N° 24: FACTOR PRODUCTIVIDAD.....	99
TABLA N° 25: COSTO DEL DESARROLLO DEL PROYECTO	100
TABLA N° 26: COSTO MENSUAL DE FUNCIONAMINETO.....	101
TABLA N° 27: COSTO ANUAL DE FUNCIONAMINETO.....	101

ÍNDICE DE ILUSTRACIONES

ILUSTRACIÓN 1: ÁRBOL DE PROBLEMAS	3
ILUSTRACIÓN N° 2: UBICACIÓN.....	9
ILUSTRACIÓN N° 3: FORMULA DE MUESTREO ALEATORIO SIMPLE	14
ILUSTRACIÓN N° 4: MODELO EN CASCADA	19
ILUSTRACIÓN N° 5: CASOS DE USO DEL USUARIO ADMINISTRADOR.....	42
ILUSTRACIÓN N° 6: CASOS DE USO DEL USUARIO MÉDICO	43
ILUSTRACIÓN N° 7: CASOS DE USO DEL USUARIO PACIENTE	44
ILUSTRACIÓN N° 8: CASOS DE USO DEL USUARIO ASISTENTE	45
ILUSTRACIÓN N° 9: DIAGRAMA DE SECUENCIA REGISTRARSE.....	46
ILUSTRACIÓN N° 10: DIAGRAMA DE SECUENCIA RESERVA DE CITA	47
ILUSTRACIÓN N° 11: DIAGRAMA DE SECUENCIA REALIZAR COMPRA.....	48
ILUSTRACIÓN N° 12: DIAGRAMA DE COMPONENTES	49
ILUSTRACIÓN N° 13: DIAGRAMA DE CLASES.....	50
ILUSTRACIÓN N° 14: DIAGRAMA ENTIDAD-RELACIÓN.....	51
ILUSTRACIÓN N° 15: PALETA DE COLORES	52
ILUSTRACIÓN N° 16: LOGO DE LA CLÍNICA.....	52
ILUSTRACIÓN N° 17: LANDING PAGE.....	53
ILUSTRACIÓN N° 18: DISEÑO UI DE REGISTRO.....	54

ILUSTRACIÓN N° 19: DISEÑO UI INICIO DE SESIÓN	55
ILUSTRACIÓN N° 20: DISEÑO PANEL DE ADMINISTRACIÓN.....	56
ILUSTRACIÓN N° 21: DISEÑO UI PANEL DE PACIENTES.....	57
ILUSTRACIÓN N° 22: LOGOS BOOTSTRAP Y PHP.....	58
ILUSTRACIÓN N° 23: INTERFAZ DE VISUAL STUDIO CODE	59
ILUSTRACIÓN N° 24: CARPETAS ORDENADAS SEGÚN ARQUITECTURA MVC.....	59
ILUSTRACIÓN N° 25: INTERFAZ DE PHPMYADMIN	60
ILUSTRACIÓN N° 26: TABLAS DE LA BASE DE DATOS	61
ILUSTRACIÓN N° 27: ESTRUCTURA DE LA BASE DE DATOS	61
ILUSTRACIÓN N° 28: FORMULARIO PARA REGISTRARSE	63
ILUSTRACIÓN N° 29: INICIAR SESIÓN	63
ILUSTRACIÓN N° 30: PANEL DE ADMINISTRACIÓN	64
ILUSTRACIÓN N° 31: GESTOR DE USUARIOS	65
ILUSTRACIÓN N° 32: FORMULARIO CREAR USUARIO.....	65
ILUSTRACIÓN N° 33: EDITAR USUARIO.....	66
ILUSTRACIÓN N° 34: ELIMINAR USUARIO	67
ILUSTRACIÓN N° 35: GESTOR DE PACIENTES	68
ILUSTRACIÓN N° 36: FORMULARIO CREACIÓN DE PACIENTE	68
ILUSTRACIÓN N° 37: EDITAR PACIENTE	69

ILUSTRACIÓN N° 38: ELIMINAR PACIENTE.....	69
ILUSTRACIÓN N° 39: AGENDAR CITA.....	70
ILUSTRACIÓN N° 40: PLANTILLAS CREADA EN META	71
ILUSTRACIÓN N° 41: MENSAJE DE CONFIRMACIÓN DE CITA.....	71
ILUSTRACIÓN N° 42: CÓDIGO PARA ENVIÓ DEL MENSAJE.....	72
ILUSTRACIÓN N° 43: LISTA DE CITAS	73
ILUSTRACIÓN N° 44: INGRESAR DIAGNÓSTICO Y TRATAMIENTO	73
ILUSTRACIÓN N° 45: VER HISTORIAL DE LA CITA	74
ILUSTRACIÓN N° 46: FACTURA DE LA CITA GENERADA	75
ILUSTRACIÓN N° 47: GESTOR DE PRODUCTOS	76
ILUSTRACIÓN N° 48: CREAR PRODUCTO.....	76
ILUSTRACIÓN N° 49: AGREGAR STOCK.....	77
ILUSTRACIÓN N° 50: CARRITO DE COMPRAS.....	78
ILUSTRACIÓN N° 51: REALIZAR COMPRA	78
ILUSTRACIÓN N° 52: LISTA DE VENTAS.....	79
ILUSTRACIÓN N° 53: PDF DE LA FACTURA.....	79
ILUSTRACIÓN N° 54: PANEL DE ADMINISTRACIÓN CLIENTE	80
ILUSTRACIÓN N° 55: CITAS VISTA PACIENTE	81
ILUSTRACIÓN N° 56: COMPRAS VISTA CLIENTE	81

ILUSTRACIÓN N° 57: DISEÑO RESPONSIVE PARA MÓVILES	83
ILUSTRACIÓN N° 58: CODIGO DE LAS PRUEBAS UNITARIAS	84
ILUSTRACIÓN N° 59: CODIGO DE LAS PRUEBAS DE INTEGRACIÓN	85
ILUSTRACIÓN N° 60: LOGO TESTIM	86
ILUSTRACIÓN N° 61: CREAR UNA PRUEBA EN TESTIM	86
ILUSTRACIÓN N° 62: SOLICITUD DE LA URL DEL SISTEMA.....	87
ILUSTRACIÓN N° 63: RECOLGIDA DE PASOS.....	87
ILUSTRACIÓN N° 64: EJECUCION DE LA PRUEBA APROBADA.....	88
ILUSTRACIÓN N° 65: PRUEBAS GUARDAS EN TESTIM	89
ILUSTRACIÓN N° 66: PRUEBAS DESDE LA INTERFAZ DE TESTIM	90
ILUSTRACIÓN N° 67: MENSAJE DE TEST COMPLETADO.....	90
ILUSTRACIÓN N° 68: BASE DE DATOS EN HOSTINGER.....	92
ILUSTRACIÓN N° 69: ARCHIVOS DEL SISTEMA EN EL HOSTING	92
ILUSTRACIÓN N° 70: VISTA PREVIA DEL SISTEMA EN HOSTINGER	93
ILUSTRACIÓN N° 71: SISTEMA DESDE EL DOMINO.....	94
ILUSTRACIÓN N° 72: CERTIFICADO SSL.....	94
ILUSTRACIÓN N° 73: PROCESOS MANUALES EN AGENDA DE CITAS.	113
ILUSTRACIÓN N° 74: PROCESOS MANUALES EN REGISTROS.....	114
ILUSTRACIÓN N° 75: PROCESOS MANUALES EN LA FACTURACION	115

ILUSTRACIÓN N° 76: DOCUMENTACION DUPLICADA.....	115
ILUSTRACIÓN N° 77: ERRORES EN LA DOCUMENTACIÓN.....	116
ILUSTRACIÓN N° 78: ENTREVISTA CON LA DUEÑA DE LA CLÍNICA.....	118
ILUSTRACIÓN N° 79: ENTREVISTA CON LA DUEÑA DE LA CLÍNICA.....	119
ILUSTRACIÓN N° 80: CODIGO DE REGISTRARSE	121
ILUSTRACIÓN N° 81: CODIGO DE INICIAR SESIÓN	121
ILUSTRACIÓN N° 82: CODIGO PANEL DE ADMINISTRACIÓN.....	122
ILUSTRACIÓN N° 83: CODIGO GESTOR DE USUARIOS.....	122
ILUSTRACIÓN N° 84: CODIGO FORMULARIO PARA CREAR USUARIO	123
ILUSTRACIÓN N° 85: CODIGO EDITAR USUARIO	123
ILUSTRACIÓN N° 86: CODIGO ELIMINAR USUARIO	124
ILUSTRACIÓN N° 87: CODIGO AGENDAR CITAS.....	124
ILUSTRACIÓN N° 88: CODIGO LISTA DE PACIENTES.....	125
ILUSTRACIÓN N° 89: CODIGO CREAR HISTORIAL DE CITA	125
ILUSTRACIÓN N° 90: CÓDIGO VER HISTORIAL	126
ILUSTRACIÓN N° 91: CÓDIGO GERNERAR FACTURA PARA CITAS.....	126
ILUSTRACIÓN N° 92: CÓDIGO DE GESTOR DE PRODUCTOS	127
ILUSTRACIÓN N° 93: CODIGO DE AGREGAR STOCK	127
ILUSTRACIÓN N° 94: CÓDIGO DE CARRITO DE COMPRAS.....	128

ILUSTRACIÓN N° 95: CÓDIGO DE REALIZAR COMPRA128

ILUSTRACIÓN N° 96: CÓDIGO DE FACTURA DE PRODUCTOS129

CAPÍTULO I

INTRODUCCIÓN

INTRODUCCIÓN**1.1. ANTECEDENTES**

En distintas partes del mundo, la necesidad de contar con información precisa y oportuna en el ámbito hospitalario ha sido reconocida y abordada de diversas maneras a lo largo del tiempo. En Zaragoza, España, según Salvador Olivan (1997), se resaltaba este creciente requerimiento, reconociendo la complejidad del entorno hospitalario y la necesidad de sistemas de información adecuados para satisfacerla. Se identificaron cuatro áreas principales donde estos sistemas podrían ser útiles: información externa sobre el entorno, información financiera interna, información administrativa y de gestión, y, crucialmente, información clínica.

Este enfoque en la información clínica, como una parte fundamental de los sistemas de información en el ámbito de la salud, se refuerza con la implementación de sistemas específicos en otros lugares. Por ejemplo, en América según un artículo de OasisCom (2023), a partir de abril de 2001, se estableció un Sistema de Información sobre los servicios de salud, que abarcaba todas las instituciones prestadoras de servicios de salud. Este sistema tenía como objetivo generar datos desde la facturación, siguiendo especificaciones técnicas establecidas en la Resolución 3374 de 2000. Este enfoque indica una preocupación por la recopilación y gestión de datos relacionados con los servicios de salud a nivel nacional.

Mientras tanto, en Bolivia el Ministerio de Salud (2014), implementó el Sistema de Información Integrado Clínico Estadístico (SICE), el cual tiene como propósito principal centralizar todos los registros médicos de un establecimiento de salud en un único sistema. Esta iniciativa apuntaba a mejorar la accesibilidad y gestión de la información clínica y estadística dentro del país.

Según Achá (2024) la Clínica Dermatológica “DERMA - CAM” fundada en la ciudad de Cochabamba el año 2010 por la Dra. Carolina Achá, especialista en el área de dermatología con más de 20 años de experiencia; comenzó como un consultorio dermatológico en la ciudad de Cochabamba. Con el paso del tiempo logró

consolidarse gracias a su excelente servicio, calidad y seguridad en la atención de sus pacientes.

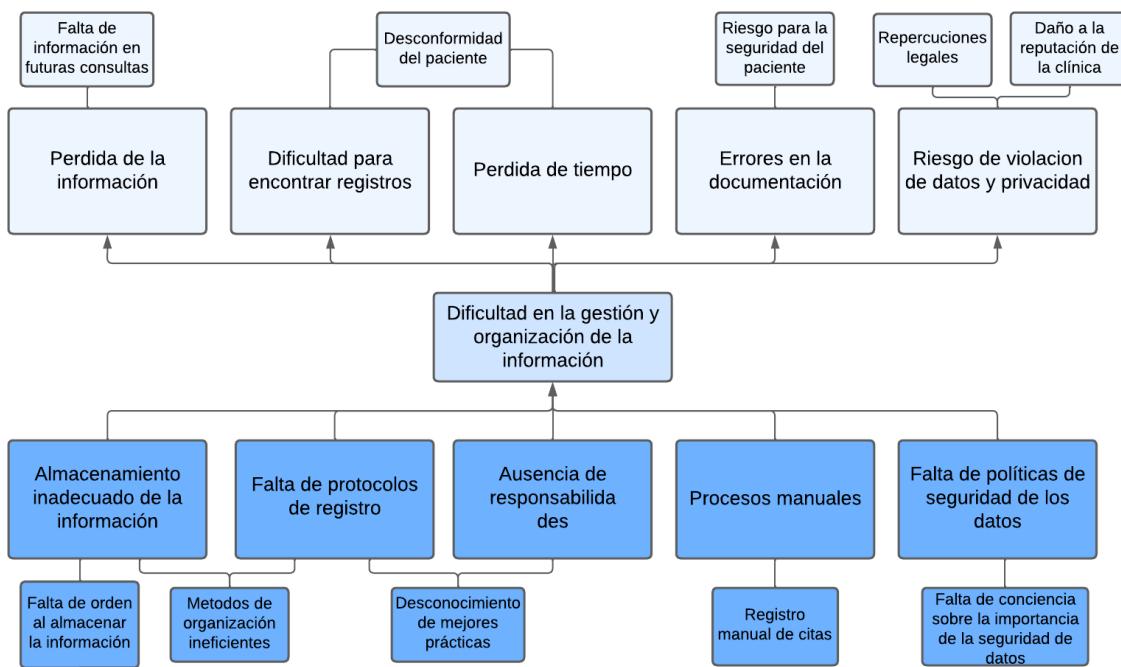
1.2. PLANTEAMIENTO DEL PROBLEMA

El estudio de soluciones ayudará a analizar, evaluar y valorar las soluciones de diferentes opciones para abordar un problema o satisfacer una necesidad.

1.2.1. Situación problemática y/o requerimiento de la institución

La Clínica Dermatológica "DERMA-CAM" enfrenta desafíos significativos debido a la falta de organización y gestión eficiente de su información. Los procesos manuales vigentes son propensos a errores y carecen de una estructura clara de responsabilidades, lo que resulta en un almacenamiento inadecuado de datos y documentación inconsistente como se puede apreciar en los anexos (ver Anexo I). Estos problemas representan un riesgo para la integridad de la información, provocando retrasos en la atención de los pacientes y posibles errores en los registros como se puede ver en el árbol de problemas que se puede ver en la ilustración 1, lo que coloca a la clínica en una posición desventajosa frente a la competencia.

ILUSTRACIÓN 1: ÁRBOL DE PROBLEMAS



FUENTE: Elaboración Propia

1.2.2 Objeto de estudio

La implementación de un sistema de información para la organización y gestión de la información en la clínica dermatológica es necesaria por la capacidad para mejorar la accesibilidad, organización y eficiencia de la información. Al eliminar la dependencia de registros manuales, se reducen los errores, se aumenta la coordinación entre los empleados en la atención y se mejora la experiencia del paciente, lo que convierte a este proyecto en una inversión valiosa y necesaria para la clínica.

1.2.3 Estudio de soluciones

TABLA N° 1: VALORACIÓN DE SOLUCIONES

CASOS DE ESTUDIO	PROBLEMA	SOLUCIÓN APLICADA	VALORACIÓN
Título: Implementación De Sistema De Reserva De Citas Médicas En Línea - Sanna S.A.C. Autor: Luis Gabriel Gonzales León Universidad San Ignacio de Loyola Lima – Perú 2019	¿Cómo implementar un sistema web de entrega de citas médicas a los pacientes de la Red de Clínicas SANNA, que sirva para reducir el tiempo de espera para reservar una cita?	Implementación de sistema de reserva de citas médicas en línea.	La solución es completa y es una potencial solución para la reserva de citas médicas.
Título: Sistema De administración Y Control De Historiales clínicos Para Los Consultorios De La UMSA Autor: Rosmery Lozano Flores Universidad Mayor de San Andrés Lugar: La Paz – Bolivia 2014	¿Cómo se puede administrar y controlar la información de los historiales clínicos de los pacientes para mejorar las actividades que realiza el personal como registrar, consultar, buscar y procesar informes o reportes en la UMSA?	Desarrollo e implementación un Sistema de Administración y Control de Historiales Clínicos de los pacientes universitarios.	La propuesta de solución tiene un gran potencial para mejorar la gestión de historiales clínicos.
Título: Diseño Y Elaboración De Un Sistema De Información Para El Análisis Y Control De Historias Clínicas Para La Clínica Maternidad Padre Luis Variara Autor: José Antonio Postes de la Torre Universidad Estatal Península de Santa Elena Lugar: Playas – Ecuador Año: 2010	El incremento de la demanda de los pacientes por parte de los directivos de institución llegaron a tomar la decisión de implementar un sistema informático.	Diseño y elaboración de un sistema de información para el análisis y control de historias clínicas.	El proyecto es una solución eficaz para mejorar la atención de pacientes la cual presenta una solución diferente a las anteriores ya que esta es un sistema de escritorio.

FUENTE: Elaboración propia en base a (Gonzales León, 2019; Lozano Flores, 2014; Postes de la Torre, 2010)

1.2.4 Pregunta de investigación

¿Cómo mejorar la gestión y organización de la información en la clínica dermatológica “DERMA - CAM” en la ciudad de Cochabamba en la gestión 2024?

1.3. OBJETIVOS DE LA INVESTIGACIÓN

Con la implementación del sistema de información, los objetivos de investigación podrán ayudar a la clínica a mejorar la gestión de su información y como resultado alcanzar una serie de beneficios claves para la clínica.

1.3.1 Objetivo General

Implementar un sistema web para mejorar la gestión y organización de la información en la Clínica Dermatológica “DERMA – CAM” en la ciudad de Cochabamba en la gestión 2024.

1.3.2 Objetivos específicos

- Analizar los requerimientos para definir los componentes del sistema y las interacciones entre ellos.
- Diseñar la estructura y los componentes del sistema para guiar el desarrollo y la implementación.
- Desarrollar el sistema de acuerdo con los diseños y especificaciones establecidos para así pasar a código funcional y ejecutable.
- Realizar pruebas para validar que el software cumpla con los requerimientos establecidos.
- Realizar el despliegue del sistema en un entorno de producción para su uso por parte de los usuarios finales.

1.4. DEFINICIÓN DE VARIABLES

TABLA N° 2: VARIABLES DE LA INVESTIGACIÓN

Pregunta de investigación ¿Cómo mejorar la gestión y organización de la información en la clínica dermatológica “DERMA - CAM” en la ciudad de Cochabamba en la gestión 2024?				
Objetivo general Implementar un sistema web para mejorar la gestión y organización de la información en la Clínica Dermatológica “DERMA – CAM” en la ciudad de Cochabamba en la gestión 2024.				
Variable	Definición conceptual	Definición operacional	Indicadores (qué se espera notar u obtener de esta variable)	Herramientas (Instrumentos)
Implementación de un sistema web para mejorar la gestión y organización de la información en la clínica dermatológica “DERMA - CAM”.	“La implementación de sistemas web implica la creación de aplicaciones y software que funcionan a través de un navegador web. Estos sistemas pueden ser herramientas internas de una empresa, aplicaciones de usuario final o plataformas en línea.” (Borovskoy, 2023).	La implementación de un sistema web se puede medir según diferentes indicadores como ser, Tiempo de carga del sistema web, Analítica web, Pruebas de rendimiento, entre otros los cuales ayudan a garantizar un rendimiento óptimo del sistema web, mejorar la experiencia de usuario y mantener la seguridad.	I1. Tiempo de carga del sistema web.	$\frac{\text{Tiempo de Carga promedio}}{\text{Puntuación de velocidad de sistema}}$
			I2. Analítica web.	$\frac{\text{Trafico total del sistema}}{\text{Eficacia general del sistema}}$
			I3. Pruebas de rendimiento.	$\frac{\text{Tasa de Errores}}{\text{Tiempo de respuesta del servidor}}$

FUENTE: Elaboración propia en base a (Borovskoy, 2023)

INTRODUCCIÓN

TABLA N° 3: VARIABLES DE LA INVESTIGACIÓN

Pregunta de investigación ¿Cómo mejorar la gestión y organización de la información en la clínica dermatológica “DERMA - CAM” en la ciudad de Cochabamba en la gestión 2024?				
Objetivo general Implementar un sistema web para mejorar la gestión y organización de la información en la Clínica Dermatológica “DERMA – CAM” en la ciudad de Cochabamba en la gestión 2024.				
Variable	Definición conceptual	Definición operacional	Indicadores (qué se espera notar u obtener de esta variable)	Herramientas (Instrumentos)
Dificultad en la gestión y organización de la información en la clínica dermatológica “DERMA - CAM”.	“Gestión de la información (GI) es la denominación convencional de un conjunto de procesos por los cuales se controla el ciclo de vida de la información, desde su obtención (por creación o captura), hasta su disposición final (su archivo o eliminación). Tales procesos también comprenden la extracción, combinación, depuración y distribución de la información a los interesados.” (toximed, 2017).	La dificultad para la gestión y organización se pueden medir según varios enfoques y métricas como ser, entrevistas, análisis de la eficiencia operativa, comparación de mejores prácticas y estándares, entre otras las cuales ayudaran a identificar y abordar áreas de mejora, mejorar la eficiencia y productividad, reducir riesgos y vulnerabilidades, y aumentar la satisfacción del cliente y del empleado	I1. Indicador de eficacia (Satisfacción del Usuario) $\frac{\text{Total de Usuarios Satisfechos}}{\text{Total Usuarios atendidos}} * 100\%$	Encuesta.
			I2. Ratio de errores en los datos. $\frac{\text{Número de Errores}}{\text{Total de registros}} * 100\%$	Documentación. Registros.
			I3. Calidad de datos. $\frac{\text{Número de Datos Correctos}}{\text{Número Total de Datos}} * 100\%$	Documentación. Registros.

FUENTE: Elaboración propia en base a (toximed, 2017)

1.5. DELIMITACIÓN

TABLA N° 3: DELIMITACIÓN DEL PROYECTO

Límites	Justificación
¿Hasta dónde se quiere llegar con este proyecto de grado? El proyecto llegará hasta la fase de implementación del sistema el cual estará alojado en un servidor.	El proyecto de grado tiene como objetivo principal alcanzar la fase de implementación del sistema, la cual implica alojar la solución en un servidor, esta decisión se fundamenta en la necesidad de cumplir con los objetivos definidos para el proyecto, donde la entrega de un sistema completamente funcional es esencial.
¿Qué cosas pretende solucionar este proyecto? Con el proyecto se pretende solucionar la dificultad en la gestión y organización de la información en la Clínica Dermatológica “DERMA - CAM”.	Con este proyecto se podrá mejorar la gestión y organización de la información al implementar un sistema web que ayudará a la clínica, evitando la perdida de la información, facilitando encontrar registros, disminuyendo errores en la documentación y el riesgo de violación de datos y privacidad.
¿Se pondrá a prueba el proyecto o sólo se ejecutará una experiencia piloto o no es necesario ninguna? No se realizará ninguna.	Aunque la realización de pruebas y experiencias piloto puede ser útil para validar el funcionamiento del sistema y realizar ajustes necesarios, en este caso, la implementación directa en un entorno operativo puede ser considerada como una medida efectiva y necesaria.

FUENTE: Elaboración propia

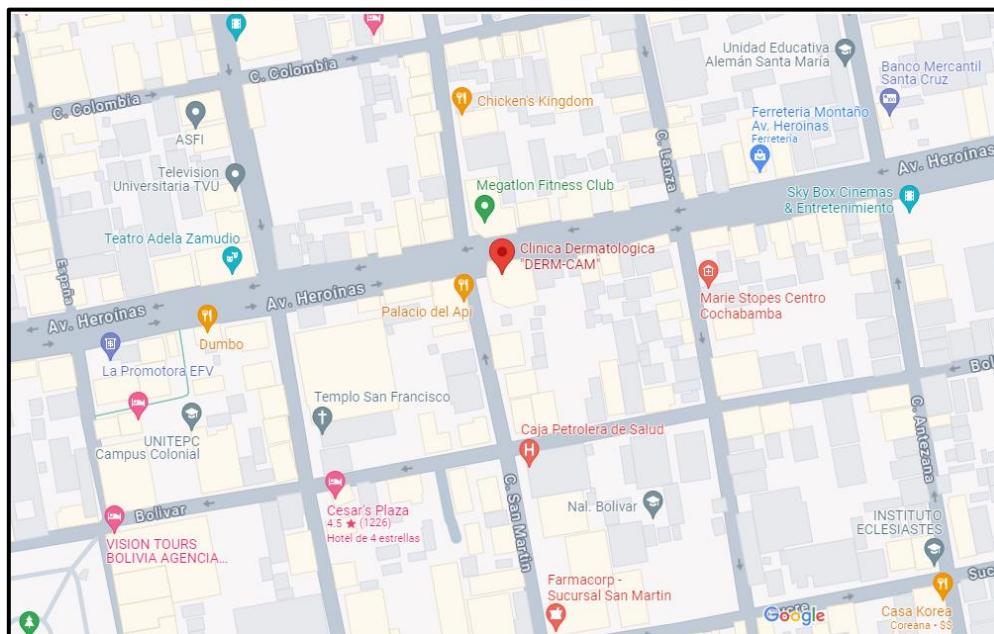
1.5.1. Límite temporal

El proyecto se llevará a cabo en el primer semestre del año 2024, siguiendo el calendario académico de la modalidad de grado de la Universidad Privada Domingo Savio. Este periodo abarca desde febrero hasta julio del presente año.

1.5.2. Límite geográfico

El proyecto se realizará en Bolivia en el departamento de Cochabamba en la provincia Cercado en la Clínica Dermatológica “Achá”, ubicado en la Avenida Heroínas edif. “CENTRUM” como se puede observar a continuación, en la siguiente ilustración.

ILUSTRACIÓN N° 2: UBICACIÓN



FUENTE: (Google Maps, 2024)

1.6. JUSTIFICACIÓN

La implementación de un sistema web en la clínica dermatológica responde a la necesidad de modernizar y optimizar los procesos de gestión de la información médica en un entorno cada vez más digitalizado.

Contribuye a mejorar la calidad de la atención médica al facilitar el acceso rápido y preciso a los registros de los pacientes, lo que permite diagnósticos más precisos y tratamientos más efectivos.

1.6.1. Justificación técnica

Los avances tecnológicos en el campo de la informática médica han hecho posible el desarrollo de sistemas web diseñados para clínicas dermatológicas, que ofrecen funcionalidades adaptadas a las necesidades y particularidades de este tipo de establecimientos.

La implementación de un sistema web permitirá la integración de diferentes procesos y datos clínicos, como historias clínicas electrónicas, resultados de pruebas diagnósticas entre otros, en una plataforma centralizada y segura.

La utilización de tecnología, como el cifrado de datos y los sistemas de autenticación, garantiza la seguridad y confidencialidad de la información del paciente, cumpliendo con los estándares de privacidad y regulaciones legales en materia de salud.

1.6.2. Justificación económica

Aunque la implementación inicial del sistema web conlleva una inversión significativa en términos de adquisición de software, hardware y capacitación del personal, a largo plazo se espera obtener un retorno de la inversión a través de la mejora en la eficiencia operativa y la reducción de costos asociados con la gestión de registros en papel y la corrección de errores.

La optimización de los procesos administrativos, como la gestión de citas, facturación y seguimiento de pacientes, puede resultar en ahorros significativos de tiempo y recursos, lo que se traduce en una mayor rentabilidad para la clínica dermatológica.

1.6.3. Justificación social

La implementación de un sistema web en la clínica dermatológica contribuye a mejorar la accesibilidad y calidad de la atención médica para los pacientes, lo que tiene un impacto positivo en su bienestar y satisfacción.

Al promover una atención médica más eficiente y coordinada, el sistema de información ayuda a reducir los tiempos de espera y mejorar la experiencia general del paciente en la clínica.

Además, la seguridad y confidencialidad de la información del paciente garantizadas por el sistema de información refuerzan la confianza y la relación de la clínica con sus pacientes, así como su reputación en la comunidad.

1.7. TIPOLOGÍA DE PROYECTOS

TABLA N° 4: ALCANCES DE LA INVESTIGACIÓN

ALCANCES	JUSTIFICACIÓN
Descriptivo “En este alcance de la investigación, ya se conocen las características del fenómeno y lo que se busca, es exponer su presencia en un determinado grupo humano. En el proceso cuantitativo se aplican análisis de datos de tendencia central y dispersión. En este alcance es posible, pero no obligatorio, plantear una hipótesis que busque caracterizar el fenómeno del estudio.” (Ramos, 2020).	La investigación descriptiva sería aplicable a este proyecto porque proporcionaría una comprensión completa de la situación actual de la clínica dermatológica y permitiría definir objetivos claros y específicos para la implementación del sistema de información. Esto sería fundamental para el diseño, desarrollo e implementación exitosos del sistema, asegurando que satisfaga las necesidades y expectativas tanto de la clínica como de los pacientes.
Explicativo “En este alcance de la investigación se busca una explicación y determinación de los fenómenos. En el contexto cuantitativo se pueden aplicar estudios de tipo predictivo en donde se pueda establecer una relación causal entre diversas variables. Por otro lado, los estudios experimentales, en los cuales se pueda generar una manipulación intencionada de la variable independiente, pueden permitir comprobar hipótesis que expliquen el comportamiento de un determinado fenómeno.” (Ramos, 2020).	La investigación explicativa sería aplicable a este proyecto porque permitiría ir más allá de simplemente describir la situación actual, buscando comprender las causas subyacentes de la desorganización y analizando críticamente el impacto del nuevo sistema de información. Esto proporcionaría información valiosa para mejorar continuamente la gestión y la eficiencia en la clínica dermatológica.

FUENTE: Elaboración propia en base a (Ramos, 2020)

1.8. TIPO Y ESTUDIO DE LA INVESTIGACIÓN

El tipo de estudio será de diseño no experimental longitudinal.

“La investigación no experimental, es cualquier investigación en la que resulta imposible manipular variables o asignar aleatoriamente a los sujetos o a las condiciones. De hecho, no hay condiciones o estímulos a los cuales se expongan los sujetos del estudio. Los sujetos son observados en su ambiente natural, en su realidad.” (Agueldo, Aigneran, & Ruiz Restrepo, 2010, pág. 10) .

La investigación no experimental sería aplicable a este proyecto porque permitiría observar, describir y analizar la situación actual y los efectos de la intervención sin manipular deliberadamente variables o aplicar tratamientos específicos. Esto sería fundamental para comprender la complejidad del problema y evaluar los impactos del sistema de información en la clínica dermatológica.

“Los diseños longitudinales realizan observaciones en dos o más momentos o puntos en el tiempo. Si estudian a una población son diseños de tendencia, Si analizan a una subpoblación o grupo específico son diseños de análisis evolutivo de grupo y si estudian a los mismos sujetos son diseños panel.” (Agueldo, Aigneran, & Ruiz Restrepo, 2010, pág. 41).

El enfoque longitudinal sería apropiado para este proyecto porque permite un análisis detallado y continuo de los cambios en la clínica dermatológica a medida que se implementa y se adapta el sistema de información. Esto proporciona una comprensión más completa del impacto del sistema y permite realizar ajustes en función de las necesidades cambiantes de la clínica.

1.9. TÉCNICAS E INSTRUMENTOS DE INVESTIGACIÓN

TABLA N° 5: DECLARACIÓN DE USO DE INSTRUMENTOS

Tipo	Instrumento	A quién o a qué	Para qué
Teórico	Tabla de comparación de contenidos	A las prácticas y estándares que maneja la clínica.	Para comparar las mejores prácticas y estándares.
	Árbol de problemas	Al problema principal de la clínica.	Para identificar sus causas y efectos.
	Diagramas UML	Al sistema web propuesto.	Para representar la estructura y el comportamiento del sistema web.
Empírico	Observación (tipo)	A los procesos operativos que dependen de la información.	Para analizar la eficiencia de dichos procesos.
	Entrevista	Al dueño de la clínica.	Para verificar la facilidad de búsqueda y recuperación de información, y evaluación de la calidad de la información disponible.
Matemático	PageSpeed Insights	Al sistema web implementado	Para conocer el tiempo de carga promedio y la puntuación de velocidad del sistema web.
	Google Analytics	Al sistema web implementado	Para conocer el tráfico y la eficacia general que tiene el sistema.
	Apache JMeter	Al sistema web implementado	Para identificar la tasa de errores y el tiempo de respuesta del servidor.

FUENTE: Elaboración Propia, en base a declaración de instrumentos.

1.8. POBLACIÓN Y MUESTRA

En el contexto del desarrollo de un sistema web para la clínica dermatológica “DERMA - CAM”, es importante considerar tanto la población de usuarios internos como externos para garantizar la eficacia y la usabilidad del sistema. Si bien la población de empleados de la clínica es de 4 usuarios los cuales son la muestra y a los que se les realizara una entrevista para obtener información necesaria para el avance del proyecto, la población de clientes o pacientes es fundamental para evaluar la experiencia del usuario y la funcionalidad del sistema desde una perspectiva externa.

El año 2023, la población de pacientes de la clínica dermatológica fue de aproximadamente 287 pacientes. Dado que esta población es más numerosa y diversa en comparación con la población interna de empleados, se ha determinado

que realizar una muestra aleatoria simple de esta población ya que será fundamental para obtener una representación significativa por su importancia para obtener una visión de la experiencia del usuario.

Para determinar el tamaño de la muestra se utilizará la siguiente formula:

ILUSTRACIÓN N° 3: FORMULA DE MUESTREO ALEATORIO SIMPLE

$$n = \frac{Z^2 * p * q * N}{e^2 * (N - 1) + Z^2 * p * q}$$

FUENTE: Elaboración Propia

Donde:

N = 287 (Tamaño de Población)

Z = 1.96 (Nivel de Confianza)

P = 0.5 (Probabilidad de Éxito)

q = 0.5 (Probabilidad de Fracaso)

e = 5 (Nivel de error)

Remplazando:

$$n = \frac{1.96^2 * 287 * 0.5 * 0.5}{5\%^2 * (287 - 1) + 1.96^2 * 0.5 * 0.5} = 165$$

Después de realizar el cálculo utilizando la fórmula de muestreo aleatorio simple se determinó que se necesitaría encuestar a 165 personas. Este resultado proporcionara una base sólida para llevar a cabo un estudio significativo y confiable para el proyecto.

CAPÍTULO II

MARCO TEORICO

CAPÍTULO II**MARCO TEÓRICO****2.1. INTRODUCCIÓN**

En el marco teórico, centrado en las herramientas y tecnologías para el proyecto de desarrollo de un sistema web para una clínica dermatológica, se examinan detalladamente las opciones disponibles en el vasto panorama tecnológico. Este análisis abarca desde los lenguajes de programación hasta los framework de desarrollo. Se destacan consideraciones como el funcionamiento y la escalabilidad del sistema, con el objetivo de garantizar la selección y aplicación adecuadas de las herramientas para crear un sistema web robusto y confiable.

La investigación teórica en este ámbito tiene como propósito proporcionar una base sólida y actualizada que guíe el diseño, la implementación y la evaluación del sistema web para la clínica dermatológica. Integrando conocimientos teóricos con prácticas de vanguardia en el desarrollo de software, se busca crear un sistema tecnológico innovador que mejore significativamente la prestación de servicios de atención médica en el campo dermatológico, ofreciendo una solución eficaz y adaptada a las necesidades específicas del entorno clínico.

2.2. SISTEMA WEB

En la actualidad, los sistemas web son una parte esencial de nuestra vida diaria y de numerosos sectores. Estos sistemas, accesibles a través de navegadores web, facilitan la interacción y gestión de información en línea. Para desarrollar sistemas web efectivos, es crucial comprender principios como la arquitectura de software, la experiencia del usuario y la seguridad de la información.

Jiménez (2013) define que un sistema web es una aplicación informática que se ejecuta en un servidor web y se accede a través de un navegador web. Esta aplicación proporciona servicios y funcionalidades a los usuarios sin necesidad de instalar software adicional en sus dispositivos. Los usuarios pueden interactuar con la aplicación a través de una interfaz web utilizando su navegador favorito (pág. 7).

El concepto de sistemas web es esencial para el proyecto debido a su capacidad para proporcionar accesibilidad, interacción, gestión de información, actualización y escalabilidad, así como seguridad en el manejo de datos. Estos sistemas ofrecen una plataforma versátil que mejora la eficiencia operativa, optimiza la atención médica y garantiza la confidencialidad de la información del paciente, lo que resulta en una mejor experiencia tanto para los profesionales médicos como para los pacientes.

2.3. METODOLOGIA DE DESARROLLO

La metodología de desarrollo es un enfoque sistemático y estructurado para la planificación, diseño, implementación y evaluación de proyectos de software, esencial para garantizar la eficiencia y el éxito en el desarrollo de sistemas web. Al adoptar una metodología adecuada, como Tradicional o Ágil, los equipos de desarrollo pueden gestionar de manera efectiva los recursos, mitigar riesgos y mantener la coherencia a lo largo del ciclo de vida del proyecto. La aplicación de una metodología de desarrollo proporciona una guía clara y coherente para el equipo, promoviendo la colaboración, la transparencia y la entrega oportuna de soluciones de alta calidad que satisfagan las necesidades específicas del cliente y del usuario final.

Lawrence (2006) sostiene que una metodología en desarrollo de software es un marco de trabajo que define un conjunto de prácticas, técnicas y procesos para planificar, organizar, ejecutar y controlar el desarrollo de software. Su objetivo es guiar a los desarrolladores en la creación de software de alta calidad que cumpla con los requisitos del cliente de manera eficiente y eficaz (pág. 25).

TABLA N° 6: COMPARACIÓN DE METODOLOGIAS

ASPECTO	METODOLOGÍA EN CASCADA	METODOLOGÍA AGIL
Planificación detallada	Proporciona una planificación exhaustiva al inicio del proyecto, lo que puede ser beneficioso para proyectos con requisitos estables y bien definidos desde el principio.	La planificación iterativa y flexible permite adaptarse a los cambios en los requisitos, lo que es útil en proyectos donde estos pueden evolucionar o no están completamente definidos inicialmente.
Estructura y secuencia lineal	Ofrece una estructura clara y una secuencia lineal de etapas, lo que puede ser útil para proyectos donde se requiere una clara definición de requisitos y un enfoque paso a paso.	Los ciclos cortos y frecuentes de desarrollo permiten una rápida adaptación a los cambios y una entrega temprana de valor, lo que es beneficioso en proyectos donde se necesita flexibilidad y rápida retroalimentación.
Enfoque en la documentación	Enfatiza la documentación detallada en cada etapa del proyecto, lo que puede ser valioso para proyectos que requieren trazabilidad y un registro exhaustivo de cada paso del proceso.	Se valora más la colaboración cara a cara y el software funcionando sobre la documentación exhaustiva, lo que puede ser más adecuado para proyectos donde la comunicación directa y la entrega rápida de funcionalidades son prioritarias.

ASPECTO	METODOLOGÍA EN CASCADA	METODOLOGÍA AGIL
Gestión de riesgos tempranos	Identifica y aborda los riesgos potenciales al principio del proyecto, lo que puede ayudar a mitigar problemas importantes antes de que ocurran.	La adaptabilidad y la entrega continua permiten abordar los riesgos a medida que surgen, lo que puede ser beneficioso en proyectos donde los riesgos pueden evolucionar o no se conocen completamente desde el principio.
Mayor control y seguimiento	Proporciona un mayor control sobre el progreso del proyecto y permite un seguimiento más detallado de las actividades en cada etapa.	La adaptabilidad y la entrega continua pueden hacer que el progreso del proyecto sea más difícil de prever y controlar, lo que puede ser más desafiante en proyectos donde se requiere una gestión precisa y detallada del progreso.

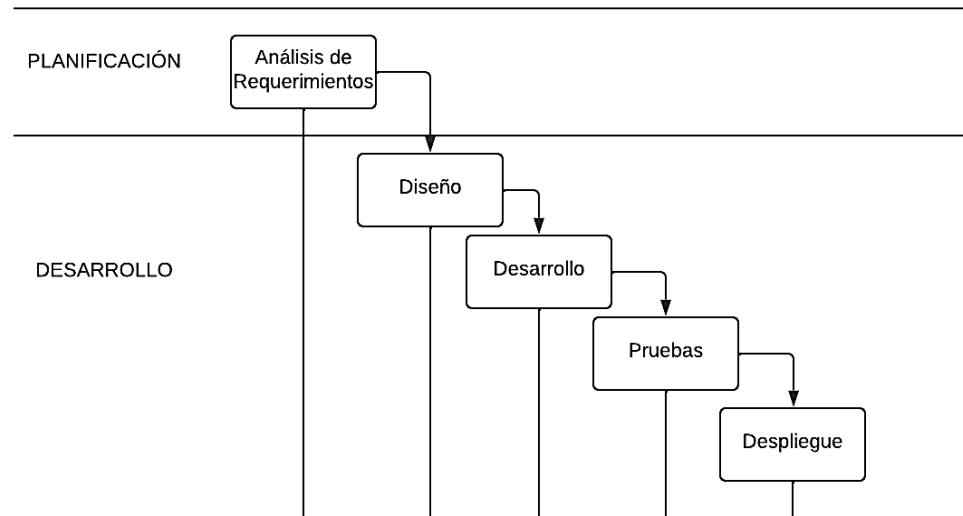
FUENTE: Elaboración Propia, en base a comparación de metodologías.

2.3.1. MODELO TRADICIONAL O EN CASCADA

Según Pressman (2010) el modelo en cascada es una metodología de desarrollo de software que sigue un enfoque secuencial y lineal. Este modelo organiza el proceso de desarrollo en una serie de fases, donde cada fase debe completarse antes de pasar a la siguiente. Las fases típicas del modelo en cascada incluyen requisitos, diseño, implementación, pruebas y mantenimiento. Este enfoque es adecuado para proyectos donde los requisitos son claros y estables desde el principio. El modelo en cascada proporciona una estructura clara y fácil de entender, lo que facilita la planificación y el seguimiento del progreso del proyecto. Sin embargo, puede resultar inflexible ante cambios en los requisitos durante el proceso.

de desarrollo. A pesar de sus limitaciones, el modelo en cascada sigue siendo ampliamente utilizado en proyectos donde los requisitos son bien definidos y el alcance del proyecto es claro desde el principio (pág. 33,34).

ILUSTRACIÓN N° 4: MODELO EN CASCADA



FUENTE: Elaboración propia, en base a metodología en cascada.

El modelo en cascada es la mejor opción para este proyecto debido a la naturaleza bien definida y estable de los requisitos. Ofrece una planificación detallada desde el inicio, una secuencia de etapas lineal y predecible, una documentación exhaustiva y un control riguroso de cambios, lo que garantiza una gestión eficiente y efectiva del proyecto.

2.4. ARQUITECTURA MODELO VISTA CONTROLADOR

La arquitectura Modelo Vista Controlador (MVC) es ampliamente reconocida por su capacidad para separar las preocupaciones en el desarrollo de aplicaciones, lo que facilita la gestión y escalabilidad del software. Esta separación permite a los desarrolladores trabajar simultáneamente en diferentes componentes de la aplicación sin interferencias.

Según Ullman (2013), la arquitectura MVC es efectiva porque permite a los desarrolladores dividir una aplicación en tres componentes interconectados, facilitando así la gestión del desarrollo de software y mejorando la escalabilidad y el mantenimiento de las aplicaciones web (p. 215).

Implementar la arquitectura MVC en el proyecto de la Clínica Dermatológica "DERMA - CAM" es importante para organizar y gestionar eficientemente el código, permitiendo una clara separación entre la lógica de negocios, la interfaz de usuario y el control de flujo.

2.5. LENGUAJE DE PROGRAMACION

Los lenguajes de programación son herramientas fundamentales en informática que permiten a los programadores comunicarse con las computadoras para crear software. Cada lenguaje tiene sus propias reglas sintácticas y semánticas, y se clasifican en diferentes paradigmas según su estilo de programación. Estos lenguajes pueden ser de bajo nivel o de alto nivel, cada uno con sus propias ventajas y aplicaciones específicas. En última instancia, los lenguajes de programación son cruciales para el desarrollo de software en una variedad de campos y aplicaciones.

De acuerdo con Campos, M. & Campos, E. (2023) un lenguaje de programación es un sistema formal que permite a los programadores crear programas para controlar el comportamiento de un sistema informático. Estos lenguajes proporcionan instrucciones y algoritmos que definen las acciones que el sistema debe realizar. El resultado de escribir estas instrucciones se conoce como programa. Cada lenguaje de programación tiene sus propias reglas y estructuras que definen cómo se escriben las instrucciones. Una vez que se ha escrito el programa, debe ser traducido al lenguaje que comprende la computadora, que es el lenguaje máquina, compuesto por ceros y unos. Esta traducción se realiza a través de un compilador. Durante este proceso de traducción, pueden surgir errores en el código fuente del programa que deben ser corregidos (pág.164).

TABLA N° 7: COMPARACIÓN DE LENGUAJES DE PROGRAMACIÓN

ASPECTO	PHP	JAVASCRIPT	PYTHON
Sintaxis	Sintaxis similar a C y Perl.	Sintaxis flexible y dinámica.	Sintaxis limpia y legible.
Paradigma	Imperativo y orientado a objetos.	Multiparadigma: orientado a objetos, funcional y basado en prototipos.	Multiparadigma: orientado a objetos, funcional e imperativo.
Aplicaciones	Principalmente utilizado en desarrollo web para la generación de contenido dinámico.	Ampliamente utilizado en desarrollo web, tanto en el lado del cliente como en el lado del servidor.	Ampliamente utilizado en desarrollo web, análisis de datos, aprendizaje automático, entre otros.
Comunidad y bibliotecas	Gran comunidad y una amplia variedad de bibliotecas y frameworks establecidos.	Gran comunidad y amplia gama de bibliotecas y frameworks disponibles.	Comunidad activa y una gran cantidad de bibliotecas para diversas aplicaciones.
Portabilidad	Principalmente utilizado en entornos web, aunque se puede ejecutar en diferentes sistemas operativos con soporte para PHP.	Principalmente utilizado en navegadores web, aunque también se puede ejecutar en otros entornos a través de Node.js.	Altamente portátil, puede ejecutarse en múltiples plataformas sin cambios significativos en el código.
Facilidad de aprendizaje	La curva de aprendizaje puede ser moderada, especialmente para aquellos nuevos en la programación web.	Relativamente fácil de aprender, especialmente para aquellos familiarizados con la programación web.	Conocido por su facilidad de aprendizaje y sintaxis simple.

ASPECTO	PHP	JAVASCRIPT	PYTHON
Seguridad	Se centra en la seguridad, pero puede ser propenso a vulnerabilidades si no se siguen las mejores prácticas de seguridad.	Limitada en términos de seguridad debido a su ejecución en el navegador y la manipulación del lado del cliente.	Enfoque en la seguridad, pero puede tener vulnerabilidades como cualquier otro lenguaje.

FUENTE: Elaboración Propia, en base a tipos de Lenguajes de Programación.

2.5.1. PHP

PHP es un lenguaje de programación de código abierto que se ha consolidado como una herramienta poderosa y flexible en el ámbito del desarrollo web. Una de sus principales ventajas radica en su carácter gratuito, lo que reduce significativamente los costos asociados al desarrollo y lo hace accesible a desarrolladores de diversos niveles de experiencia. Al ser un lenguaje de código abierto, PHP permite a los desarrolladores modificar y personalizar el código fuente de acuerdo con sus necesidades específicas, facilitando así la creación de soluciones personalizadas y altamente adaptables. Además, la simplicidad y facilidad de uso de PHP lo convierten en un lenguaje accesible y fácil de aprender, especialmente para aquellos desarrolladores que ya poseen experiencia en otros lenguajes de programación como C y Perl.

Según Welling y Thomson (2008), PHP es un lenguaje de script del lado del servidor que se ha diseñado específicamente para el desarrollo web. Destacan que su simplicidad, flexibilidad y capacidad de expansión lo convierten en una herramienta poderosa para la creación de aplicaciones web dinámicas e interactivas (pág. 23).

En el contexto del desarrollo de un sistema de gestión para una clínica dermatológica, PHP es ideal por varias razones clave. La primera es su capacidad para manejar interacciones dinámicas y en tiempo real con los usuarios. PHP permite la creación de interfaces de usuario dinámicas que pueden responder de

manera eficiente y en tiempo real, mejorando significativamente la experiencia del usuario y la eficiencia operativa de la clínica.

2.6. BASE DE DATOS

Una base de datos es un sistema organizado de almacenamiento de datos que permite a las organizaciones almacenar, gestionar y acceder a grandes cantidades de información de manera eficiente. Funciona como un repositorio centralizado donde los datos se organizan en tablas estructuradas que pueden relacionarse entre sí. Esto permite a los usuarios realizar consultas y manipulaciones para recuperar y modificar la información según sea necesario. Las bases de datos se utilizan en una amplia gama de aplicaciones, desde sistemas de gestión de inventario hasta registros médicos y plataformas de comercio electrónico, proporcionando una manera efectiva de gestionar y utilizar datos de manera significativa.

Marqués (2009) sostiene que una base de datos es un conjunto organizado de datos almacenados en memoria externa, diseñado para satisfacer las necesidades de información de una empresa u organización. Antes de la existencia de las bases de datos, se trabajaba con sistemas de archivos descentralizados, donde cada departamento gestionaba sus propios datos con programas de aplicación independientes. Sin embargo, esta descentralización llevaba duplicación de información e inconsistencias. Por otro lado, una base de datos centraliza la información, permitiendo compartir datos entre diferentes departamentos y usuarios. Además, proporciona metadatos que describen los datos almacenados, facilitando la independencia entre la estructura lógica y física de los datos, lo que simplifica la gestión y promueve la consistencia en la información almacenada (pág. 2).

Una base de datos es esencial para este proyecto porque proporciona un medio eficiente y seguro para almacenar, gestionar y acceder a la información de los pacientes, gestionar citas y horarios, y garantizar la seguridad y privacidad de los datos.

2.6.1. BASE DE DATOS SQL

Las bases de datos SQL son sistemas de almacenamiento de datos que utilizan el lenguaje SQL para gestionar y manipular los datos. Estas bases de datos siguen un modelo relacional, donde los datos se organizan en tablas compuestas por filas y columnas.

Según Silva (2018) Una base de datos es un conjunto de datos almacenados de forma organizada para prevenir la duplicación innecesaria de información y posibilitar su acceso desde diversas aplicaciones mediante métodos o funciones predefinidas para insertar, capturar o eliminar datos (pág. 20).

Según Quintana (2014), SQL (Structured Query Language) es un lenguaje de programación desarrollado específicamente para interactuar con Sistemas de Gestión de Bases de Datos Relacionales. Dado que la mayoría de los sistemas actuales son de este tipo y SQL es el lenguaje más comúnmente utilizado en ellos, se puede afirmar con certeza que este lenguaje es predominantemente empleado en los sistemas contemporáneos (pág. 8).

Una base de datos SQL es necesaria para este proyecto porque permite organizar y estructurar los datos eficientemente, asegurar la integridad y coherencia de la información, realizar consultas complejas de manera rápida, manejar transacciones críticas con fiabilidad, garantizar la seguridad y privacidad de los datos sensibles de los pacientes, y ofrecer escalabilidad y facilidad de mantenimiento para gestionar grandes volúmenes de información de forma efectiva.

TABLA N° 8: COMPARACIÓN DE BASE DE DATOS

ASPECTO	MYSQL	MONGODB	POSTGRESQL
Modelo de datos	Relacional (tablas con filas y columnas)	NoSQL (documentos JSON)	Relacional (tablas con filas y columnas)
Escalabilidad	Escalabilidad vertical y horizontal	Escalabilidad horizontal	Escalabilidad vertical y horizontal

ASPECTO	MYSQL	MONGODB	POSTGRESQL
Esquema de datos	Esquema fijo (requiere definición previa)	Esquema flexible (sin esquema fijo)	Esquema fijo (requiere definición previa)
Consultas complejas	Soporta JOINs y consultas SQL complejas	No soporta JOINs, pero permite consultas anidadas y agregaciones poderosas	Soporta JOINs y consultas SQL complejas
Escritura y lectura	Equilibrio entre escritura y lectura	Escritura rápida, lectura eficiente para datos no relacionales	Equilibrio entre escritura y lectura
Flexibilidad de esquema	Menos flexibilidad, requiere definir un esquema	Alta flexibilidad, ideal para datos no estructurados	Menos flexibilidad, requiere definir un esquema
Comunidad y soporte	Gran comunidad y soporte disponible	Gran comunidad y soporte disponible	Gran comunidad y soporte disponible

FUENTE: Elaboración Propia, en base a tipos de Base de Datos.

2.6.2. MySQL

MySQL es un sistema de gestión de bases de datos relacional de código abierto que utiliza SQL para gestionar datos. Es conocido por su alta performance, escalabilidad y seguridad, siendo ampliamente utilizado en aplicaciones web y empresariales para almacenar y gestionar grandes volúmenes de datos de manera eficiente. MySQL permite realizar operaciones como la inserción, consulta, actualización y eliminación de datos.

Natsys (2014) indica que MySQL es un sistema de gestión de bases de datos relacional (RDBMS) de código abierto que utiliza el lenguaje de consulta

estructurado (SQL) para gestionar y manipular bases de datos. Desarrollado inicialmente por MySQL AB y ahora propiedad de Oracle Corporation, MySQL es uno de los sistemas de bases de datos más populares del mundo, ampliamente utilizado en aplicaciones web y de servidor. (pág. 6).

MySQL es una excelente elección para este proyecto debido a su fiabilidad, rendimiento, facilidad de uso, compatibilidad y costo efectivo. Proporcionaría una base de datos sólida y confiable para el sistema de información de la clínica dermatológica, ayudando a garantizar una gestión eficiente de los datos de los pacientes y las operaciones de la clínica.

2.6.3. PHPMyAdmin

PHPMyAdmin es una herramienta gratuita y de código abierto escrita en PHP, diseñada específicamente para la gestión de bases de datos MySQL y MariaDB a través de una interfaz web. Una de las características más destacadas de PHPMyAdmin es su capacidad para simplificar y facilitar tareas complejas de administración de bases de datos, lo que la convierte en una herramienta invaluable para desarrolladores y administradores de bases de datos. Al permitir la ejecución de consultas SQL, la gestión de bases de datos, tablas, usuarios y permisos, así como la importación y exportación de datos en diversos formatos, PHPMyAdmin ofrece una solución integral y accesible para la administración de bases de datos.

Welling y Thomson (2008) señalan que PHPMyAdmin es una herramienta de administración de bases de datos de código abierto que facilita significativamente la gestión de bases de datos MySQL y MariaDB mediante una interfaz web intuitiva, permitiendo a los usuarios realizar tareas administrativas sin necesidad de conocimientos profundos de SQL (pág. 123).

PHPMyAdmin es una buena opción en el proyecto de desarrollo del sistema de gestión de la clínica dermatológica debido a su capacidad para facilitar la administración eficiente y efectiva de la base de datos MySQL. La herramienta permite la creación, modificación y eliminación de bases de datos, tablas, campos e índices a través de una interfaz gráfica intuitiva.

2.7. FIGMA

Figma es una herramienta de diseño colaborativa basada en la web que se ha convertido en una elección popular entre los diseñadores por su capacidad para facilitar la colaboración en tiempo real y la creación de interfaces de usuario de alta calidad. Además, Figma ofrece una amplia gama de características avanzadas, como la creación de prototipos interactivos, la exportación de activos y la integración con otras herramientas de diseño y desarrollo.

Según Penning y Simões (2020), Figma permite a los diseñadores "crear y colaborar en tiempo real, facilitando un flujo de trabajo eficiente y sin interrupciones" (p. 45). Esta capacidad de colaboración es crucial para proyectos que requieren la integración continua de comentarios y ajustes, lo que asegura que el producto final cumpla con las expectativas del cliente y los estándares de calidad.

2.8. VISUAL STUDIO CODE

Visual Studio Code (VS Code) es un entorno de desarrollo integrado (IDE) que ha ganado una amplia aceptación debido a su flexibilidad, extensibilidad y rendimiento. Su interfaz de usuario intuitiva y personalizable facilita la escritura y gestión de código, lo que aumenta significativamente la productividad de los desarrolladores. Una de las mayores ventajas de VS Code es su extensa biblioteca de extensiones que mejoran y amplían sus funcionalidades básicas.

Como subrayan Becker y Schultz (2019), Visual Studio Code ofrece una plataforma robusta y extensible que puede adaptarse a las necesidades específicas de cualquier proyecto de desarrollo, proporcionando herramientas esenciales para la escritura, depuración y gestión de código (p. 45).

La importancia de VS Code para el proyecto de la clínica dermatológica radica en su capacidad para integrar múltiples herramientas y servicios en un único entorno de desarrollo. Esto incluye la posibilidad de conectarse a bases de datos, realizar pruebas unitarias e integradas, y desplegar la aplicación directamente desde el IDE.

Al centralizar todas estas tareas, VS Code no solo facilita el desarrollo, sino que también mejora la coordinación y la colaboración entre los desarrolladores.

2.9. PRUEBAS DE SOFTWARE

El testing, o pruebas de software, es una práctica crucial en el desarrollo de aplicaciones, ya que garantiza la calidad, fiabilidad y rendimiento del producto final. Realizar testing de manera sistemática permite identificar y corregir errores en etapas tempranas del desarrollo, lo que reduce significativamente los costos y el tiempo necesarios para solucionar problemas más adelante.

Según Ian Sommerville (2011), las pruebas de software son "una actividad esencial en el proceso de desarrollo, ya que permiten verificar y validar que el software cumple con sus especificaciones y es adecuado para su propósito" (p. 305). Las pruebas bien diseñadas y ejecutadas pueden descubrir errores y deficiencias que de otro modo pasarían desapercibidos, lo que mejora la calidad general del software.

2.9.1. PHPUnit

PHPUnit es una herramienta fundamental para realizar pruebas unitarias en aplicaciones PHP. Su principal ventaja radica en su capacidad para automatizar el proceso de pruebas, lo que asegura que el código funcione correctamente antes de su despliegue. PHPUnit permite a los desarrolladores escribir pruebas que verifiquen cada parte de su código, identificando errores y garantizando que las nuevas implementaciones no rompan las funcionalidades existentes.

Según el trabajo de Sebastian Bergmann (2014), PHPUnit es una "herramienta esencial para el desarrollo de aplicaciones PHP robustas y fiables", que permite a los desarrolladores mantener altos estándares de calidad mediante pruebas automatizadas y repetibles (p. 123). La automatización de pruebas es crucial para detectar errores en etapas tempranas del desarrollo, minimizando los problemas en producción y aumentando la confianza en el código implementado.

PHPUnit es vital para garantizar la fiabilidad y estabilidad del sistema. Las pruebas unitarias permiten verificar que las funciones críticas, como el manejo de citas, la autenticación de usuarios y la gestión de datos, funcionan correctamente.

2.9.2. Testim

Testim es una herramienta avanzada de automatización de pruebas que se destaca por su capacidad para realizar pruebas funcionales y de regresión de manera eficiente. Una de las principales ventajas de Testim es su uso de inteligencia artificial para crear y mantener pruebas automatizadas, lo que simplifica significativamente el proceso de testing y reduce el tiempo y esfuerzo necesarios para mantener las pruebas a medida que la aplicación evoluciona.

Según el trabajo de Singh y Singh (2021), "las herramientas de automatización de pruebas basadas en inteligencia artificial, como Testim, están revolucionando la forma en que se realizan las pruebas de software al ofrecer capacidades avanzadas de mantenimiento automático de pruebas y generación de informes detallados" (p. 87). Estas características no solo mejoran la eficiencia del proceso de pruebas, sino que también aumentan la cobertura de pruebas y la detección temprana de errores.

En el proyecto de la clínica dermatológica, Testim es una gran herramienta la cual ayudara a realizar pruebas end to end de manera fácil y sencilla, disminuyendo el tiempo que emplean realizar estas. La capacidad de Testim para automatizar las pruebas permite realizar pruebas de regresión rápidas y efectivas cada vez que se introducen nuevas funciones o se realizan cambios en el sistema, asegurando que las actualizaciones no introduzcan nuevos errores.

2.10. MÉTODO ALBRECHT

El método Albrecht para el análisis de puntos función es una técnica estandarizada y ampliamente reconocida para medir la funcionalidad de un sistema de software desde la perspectiva del usuario. Este método es beneficioso porque permite estimar el tamaño del software de manera objetiva y cuantitativa, facilitando así la planificación y gestión del desarrollo de proyectos.

Según Albrecht y Gaffney (1983), el análisis de puntos función ofrece una manera de evaluar la productividad del software y de establecer una base para estimaciones precisas y consistentes del esfuerzo de desarrollo (p. 641).

En el proyecto, el método Albrecht será útil para encontrar los puntos de función del sistema web, para así medir con precisión el esfuerzo necesario para la implementación de este y así poder valorar el coste del proyecto.

CAPITULO III

MARCO PRÁCTICO

MARCO PRÁCTICO**3.1. ANALISIS DE REQUERIMIENTOS**

El análisis de requerimientos es fundamental para el proyecto del sistema de información de la clínica dermatológica. Este análisis ayuda a comprender las necesidades de los usuarios, definir el alcance del proyecto, establecer criterios de éxito, identificar riesgos y facilitar la comunicación. Además, proporciona una guía clara para diseñar, desarrollar e implementar un sistema que cumpla con las expectativas de los usuarios y contribuya al éxito del negocio. Los requerimientos se identificaron a través de una entrevista de toma de requerimientos realizada al propietario de la clínica (ver Anexo II).

El objetivo del análisis de requerimientos es comprender completamente las necesidades, expectativas y restricciones de un sistema de información. Esto implica identificar y documentar de manera precisa y detallada qué funciones y características debe tener el sistema, quiénes serán los usuarios finales, qué problemas debe resolver y cómo se integrará con los procesos existentes de la organización. El análisis de requerimientos establece las bases para el diseño, desarrollo e implementación exitosa del sistema, asegurando que cumpla con las necesidades y expectativas del cliente y de los usuarios finales.

3.1.1. Identificación de actores

Se identificaron los actores clave que estarán involucrados en el proceso de desarrollo del sistema según los requerimientos que se obtuvieron en la entrevista con la dueña de la clínica.

TABLA N° 9: LISTA DE ACTORES

ACTOR	DESCRIPCIÓN
Usuario Administrador	Responsable de gestionar y mantener el sistema de información, garantizando su correcto funcionamiento, seguridad y cumplimiento de los objetivos organizacionales.
Usuario Médico	Responsable de gestionar la atención médica de los pacientes, con acceso a herramientas y datos necesarios para brindar una atención de calidad, incluyendo información sobre pacientes, citas, tratamientos y resultados de pruebas.
Usuario Paciente	Usuario con acceso a su información personal, historial médico, citas médicas, recetas y tratamientos, y detalles sobre facturación
Usuario Asistente	Responsable de apoyar en la gestión diaria de las actividades administrativas y clínicas, con acceso a información relacionada con los pacientes, citas programadas, historial de citas, recursos clínicos, documentos administrativos y comunicación interna.

FUENTE: Elaboración Propia en base a Entrevista para la toma de requerimientos
 (Ver Anexo II).

3.1.2. Requerimientos Funcionales

Los requerimientos funcionales son fundamentales para asegurar que el sistema de información cumpla con las necesidades y expectativas de los usuarios, guiar el proceso de desarrollo y garantizar el éxito del proyecto en términos de funcionalidad y usabilidad.

TABLA N° 10: REQUERIMIENTOS FUNCIONALES DE USUARIO ADMINISTRADOR

Requerimientos Funcionales De Usuario Administrador
El sistema permitirá al usuario administrador crear, editar y eliminar cuentas de usuario para el personal médico, asistentes, recepcionistas y otros empleados de la clínica, asignando roles y permisos según sea necesario.
El sistema permitirá al usuario administrador supervisar y gestionar el calendario de citas de la clínica, incluyendo la capacidad de programar, modificar y cancelar citas, así como visualizar la disponibilidad de horarios de los médicos y recursos clínicos.
El sistema permitirá al usuario administrador gestionar el proceso de facturación de los servicios médicos prestados, registrar pagos recibidos, enviar facturas a compañías de seguros y generar informes financieros para la contabilidad de la clínica.
El sistema permitirá al usuario administrador gestionar el inventario de suministros médicos y equipos de la clínica
El sistema permitirá al usuario administrador generar informes y análisis sobre diferentes aspectos del funcionamiento de la clínica, como la asistencia de pacientes y los ingresos

FUENTE: Elaboración Propia en base a Entrevista para la toma de requerimientos
(Ver Anexo II).

TABLA N° 11: REQUERIMIENTOS FUNCIONALES DE USUARIO MÉDICO

Requerimientos Funcionales De Usuario Médico
El sistema permitirá al usuario médico acceder al historial médico completo de los pacientes, incluyendo diagnósticos previos, tratamientos realizados y notas de progreso.
El sistema permitirá al usuario médico programar y gestionar citas con los pacientes de manera eficiente, con funcionalidades para ver la disponibilidad de horarios, programar nuevas citas y modificar o cancelar citas existentes.
El sistema permitirá al usuario médico la prescripción de medicamentos y tratamientos para los pacientes, con herramientas para seleccionar medicamentos adecuados, establecer dosis, frecuencias y duración del tratamiento, y registrar las prescripciones en el historial médico del paciente.
El sistema permitirá al usuario médico registrar notas de progreso y observaciones clínicas durante las consultas con los pacientes, para documentar la evolución de su condición médica y los planes de tratamiento recomendados.
El sistema permitirá al usuario médico generar informes clínicos completos y detallados para los pacientes, que incluyan información sobre diagnósticos, tratamientos, recomendaciones de seguimiento y cualquier otra información relevante para la atención médica del paciente.

FUENTE: Elaboración Propia en base a Entrevista para la toma de requerimientos
(Ver Anexo II).

TABLA N° 12: REQUERIMIENTOS FUNCIONALES DE USUARIO PACIENTE

Requerimientos Funcionales De Usuario Paciente
El sistema permitirá al usuario paciente se registre en el sistema proporcionando información personal básica, como nombre, fecha de nacimiento, dirección y datos de contacto.
El sistema permitirá al usuario paciente solicitar y programar citas médicas de manera online, seleccionando fechas y horas disponibles y recibiendo confirmaciones automáticas por correo electrónico o mensajes de texto.
El sistema permitirá al usuario paciente acceder a su historial médico electrónico, incluyendo diagnósticos previos y tratamientos realizados
El sistema permitirá al usuario paciente gestionar sus datos personales incluyendo la opción de actualizar información de contacto.

FUENTE: Elaboración Propia en base a Entrevista para la toma de requerimientos
(Ver Anexo II).

TABLA N° 13: REQUERIMIENTOS FUNCIONALES DE USUARIO ASISTENTE

Requerimientos Funcionales De Usuario Asistente
El sistema permitirá al usuario asistente programar, modificar y cancelar citas para los pacientes, incluyendo la asignación de fechas y horarios disponibles, la confirmación de citas y la gestión de listas de espera.
El sistema permitirá al usuario asistente registrar nuevos pacientes en el sistema, incluyendo la recopilación de información personal básica, la creación de perfiles de pacientes y la asignación de números de expediente médico.
El sistema permitirá al usuario asistente acceder al historial médico electrónico de los pacientes para revisar información relevante, como diagnósticos previos, tratamientos realizados, resultados de pruebas y notas de progreso registradas por los médicos.
El sistema permitirá al usuario asistente generar informes y estadísticas sobre la actividad de la clínica, incluyendo el número de citas programadas, la tasa de asistencia de pacientes, los tiempos de espera y cualquier otro dato relevante para la gestión y mejora del servicio.

FUENTE: Elaboración Propia en base a Entrevista para la toma de requerimientos
(Ver Anexo II)

3.1.3. Requerimientos No Funcionales

Los requerimientos no funcionales son esenciales por varias razones clave. Estos especifican criterios de calidad, limitaciones tecnológicas, cumplimiento normativo, experiencia del usuario y rendimiento del sistema. Aseguran que el sistema sea eficaz, viable tecnológicamente, legalmente conforme, fácil de usar, y capaz de manejar la carga de trabajo esperada.

TABLA N° 14: REQUERIMIENTOS NO FUNCIONALES DE USUARIO ADMINISTRADOR

Requerimientos Funcionales De Usuario Administrador
El sistema debe garantizar que el usuario administrador tenga acceso seguro a funciones críticas del sistema, como la gestión de usuarios y la configuración del sistema, mediante autenticación y controles de acceso adecuados.
El sistema debe estar disponible en todo momento para el usuario administrador, minimizando el tiempo de inactividad planificado y no planificado mediante el uso de redundancia, tolerancia a fallos y procedimientos de recuperación ante desastres.

Requerimientos Funcionales De Usuario Administrador
El sistema debe ser capaz de manejar un número creciente de usuarios y datos a medida que la clínica dermatológica crezca, asegurando que el rendimiento no se vea afectado y que el sistema pueda escalar horizontal o verticalmente según sea necesario.
El usuario administrador debe tener la capacidad de realizar tareas de administración del sistema de manera eficiente, como realizar copias de seguridad y restauraciones de datos, monitorear el rendimiento del sistema y aplicar actualizaciones de software.
A pesar de tener privilegios de administrador, el usuario debe encontrar la interfaz del sistema intuitiva y fácil de usar, con funciones bien organizadas y documentación clara para facilitar la administración del sistema.

FUENTE: Elaboración Propia en base a Entrevista para la toma de requerimientos
(Ver Anexo II).

TABLA N° 15: REQUERIMIENTOS NO FUNCIONALES DE USUARIO MÉDICO

Requerimientos Funcionales De Usuario Médico
El sistema debe responder de manera rápida y eficiente a las consultas y acciones realizadas por el usuario médico, garantizando tiempos de respuesta mínimos para evitar retrasos en la atención al paciente.
El sistema debe ser confiable y estar disponible en todo momento para que el usuario médico pueda acceder a la información clínica y realizar registros de manera continua, sin interrupciones que afecten la atención al paciente.
El sistema debe garantizar la seguridad y confidencialidad de la información médica de los pacientes, implementando medidas de protección de datos para prevenir accesos no autorizados y asegurar la integridad de los registros médicos.
El sistema debe garantizar la seguridad y confidencialidad de la información médica de los pacientes, implementando medidas de protección de datos para prevenir accesos no autorizados y asegurar la integridad de los registros médicos.
Dado que los médicos pueden necesitar acceder al sistema desde dispositivos móviles, el sistema debe ser responsive y ofrecer una experiencia de usuario óptima en dispositivos móviles.

FUENTE: Elaboración Propia en base a Entrevista para la toma de requerimientos
(Ver Anexo II).

TABLA N° 16: REQUERIMIENTOS NO FUNCIONALES DE USUARIO PACIENTE

Requerimientos Funcionales De Usuario Paciente
La interfaz del sistema debe ser fácil de usar y comprensible para los pacientes, con instrucciones claras y un diseño intuitivo que facilite la navegación y la interacción con el sistema.
El sistema debe proteger la privacidad y confidencialidad de los datos de los pacientes, garantizando que la información médica y personal solo sea accesible por el paciente y el personal autorizado de la clínica dermatológica.
El sistema debe estar disponible en todo momento para que los pacientes puedan acceder a su información médica, programar citas y recibir comunicaciones importantes de la clínica.
El sistema debe ser compatible con una variedad de dispositivos y navegadores webs móviles para garantizar una experiencia óptima del usuario.
El sistema debe responder de manera rápida y eficiente a las solicitudes de los pacientes, con tiempos de carga mínimos y procesos de reserva de citas fluidos para mejorar la experiencia del usuario.

FUENTE: Elaboración Propia en base a Entrevista para la toma de requerimientos
(Ver Anexo II).

TABLA N° 17: REQUERIMIENTOS NO FUNCIONALES DE USUARIO ASISTENTE

Requerimientos Funcionales De Usuario Asistente
El sistema debe ser eficiente en el manejo de tareas administrativas, como la gestión de citas, la actualización de registros de pacientes y la coordinación de recursos médicos, para permitir una operación fluida y sin demoras en la clínica dermatológica.
La interfaz del sistema debe ser fácil de aprender para los asistentes, con una curva de aprendizaje baja que les permita dominar rápidamente las funciones y procedimientos necesarios para realizar sus tareas de manera efectiva.
El sistema debe tener un rendimiento óptimo, garantizando tiempos de respuesta rápidos y procesos ágiles para evitar retrasos en la gestión de citas, la búsqueda de información de pacientes y otras tareas críticas para el funcionamiento de la clínica.
Debe haber medidas de seguridad implementadas en el sistema para garantizar que solo los asistentes autorizados puedan acceder a funciones y datos sensibles, como la programación de citas y la información médica de los pacientes.

FUENTE: Elaboración Propia en base a Entrevista para la toma de requerimientos
(Ver Anexo II).

TABLA N° 18: REQUERIMIENTOS NO FUNCIONALES DEL SISTEMA

Requerimientos Funcionales Del Sistema
El sistema debe implementar medidas de seguridad para proteger la integridad, confidencialidad y disponibilidad de los datos. Esto incluye el cifrado de datos, autenticación de usuarios y control de acceso
El sistema debe estar disponible en todo momento, con un tiempo de inactividad planificado mínimo para mantenimiento y actualizaciones.
El sistema debe ser escalable, lo que significa que debe poder crecer y adaptarse fácilmente a medida que aumentan los usuarios, datos y demandas de rendimiento.
La interfaz del usuario debe ser intuitiva y fácil de usar, con un diseño claro y navegación sencilla. Debe ser accesible para usuarios de diferentes niveles de habilidad y experiencia.
El sistema debe ser compatible con una variedad de dispositivos y plataformas, incluyendo computadoras de escritorio, dispositivos móviles y diferentes navegadores web.

FUENTE: Elaboración Propia en base a Entrevista para la toma de requerimientos
(Ver Anexo II).

3.1.4. Definición de Procesos

- **Usuario Administrador**
 - **Gestión de Usuarios:** El usuario administrador puede encargarse de la creación, modificación y eliminación de cuentas de usuario para otros empleados del sistema, como médicos, asistentes y personal de recepción. Esto implica asignar roles y permisos apropiados a cada usuario.
 - **Configuración del Sistema:** El administrador puede configurar parámetros del sistema, como horarios de atención, días festivos, tipos de consultas disponibles, tarifas de servicios médicos y otros aspectos relacionados con la operación de la clínica.
 - **Control de Acceso y Seguridad:** Es responsabilidad del administrador asegurar que el sistema esté protegido contra accesos no autorizados y posibles vulnerabilidades de seguridad. Esto incluye la configuración de políticas de contraseñas y gestión de permisos de acceso.
 - **Generación de Informes:** El usuario administrador puede generar informes y análisis sobre el rendimiento y la eficiencia del sistema, incluyendo estadísticas de

citas, uso de recursos, cumplimiento de objetivos y otros indicadores clave de rendimiento.

- **Usuario Médico**

- **Registro de Pacientes:** El médico puede acceder al sistema para registrar nuevos pacientes, ingresando información básica como nombre, fecha de nacimiento, antecedentes médicos y detalles de contacto.
- **Gestión y Consulta de Historias Clínicas:** El médico puede crear y acceder a las historias clínicas de los pacientes para revisar su historial médico, incluyendo diagnósticos previos y tratamientos anteriores.
- **Programación de Citas:** El médico puede programar citas con los pacientes para consultas, exámenes dermatológicos o procedimientos específicos, coordinando horarios disponibles y asignando intervalos de tiempo adecuados.
- **Realización de Consultas y Exámenes:** Durante las consultas, el médico puede utilizar el sistema para registrar observaciones, diagnosticar enfermedades de la piel, prescribir tratamientos y realizar pruebas adicionales si es necesario.
- **Gestión de Tratamientos:** El médico puede registrar los tratamientos recetados a los pacientes, incluyendo medicamentos, cremas o procedimientos dermatológicos específicos, así como establecer pautas de seguimiento y revisión.
- **Consulta de Agenda y Recordatorios:** El médico puede consultar su agenda de citas programadas y recibir recordatorios automáticos sobre próximas consultas o tareas pendientes relacionadas con pacientes específicos.

- **Usuario Paciente**

- **Registro y Actualización de Datos Personales:** El paciente puede acceder al sistema para registrar o actualizar su información personal, incluyendo datos de contacto, historial médico, alergias, medicamentos actuales y cualquier otra información relevante para su atención dermatológica.
- **Obtener Citas:** El paciente puede obtener citas con médicos dermatólogos a través del sistema, seleccionando fechas y horarios disponibles que se ajusten a su agenda y preferencias.
- **Consulta de Historial Médico:** El paciente puede acceder a su historial médico en el sistema para revisar diagnósticos anteriores, tratamientos realizados, resultados de pruebas y recomendaciones médicas previas.

- **Recepción de Recordatorios y Confirmaciones:** El paciente puede recibir recordatorios automáticos sobre citas programadas, así como confirmaciones de citas solicitadas a través del sistema, lo que ayuda a mantener una comunicación efectiva y a evitar olvidos.
- **Acceso a Resultados de Pruebas y Tratamientos:** Una vez realizadas las pruebas dermatológicas o los tratamientos prescritos, el paciente puede acceder al sistema para consultar los resultados y seguir el progreso de su tratamiento.
- **Usuario Asistente**
- **Registro de Pacientes:** El asistente puede encargarse de registrar nuevos pacientes en el sistema, ingresando información básica como nombre, edad, datos de contacto y motivo de la consulta.
- **Programación de Citas:** El asistente puede gestionar el calendario de citas de los médicos dermatólogos, asignando horarios disponibles a los pacientes según sus solicitudes y la disponibilidad de los profesionales.
- **Preparación de Documentación:** El asistente puede generar y mantener documentos médicos y administrativos, como formularios de consentimiento, facturas, informes de seguros y otros registros necesarios para el funcionamiento de la clínica.
- **Gestión de Pagos:** El asistente puede realizar el seguimiento de los pagos de los pacientes por servicios médicos, registrando los pagos recibidos.
- **Soporte Administrativo:** El asistente puede proporcionar soporte administrativo a otros miembros del equipo médico y de la clínica, ayudando con tareas como la gestión de archivos, la preparación de informes y la respuesta a correos electrónicos.

3.2. DISEÑO DEL SISTEMA

El diseño del sistema es esencial para el proyecto, ya que clarifica la visión, asegura la alineación con los requerimientos, identifica problemas potenciales, facilita el desarrollo eficiente y permite la evaluación temprana. Proporciona una guía clara para la implementación exitosa del proyecto, asegurando que la solución final cumpla con las necesidades del cliente y los usuarios.

El objetivo del diseño del sistema es crear una representación detallada y estructurada de cómo funcionará y se verá el sistema una vez completado. Esto implica definir la arquitectura del software, los componentes principales, las interfaces de usuario y las interacciones entre ellos. El diseño del sistema tiene como objetivo proporcionar una guía clara para el desarrollo e implementación del software, asegurando que cumpla con los requerimientos del cliente y las expectativas de los usuarios finales. Además, busca identificar y resolver problemas potenciales antes de la implementación, garantizando la eficiencia y la calidad del producto final.

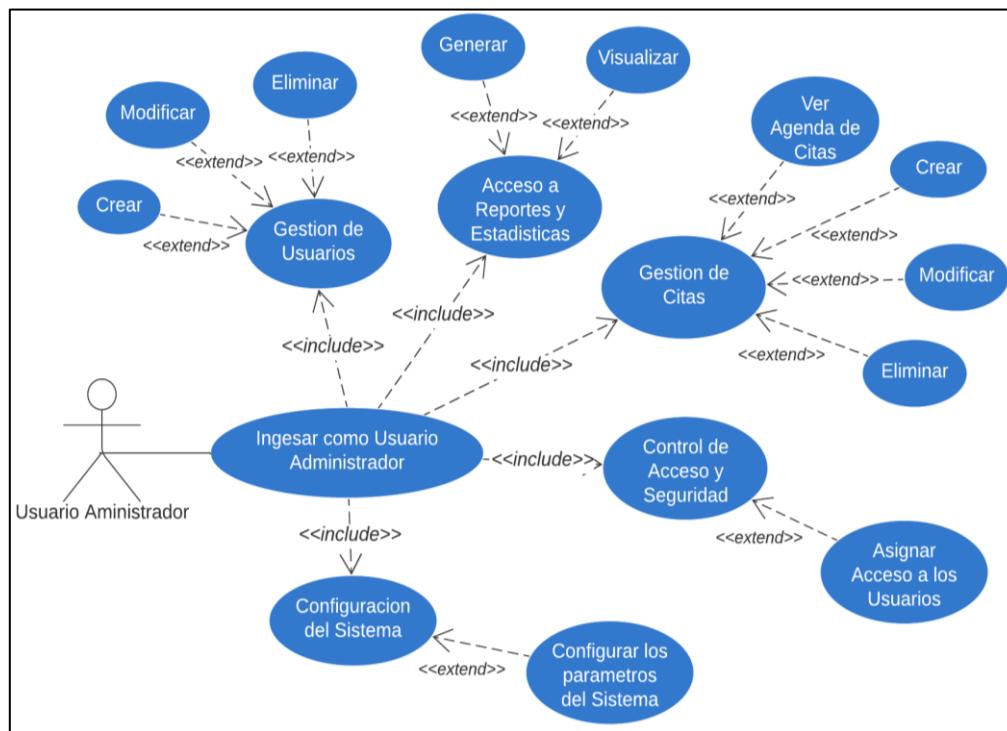
3.2.1. Modelo de Casos de Uso

Los modelos de casos de uso son útiles en el desarrollo de software porque proporcionan una forma estructurada y comprensible de capturar los requisitos funcionales del sistema desde la perspectiva del usuario.

3.2.1.1. Casos de Uso Usuario Administrador

En el caso de uso del usuario administrador se puede observar sus interacciones con el sistema, como la gestión de usuarios, citas, acceso a reportes, etc. Al ser el administrador del sistema tendrá acceso a todas las funcionalidades.

ILUSTRACIÓN N° 5: CASOS DE USO DEL USUARIO ADMINISTRADOR

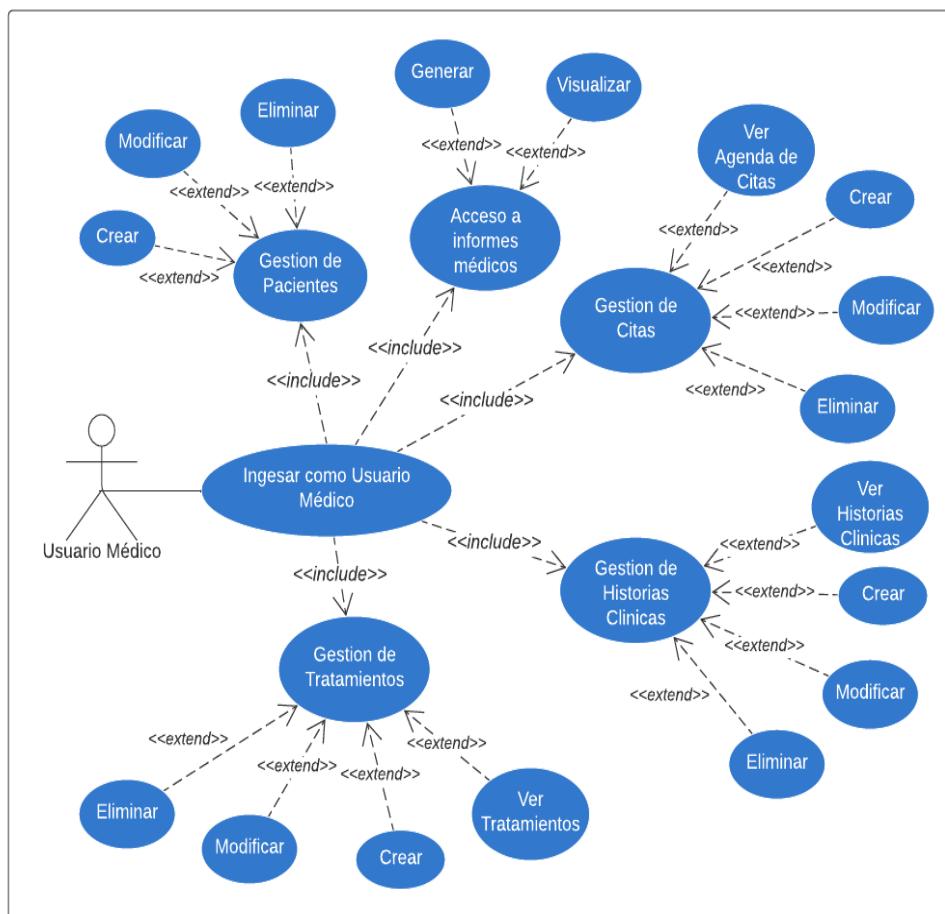


FUENTE: Elaboración propia, en base a Requerimientos Funcionales.

3.2.1.2. Casos de Uso Usuario Médico

En el caso de uso del usuario médico se puede observar las interacciones que tendrá este con el sistema desde gestionar pacientes, citas, historiales, entre otros. Esto ayuda a entender de mejor manera las funciones que el medico puede realizar dentro el sistema.

ILUSTRACIÓN N° 6: CASOS DE USO DEL USUARIO MÉDICO

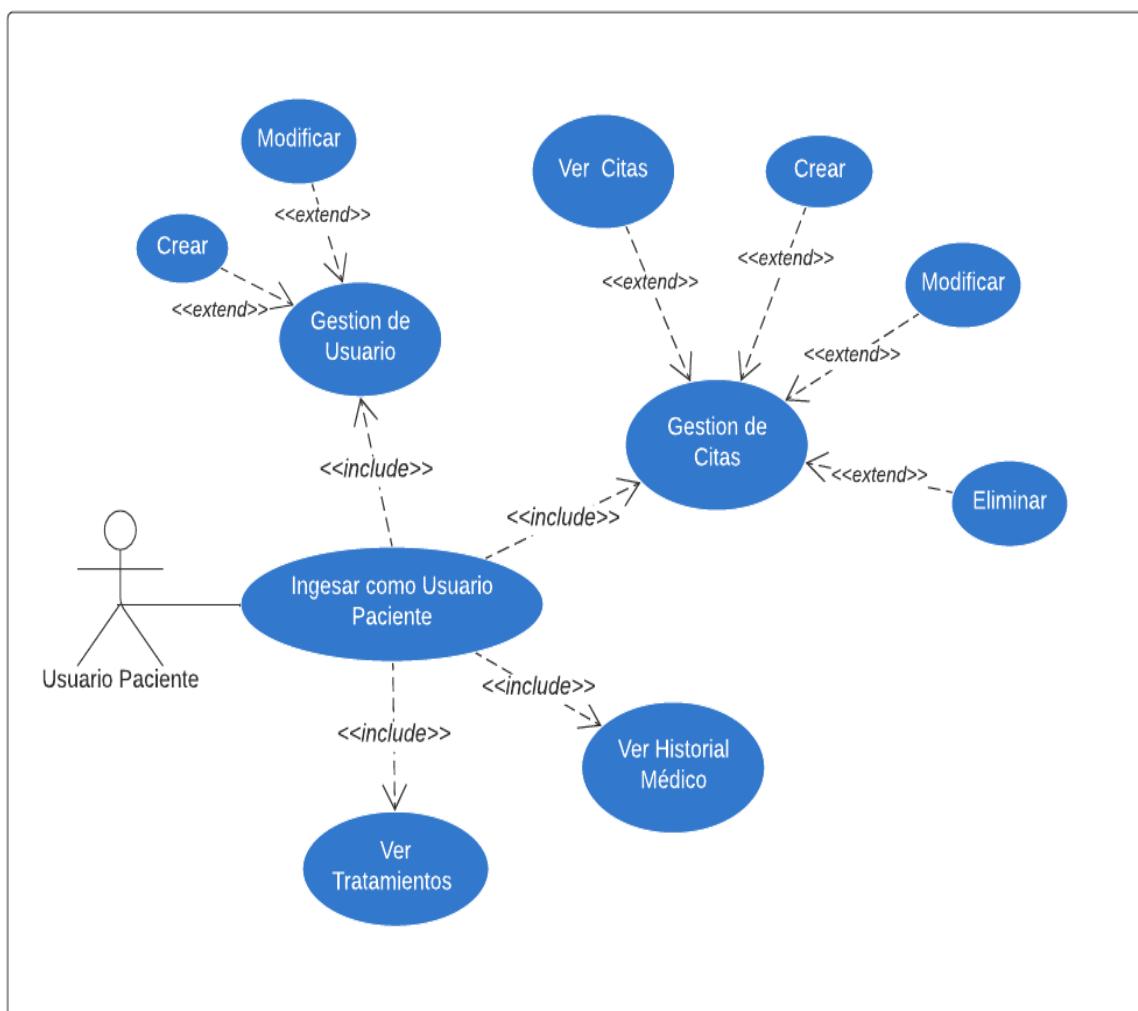


FUENTE: Elaboración propia, en base a Requerimientos Funcionales.

3.2.1.3. Casos de Uso Usuario Paciente

En el caso de uso del usuario paciente se puede observar las interacciones que tendrá este con el sistema, en este caso serán más limitadas ya que solo tendrá acceso a funciones que le competen, como gestionar su información, sus citas, y solo podrá ver sus historiales y tratamientos.

ILUSTRACIÓN N° 7: CASOS DE USO DEL USUARIO PACIENTE

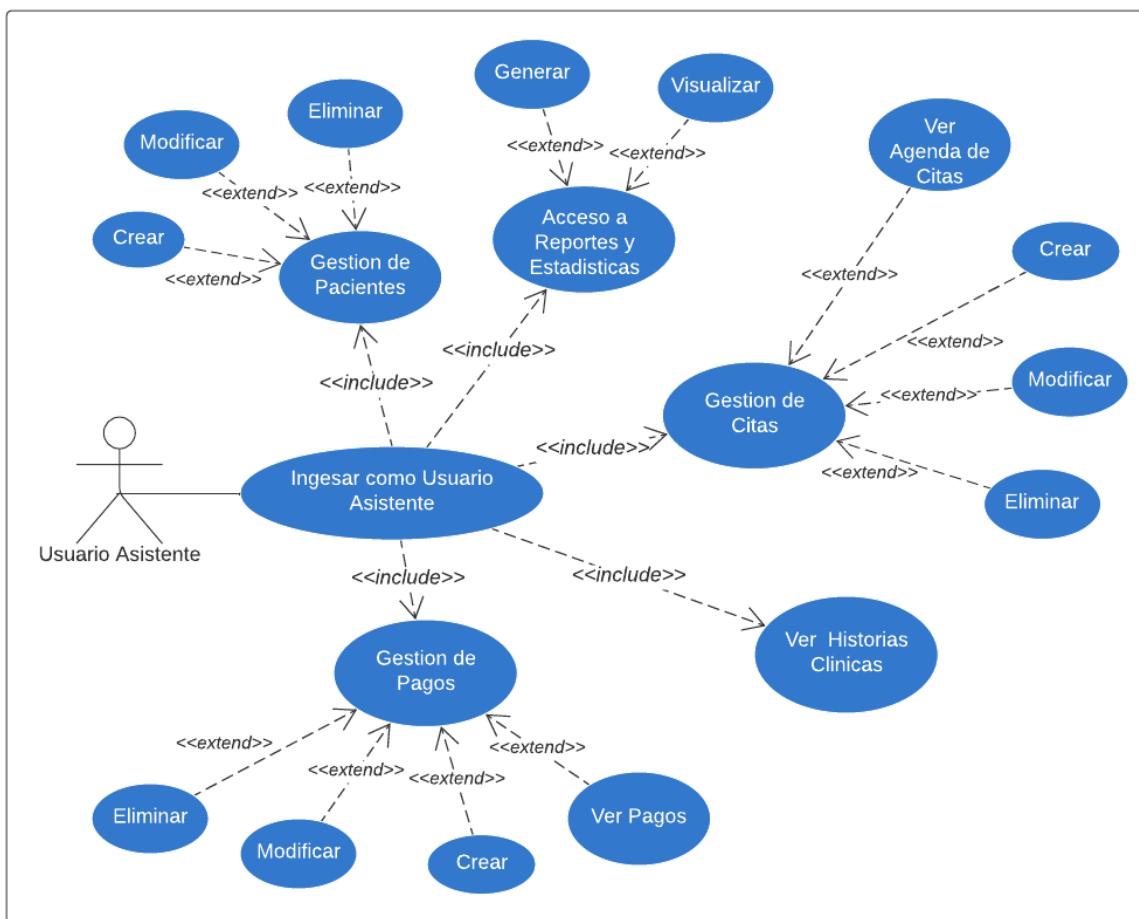


FUENTE: Elaboración propia, en base a Requerimientos Funcionales.

3.2.1.4. Casos de Uso Usuario Asistente

En el caso de uso del usuario Asistente se puede observar las interacciones que tendrá este con el sistema desde gestionar pacientes, citas, historiales, entre otros. Esto ayuda a entender de mejor manera las funciones que el asistente puede realizar dentro el sistema.

ILUSTRACIÓN N° 8: CASOS DE USO DEL USUARIO ASISTENTE



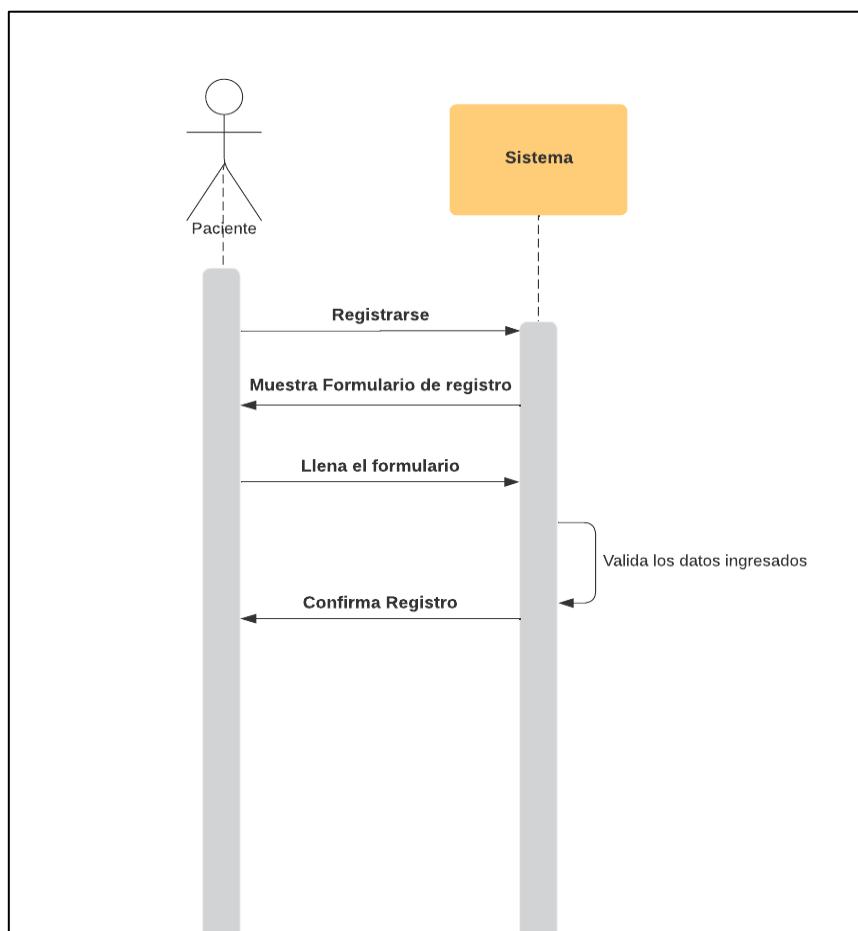
FUENTE: Elaboración propia, en base a Requerimientos Funcionales.

3.2.2. Diagrama de Secuencia

El diagrama de secuencia es una herramienta para visualizar y planificar las interacciones entre los componentes de un sistema a lo largo del tiempo. Facilita la comprensión del flujo de control y datos dentro del sistema, ayuda en la identificación de responsabilidades y asegura que todos los requisitos se cumplan correctamente a través de interacciones bien definidas.

El diagrama de secuencia ayudara a entender las interacciones que el paciente requiere para registrarse.

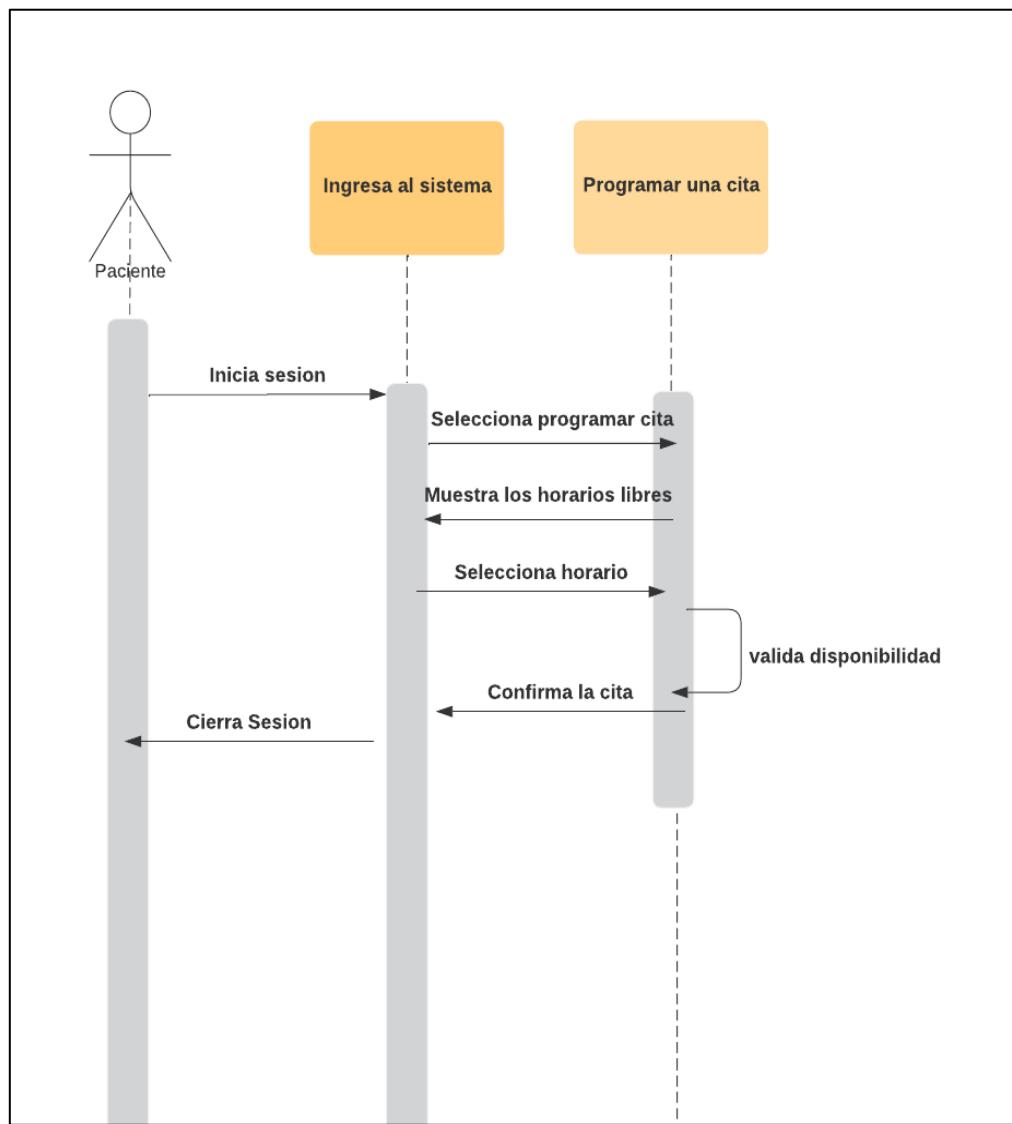
ILUSTRACIÓN N° 9: DIAGRAMA DE SECUENCIA REGISTRARSE



FUENTE: Elaboración propia, en base a Requerimientos Funcionales.

El siguiente diagrama detalla la secuencia en la programación de citas. Como se puede observar interviene el paciente el cual debe iniciar sesión para realizar los pasos siguientes.

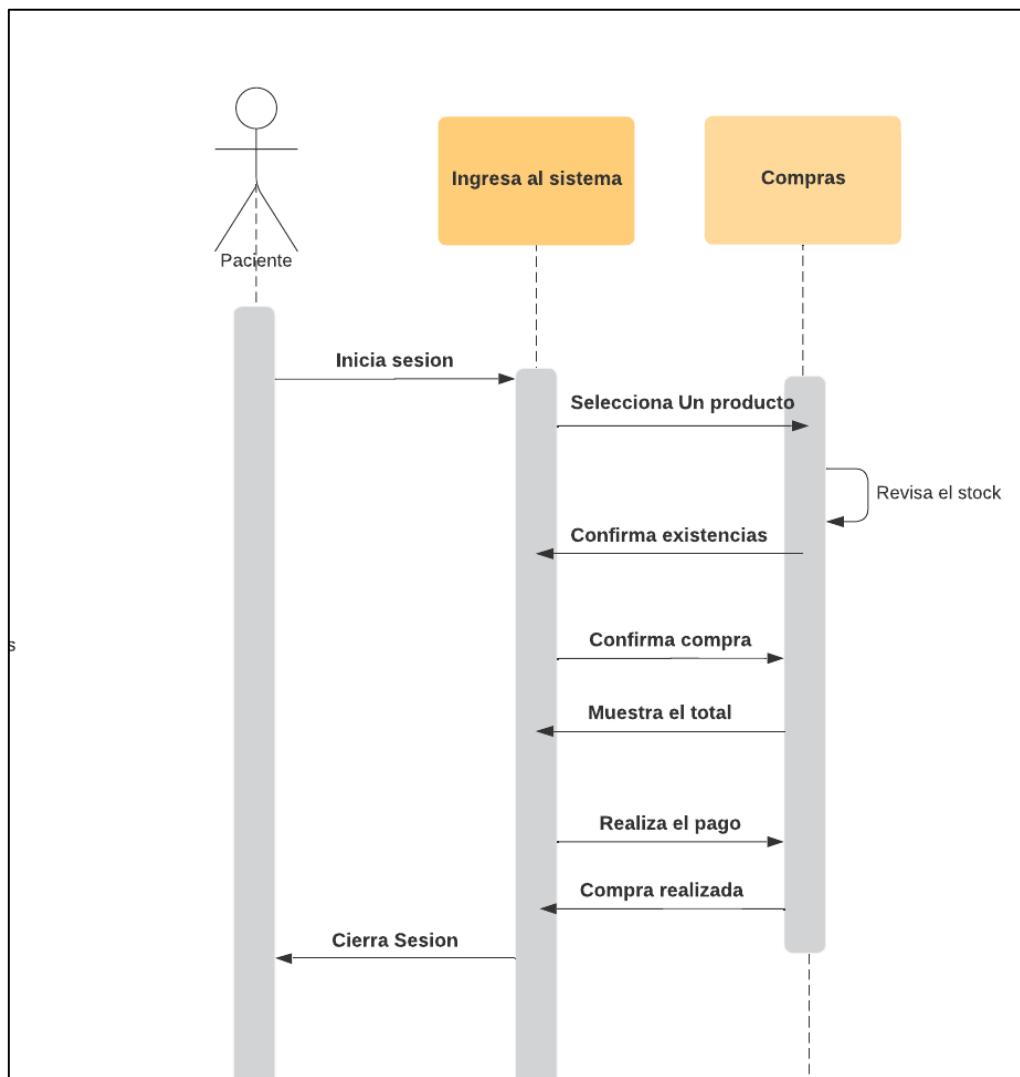
ILUSTRACIÓN N° 10: DIAGRAMA DE SECUENCIA RESERVA DE CITA



FUENTE: Elaboración propia, en base a Requerimientos Funcionales.

El siguiente diagrama detalla la secuencia para realizar una compra. Como se puede observar interviene el paciente el cual debe iniciar sesión para realizar los pasos siguientes.

ILUSTRACIÓN N° 11: DIAGRAMA DE SECUENCIA REALIZAR COMPRA

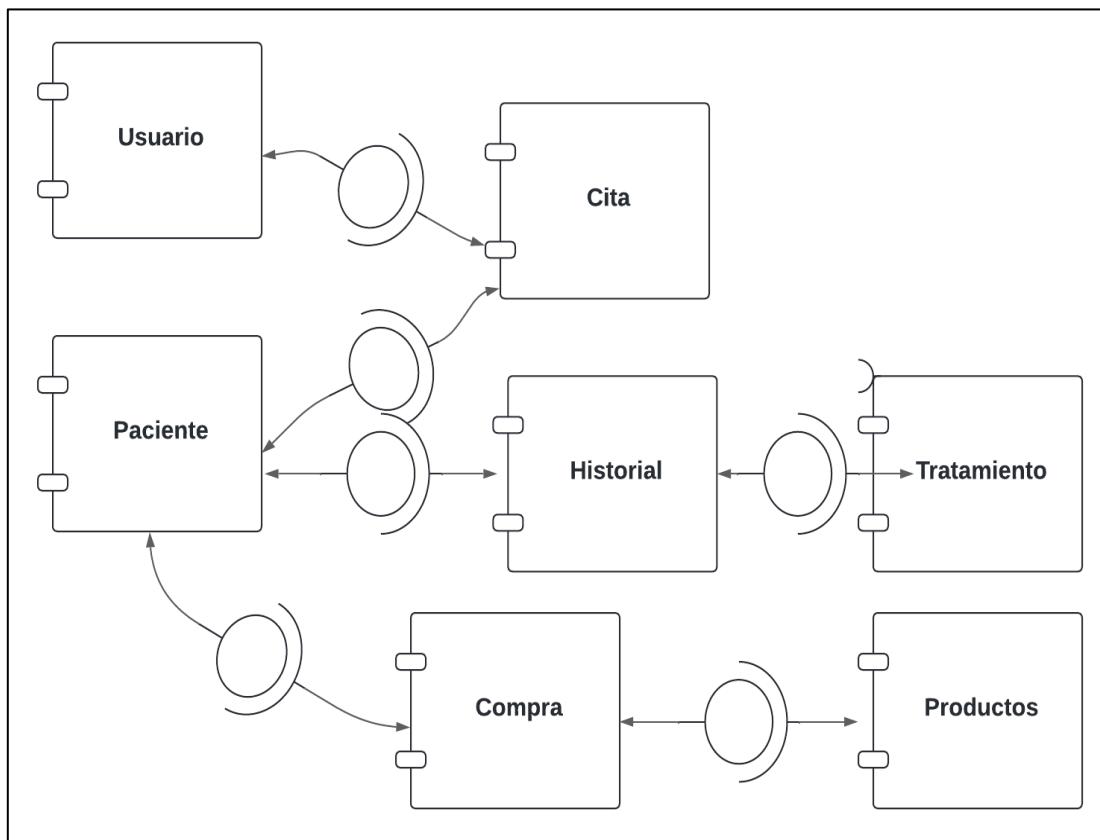


FUENTE: Elaboración propia, en base a Requerimientos Funcionales.

3.2.3. Diagrama de Componentes

En la siguiente ilustración se pueden identificar los cinco módulos definidos a partir del análisis de requerimientos. Además, se ha considerado a los usuarios en dos categorías: el personal de la clínica (que incluye administrador, médico y asistente) y los pacientes.

ILUSTRACIÓN N° 12: DIAGRAMA DE COMPONENTES



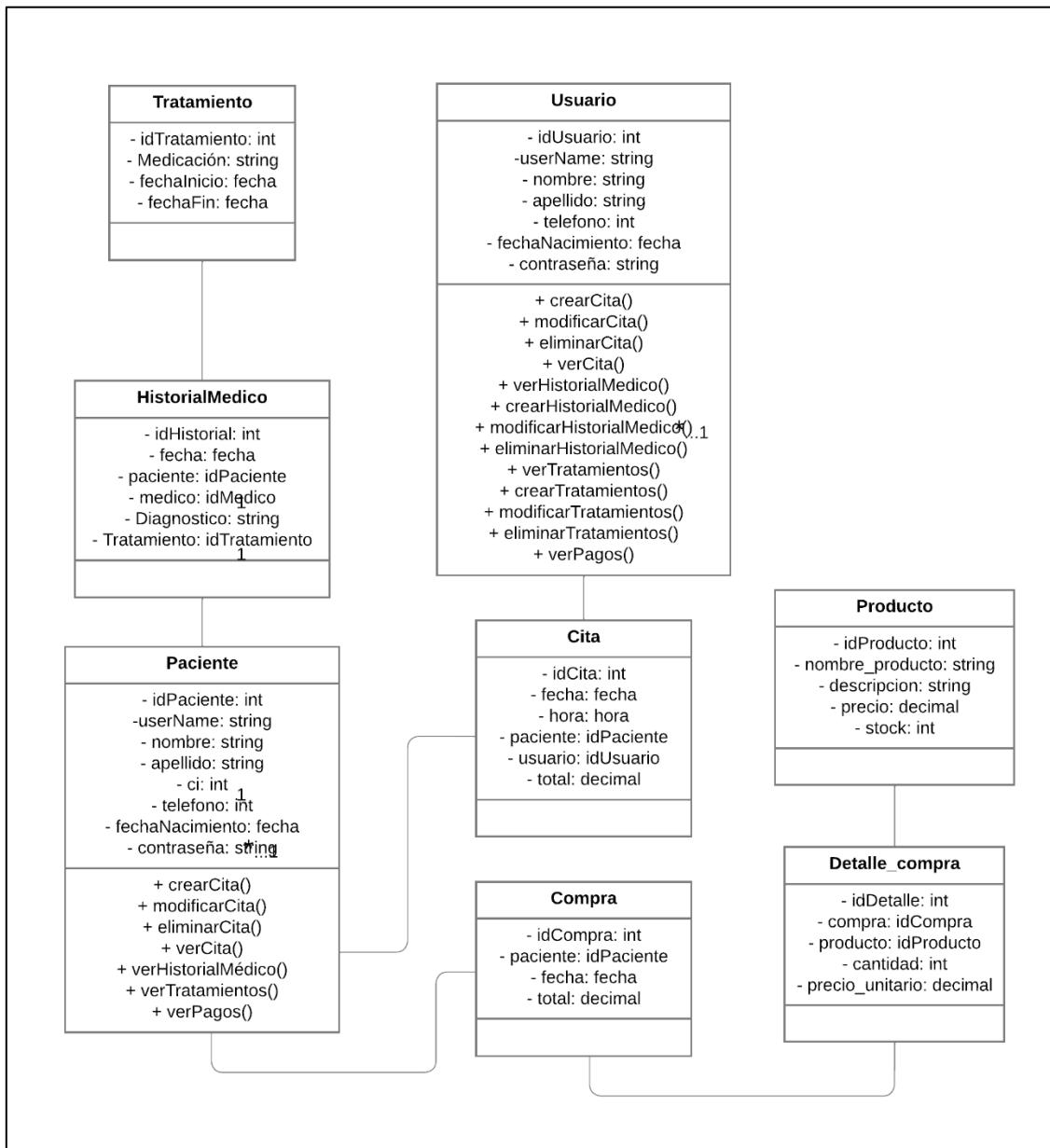
FUENTE: Elaboración propia, en base a Requerimientos Funcionales.

3.2.4. Diagrama de Clases

En la Ilustración 13, que se presenta a continuación, se pueden observar las clases identificadas a través del análisis de requerimientos. Estas clases están organizadas de manera que corresponden a los módulos previamente definidos en el diagrama de componentes. Cada clase representa una parte integral del sistema y su relación con otros componentes, proporcionando una visión clara de la

estructura y la funcionalidad del sistema propuesto. Esta organización facilita la comprensión de cómo cada módulo interactúa dentro del sistema y cómo se logra la integración de todas las partes para cumplir con los objetivos del proyecto.

ILUSTRACIÓN N° 13: DIAGRAMA DE CLASES

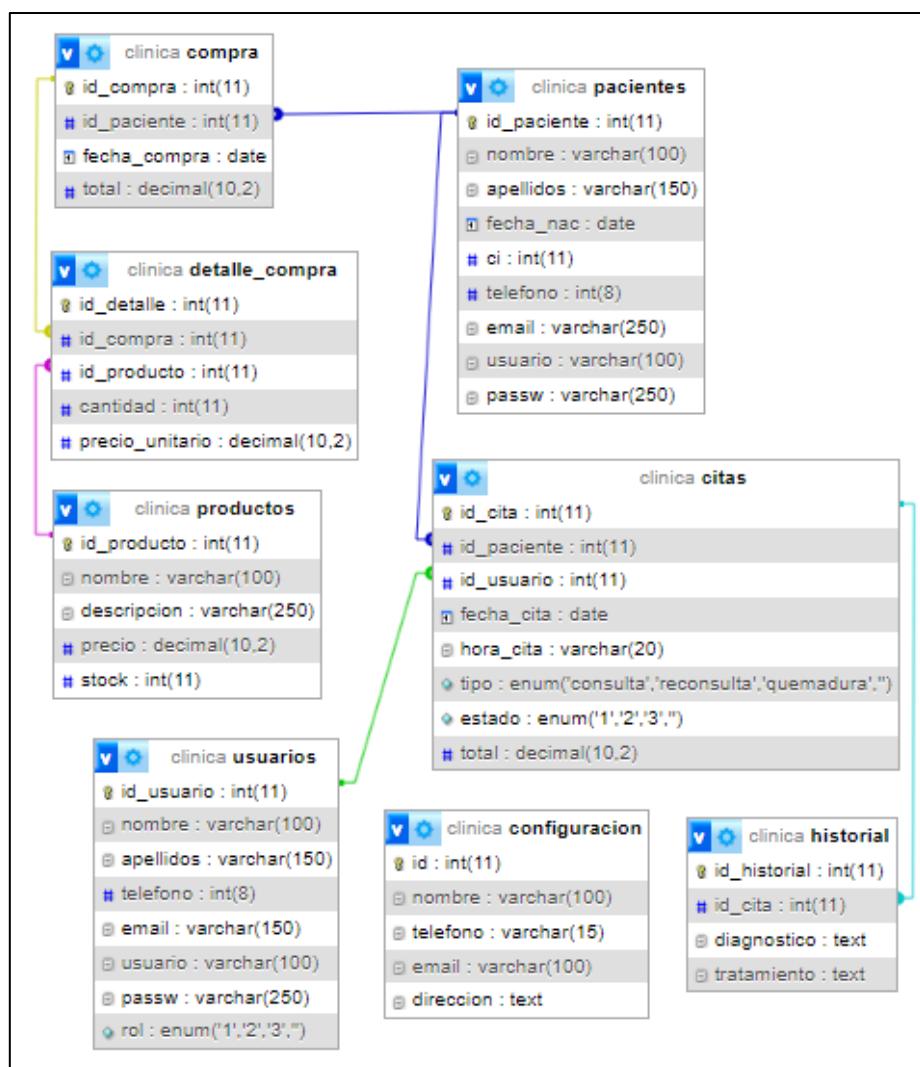


FUENTE: Elaboración propia, en base a Requerimientos Funcionales.

3.2.5. Diagrama de Entidad - Relación

El diagrama de entidad-relación será fundamental en el proyecto de la clínica dermatológica para modelar y comprender la estructura de la base de datos y las relaciones entre las diferentes entidades. Permitió visualizar cómo se relacionan las entidades principales del sistema y cómo fluye la información entre ellas. Esto proporcionó una base sólida para el diseño y la implementación de la base de datos, asegurando que esté bien estructurada y optimizada para las necesidades del sistema.

ILUSTRACIÓN N° 14: DIAGRAMA ENTIDAD-RELACIÓN



FUENTE: Elaboración propia, en base a Requerimientos Funcionales.

3.2.6. Interfaces de Usuario

El diseño de interfaces de usuario ayuda a mejorar la experiencia del usuario al proporcionar una interacción intuitiva y eficiente con el sistema. Facilita la entrada y visualización de datos. Para realizar las siguientes interfaces se usó la herramienta de diseño “Figma”, en base a los colores institucionales y el logo como se pueden observar en las siguientes ilustraciones.

ILUSTRACIÓN N° 15: PALETA DE COLORES



FUENTE: Elaboración propia, en base a los Colores institucionales.

ILUSTRACIÓN N° 16: LOGO DE LA CLÍNICA



FUENTE: Elaboración propia, en base a logo de la clínica.

3.2.6.1. Landing Page

Como se puede observar en la ilustración 15, el landing page del sistema web de la Clínica Dermatológica “DERMA - CAM” presenta de manera clara y atractiva toda la información relevante para los usuarios. La sección "Acerca de Nosotros" ofrece una breve historia de la clínica, mientras que "Tratamientos" muestra los tratamientos disponibles, y "Contacto" proporciona la información necesaria para comunicarse con la clínica. Finalmente, los botones de "Inicio de Sesión" y "Registrarse" redirigen a los usuarios a estas funcionalidades, facilitando el acceso a servicios personalizados.

ILUSTRACIÓN N° 17: LANDING PAGE



ACERCA DE NOSOTROS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.



TRATAMIENTOS

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.



Light, Fast & Powerful

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.



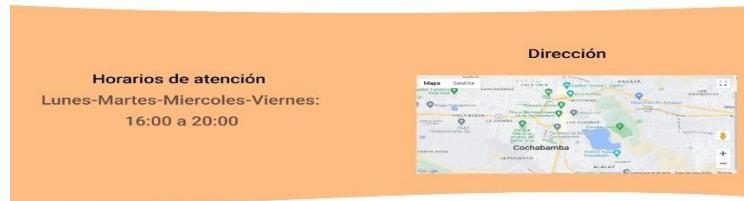
Light, Fast & Powerful

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.



Light, Fast & Powerful

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Donec quam felis, ultricies nec, pellentesque eu, pretium quis, sem. Nulla consequat massa quis enim.

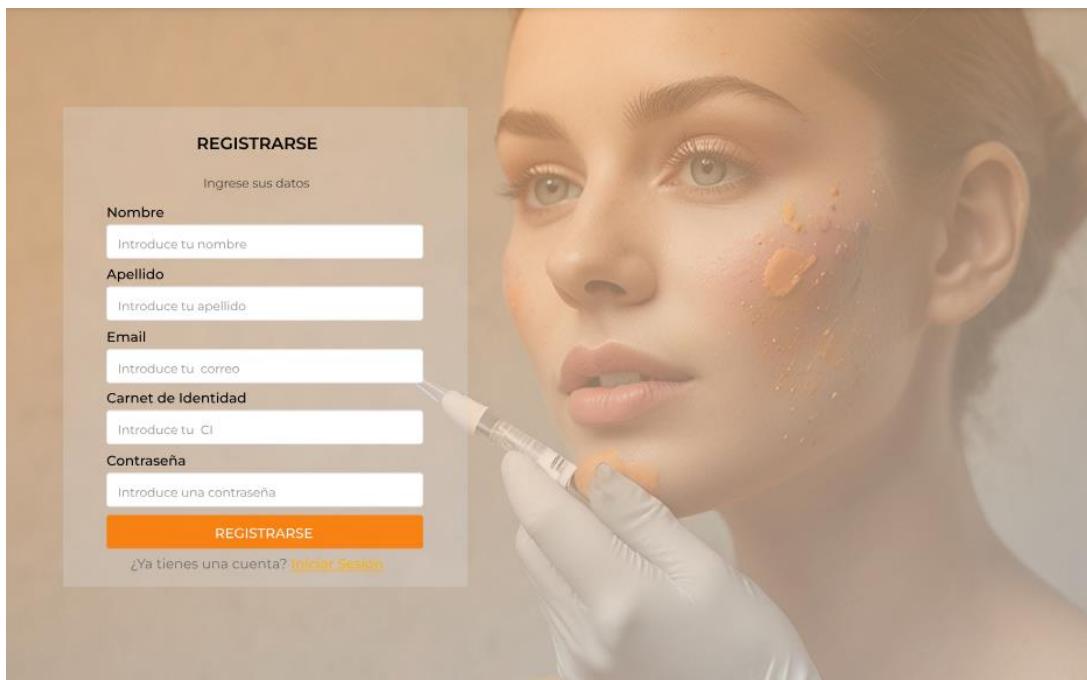


FUENTE: Elaboración propia, en base al diseño de interfaces de usuario.

3.2.6.2. Registro

Considerando los requerimientos de usuario para los datos del cliente, se diseñó un formulario con los datos indispensables del paciente. Este formulario asegura que se recopile toda la información necesaria para brindar una atención adecuada y personalizada a cada paciente, facilitando así una gestión eficiente de sus datos dentro del sistema.

ILUSTRACIÓN N° 18: DISEÑO UI DE REGISTRO

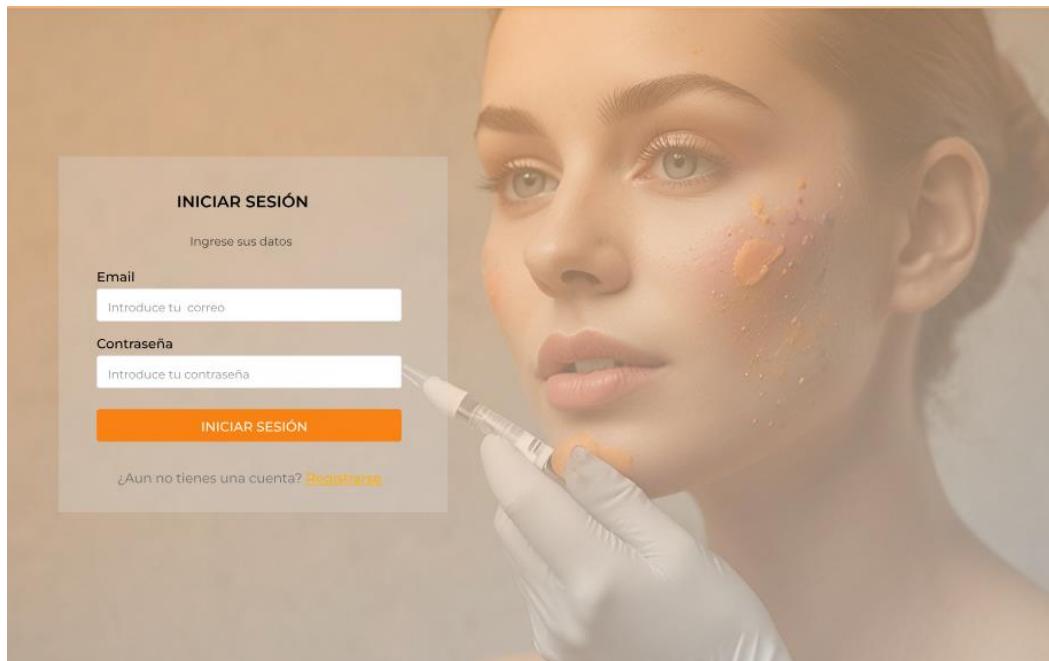


FUENTE: Elaboración propia, en base al diseño de interfaces de usuario.

3.2.6.3. Inicio de Sesión

En el diseño de interfaz de usuario del inicio de sesión podremos ver el formulario para ingresar al sistema una vez se creó el usuario y contraseña.

ILUSTRACIÓN N° 19: DISEÑO UI INICIO DE SESIÓN

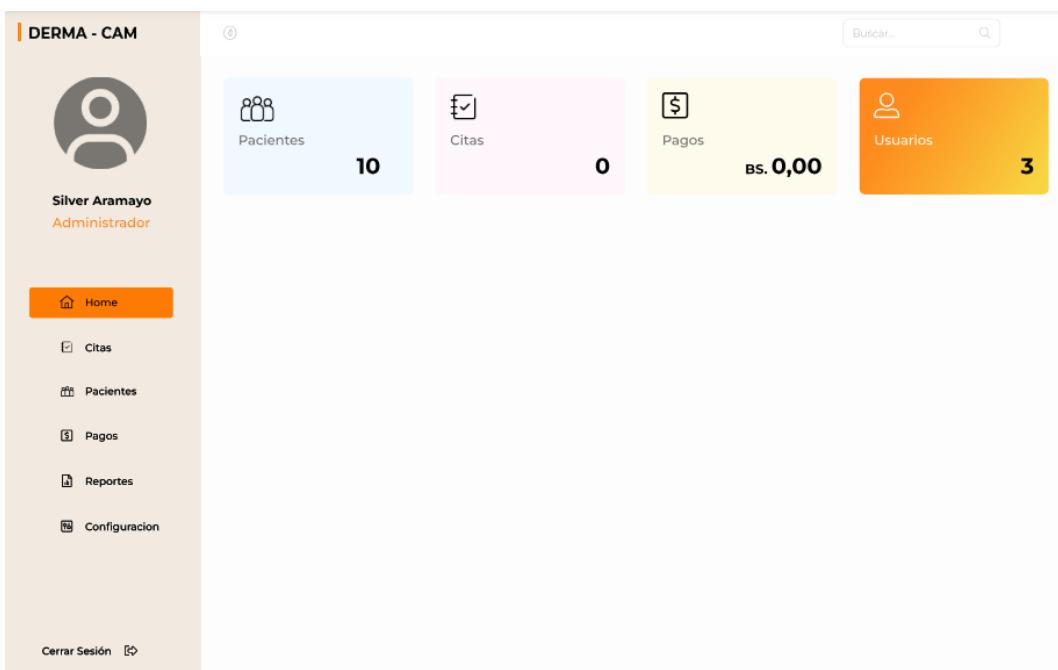


FUENTE: Elaboración propia, en base al diseño de interfaces de usuario.

3.2.6.4. Home Administrador

Al iniciar sesión en el sistema, tanto el paciente como el usuario serán redirigidos a su respectivo home como se puede, como se puede observar en la siguiente ilustración en el diseño de la interfaz de usuario. Este home actuará como un panel de administración personalizado, donde cada tipo de usuario podrá gestionar la información a la que tiene acceso según su rol, asegurando una experiencia adaptada y eficiente para cada uno.

ILUSTRACIÓN N° 20: DISEÑO PANEL DE ADMINISTRACIÓN



FUENTE: Elaboración propia, en base al diseño de interfaces de usuario.

3.2.6.5. Panel Pacientes

En el diseño de interfaz de pacientes, como se puede observar en la ilustración 19, se podrá gestionar los pacientes realizando acciones como crear, editar y eliminar.

ILUSTRACIÓN N° 21: DISEÑO UI PANEL DE PACIENTES

The screenshot shows a user interface for a medical application named 'DERMA - CAM'. On the left, there's a sidebar with a profile picture of a user named 'Silver Aramayo' (Administrator). Below the profile are links for 'Home', 'Citas', 'Pacientes' (which is highlighted with an orange background), 'Pagos', 'Reportes', and 'Configuración'. At the bottom of the sidebar is a 'Cerrar Sesión' button. The main content area is titled 'Lista de Pacientes' and contains a table with two rows of patient data:

	Nombre	Apellido	Email	Carnet de Identidad	Fecha de Nacimiento	Acciones
	Juan	Perez	JuanPerez@gmail.com	12345678	31/11/1990	Historia Tratamiento
	Pedro	Gonzales	PedroGonzales@gmail.com	12345678	31/11/1990	Historia Tratamiento

FUENTE: Elaboración propia, en base al diseño de interfaces de usuario.

3.3. DESARROLLO DEL SISTEMA

En esta parte, se explica el desarrollo del sistema desde las interfaces, las funcionalidades y la lógica necesaria para el cumplimiento de los requerimientos solicitados y así cumplir con las expectativas del proyecto.

El lenguaje principal que se utilizó para la codificación del sistema web es PHP en conjunto a la biblioteca Bootstrap, la cual fue utilizada para crear interfaces de usuario responsivas, también fue útil para estilizar los formularios y los elementos de la interfaz.

ILUSTRACIÓN N° 22: LOGOS BOOTSTRAP Y PHP

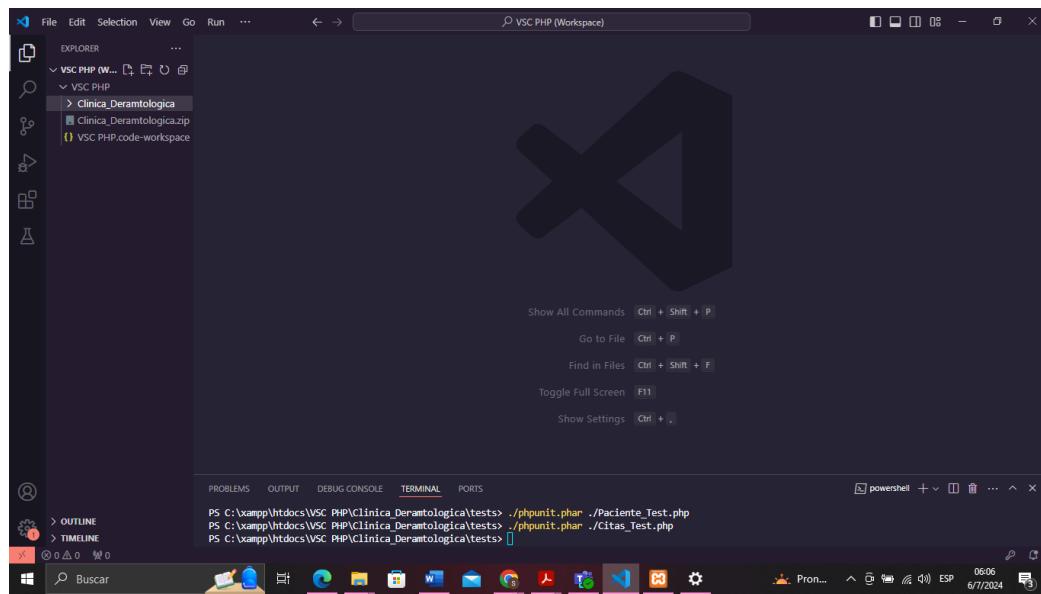


FUENTE: <https://www.bootstrapdash.com/blog/use-bootstrap-with-php,2024>.

3.3.1. Entorno de Desarrollo Integrado (IDE)

Visual Estudio Code es el entorno de desarrollo que se utilizó para el desarrollo del sistema web, con su interfaz amigable, multiplataforma, multilenguaje, extensa biblioteca de extensiones y plugins, ligero y rápido, es la mejor opción para el desarrollo de este proyecto por todas las ventajas que ofrece este entorno de desarrollo.

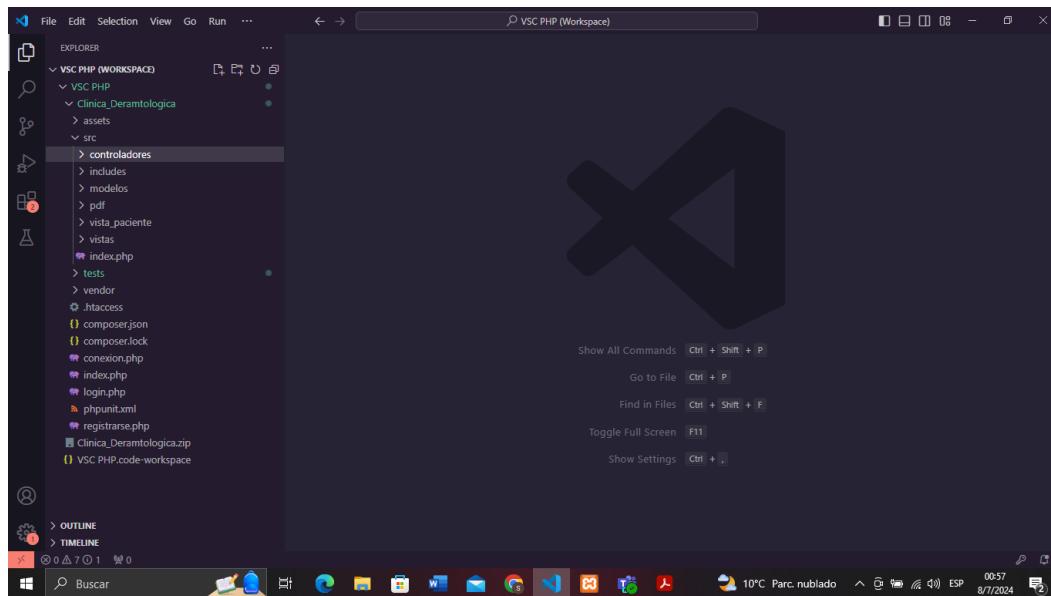
ILUSTRACIÓN N° 23: INTERFAZ DE VISUAL STUDIO CODE



FUENTE: Elaboración propia, en base a Visual Estudio Code.

Se realizo la organización de las carpetas según la arquitectura de modelo, vista, controlador, lo cual ayuda a tener mejor organización de los archivos.

ILUSTRACIÓN N° 24: CARPETAS ORDENADAS SEGÚN ARQUITECTURA MVC.



FUENTE: Elaboración propia, en base a organización carpetas según arquitectura MVC.

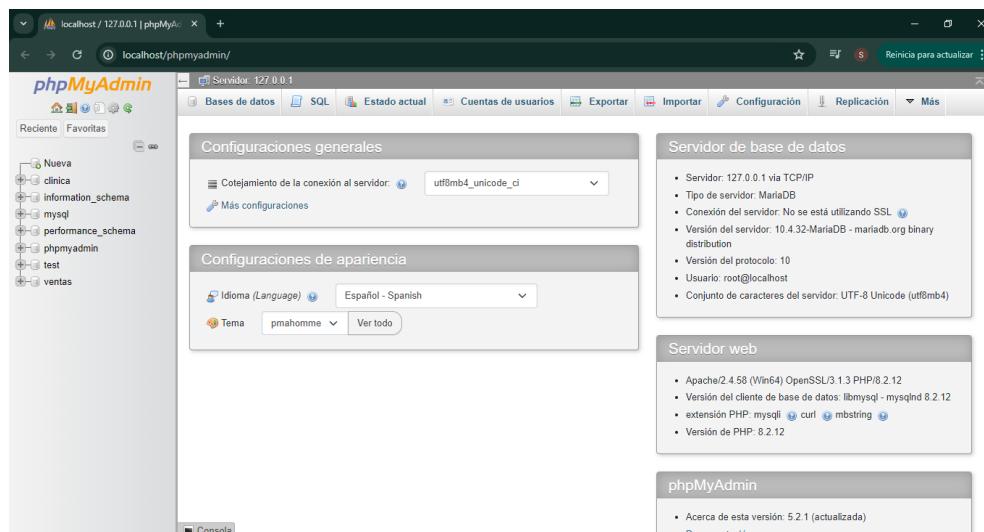
3.3.2. Desarrollo de la Base de Datos

Para el desarrollo de la base de datos, se utilizó MySQL como sistema gestor de bases de datos y phpMyAdmin como su interfaz gráfica de administración. Este entorno proporcionó una interfaz gráfica intuitiva y fácil de usar para gestionar y administrar las bases de datos de manera eficiente.

3.3.2.1. phpMyAdmin

Para el desarrollo de la base de datos del sistema web se hará uso de phpMyAdmin, esta herramienta de administración web para bases de datos MySQL que facilita la gestión mediante una interfaz gráfica amigable. Permite crear, modificar y eliminar bases de datos y tablas, ejecutar consultas SQL, y gestionar usuarios y permisos. Es útil tanto para usuarios novatos como para administradores avanzados, simplificando tareas complejas y mejorando la eficiencia en la administración de bases de datos.

ILUSTRACIÓN N° 25: INTERFAZ DE PHPMYADMIN

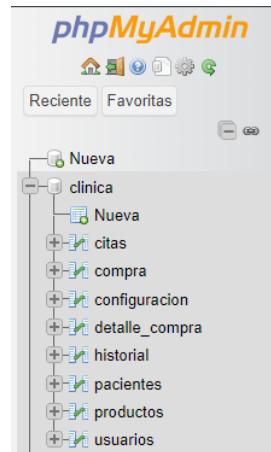


FUENTE: Elaboración propia en base a phpMyAdmin .

3.3.2.2. Desarrollo de la base de datos

Con la ayuda de la herramienta phpMyAdmin se desarrolló la base de datos según al diagrama entidad relación, creando las tablas y las relaciones entre ellas y así poder iniciar el desarrollo del sistema en relación con la base de datos.

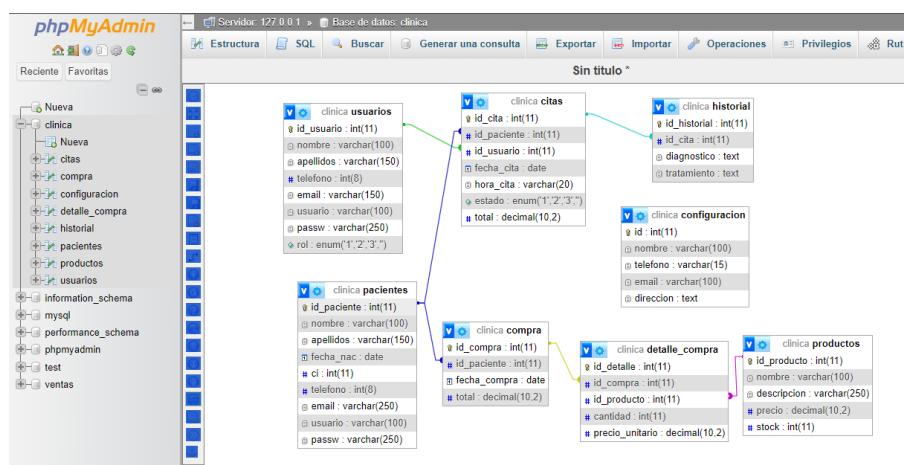
ILUSTRACIÓN N° 26: TABLAS DE LA BASE DE DATOS



FUENTE: Elaboración propia, en base a phpMyAdmin.

También con la ayuda de la herramienta phpMyAdmin se logró generar la estructura de la base de datos.

ILUSTRACIÓN N° 27: ESTRUCTURA DE LA BASE DE DATOS



FUENTE: Elaboración propia, en base a phpMyAdmin.

3.3.3. Codificación del Sistema Web

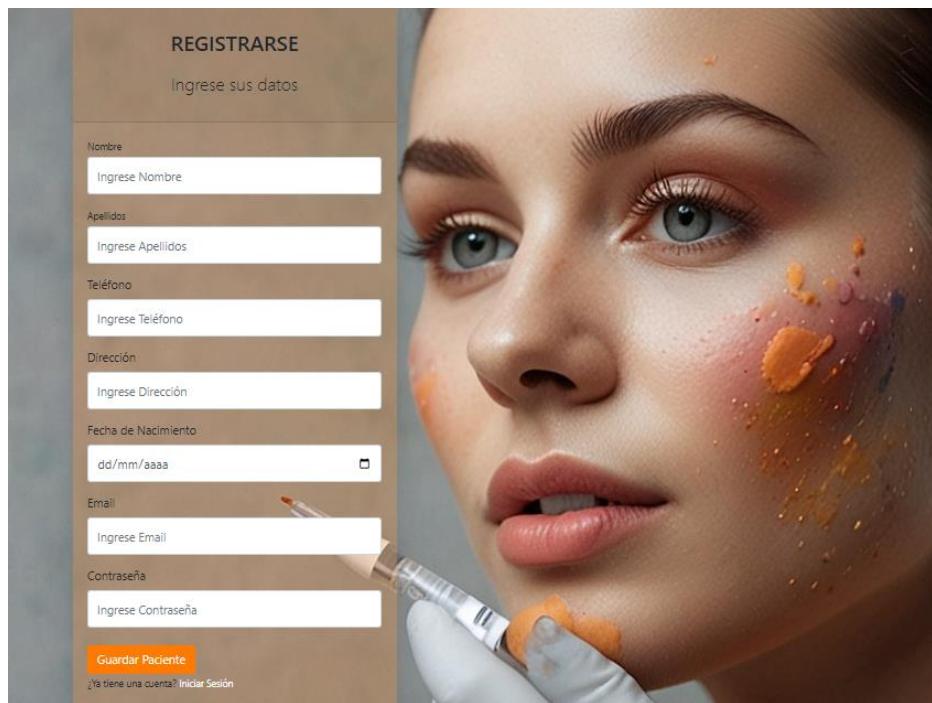
En este punto del desarrollo empieza el proceso de codificación del sistema, creando capas según la arquitectura que se pudo mostrar anteriormente la cual es el modelo, vista, controlador en las cuales se representan con las siguientes capas:

- Capa de presentación que son las interfaces de usuario, formularios y páginas los cuales estarán ubicados en la carpeta de vistas.
- Capa de lógica que maneja las solicitudes del usuario, procesos de negocio, control de flujo, controladores y scripts que procesan formularios, realizan validaciones y ejecutan la lógica de la aplicación , los cuales serán almacenadas en la carpeta de controladores.
- Capa de datos que realiza consultas SQL para interactuar con la base de datos, los cuales se entienden como modelos y estarán almacenadas en la carpeta del mismo nombre.

3.3.3.1. Registrarse e Iniciar Sesión

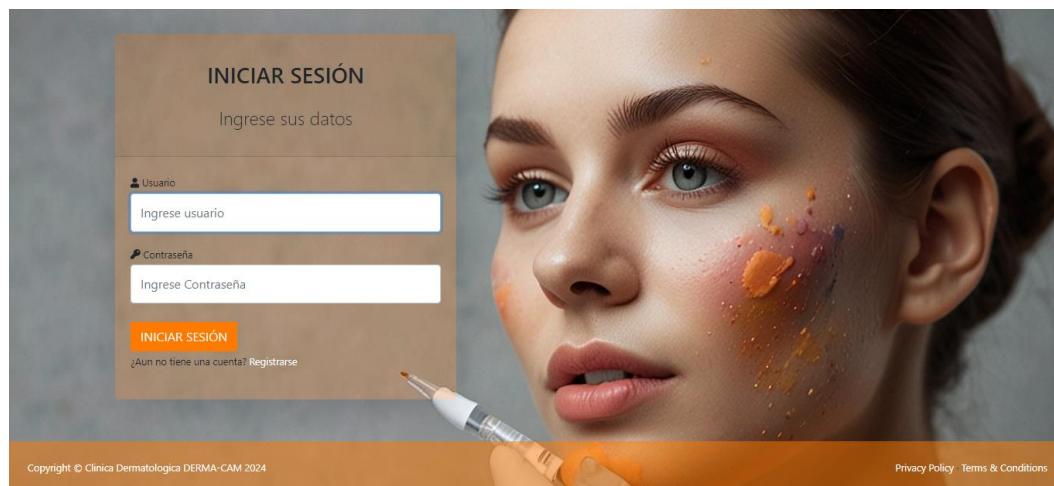
La función iniciar sesión es compartida por los usuarios y pacientes, esto para simplificar el desarrollo y mantenimiento del sistema, para lo cual se implementó un control de sesiones para asegurar el acceso. En el caso de registrarse podrán realizar su registro los pacientes que no tengan una cuenta, mediante un formulario en el cual ingresarán su información, crearán su usuario y contraseña. En el caso del usuario su cuenta tiene que ser creada necesariamente por el administrador del sistema.

En la siguiente ilustración se observa el formulario para el registro de los pacientes, si se desea revisar parte del código de esta funcionalidad se encuentra en los anexos (ver Anexo III).

ILUSTRACIÓN N° 28: FORMULARIO PARA REGISTRARSE

FUENTE: Elaboración propia, en base a Registrarse del sistema,2024

Como se puede observar sigue el diseño previamente realizado, ver ilustración 16, con la diferencia que se agregaron todos los campos necesarios según la base de datos. Parte del código realizado puede ser analizado en anexos (ver Anexo III).

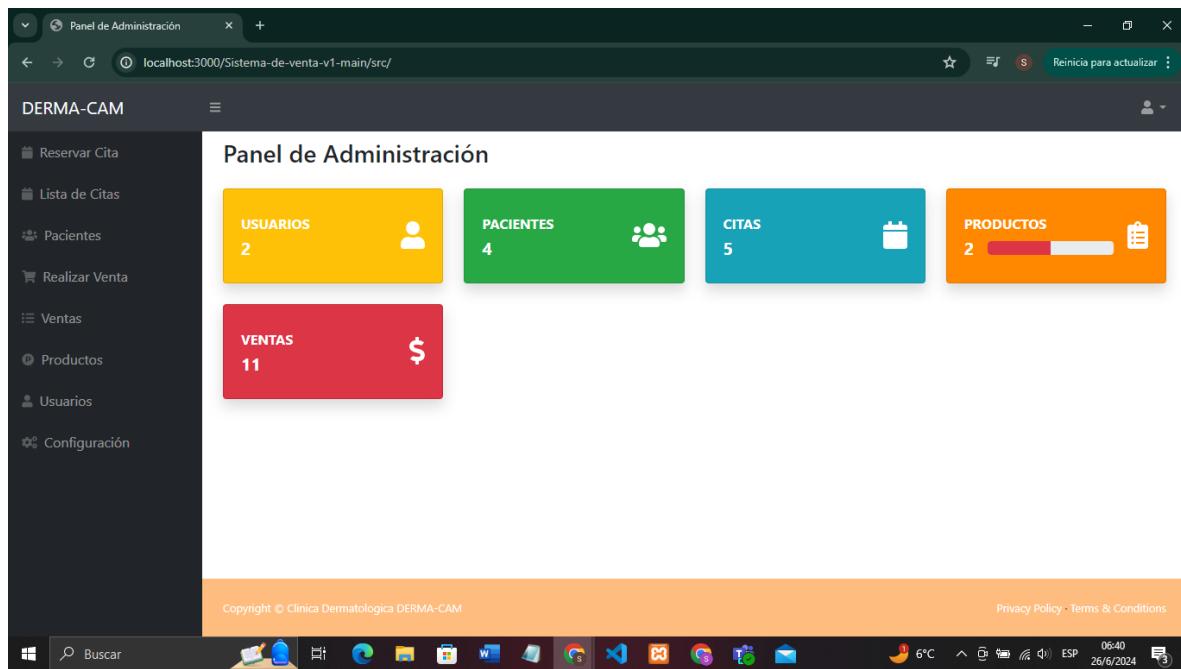
ILUSTRACIÓN N° 29: INICIAR SESIÓN

FUENTE: Elaboración propia, en base a Iniciar Sesión del sistema,2024

3.3.3.2. Panel de Administración

En el panel de administración se podrá encontrar unas tarjetas que darán una información rápida de algunos apartados como el número de pacientes, usuarios, citas, productos y ventas. Lo cual ayudara también a ingresar de manera rápida a la lista de dichos apartados (ver Anexo III).

ILUSTRACIÓN N° 30: PANEL DE ADMINISTRACIÓN



FUENTE: Elaboración propia, en base a Panel de Administracion,2024

3.3.3.3. Gestión de Usuarios

En este punto se podrá realizar la gestión de los Usuarios los cuales son los médicos, asistentes y personal de la clínica. Aquí se podrá gestionar desde la creación de usuarios, buscarlos, también permitirá editar los datos del usuario y también borrarlos (ver Anexo III).

ILUSTRACIÓN N° 31: GESTOR DE USUARIOS

The screenshot shows a web-based administration panel titled 'Panel de Administración'. The main title is 'DERMA-CAM'. On the left sidebar, there are links for 'Reservar Cita', 'Lista de Citas', 'Pacientes', 'Realizar Venta', 'Ventas', 'Productos', 'Usuarios' (which is selected), and 'Configuración'. The central area has a heading 'Crear Usuario +' and a table showing three user entries:

ID	Nombre	Apellidos	Teléfono	Email
1	Administrador	Adm	69494776	Adm@gmail.com
3	Nataly Belen	Aramayo Borges	76469566	natalyAramayo@gmail.com
4	Carolina	Achá Morato	69452541	carolinaacham@gmail.com

Below the table, it says 'Showing 1 to 3 of 3 entries'. At the bottom right, there are 'Previous' and 'Next' buttons. The status bar at the bottom shows system information like '6°C', '06:47', 'ESP', and the date '26/6/2024'.

FUENTE: Elaboración propia, en base a Gestor de Usuarios,2024

Al hacer un clic en Crear Usuario se desplegará un formulario para la creación de un nuevo usuario. Parte del código en los anexos (ver Anexo III).

ILUSTRACIÓN N° 32: FORMULARIO CREAR USUARIO

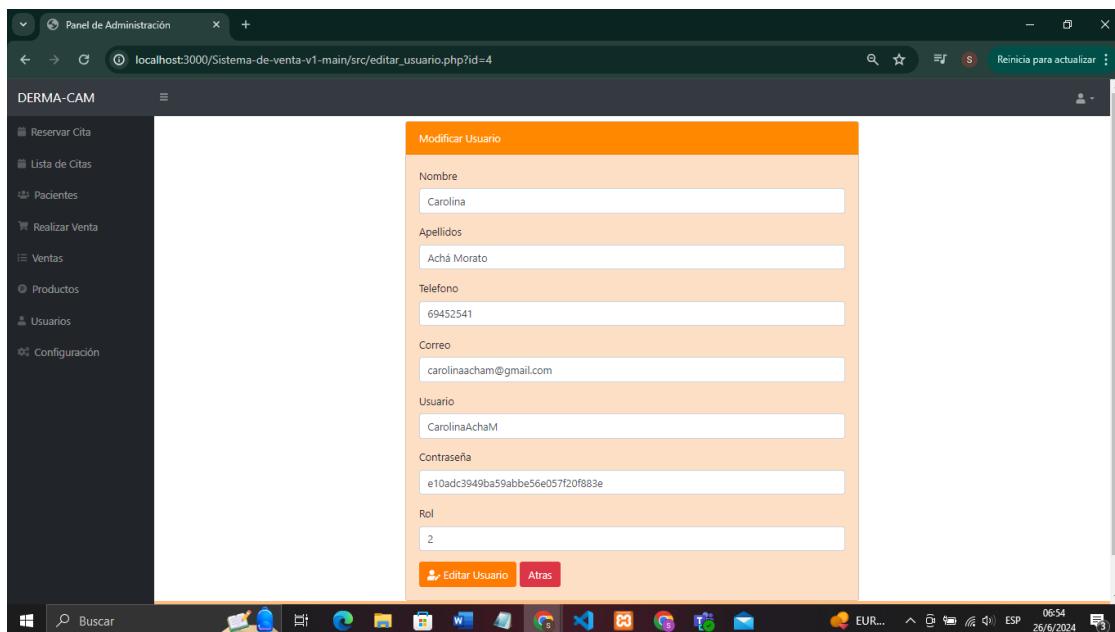
The screenshot shows the same administration panel as in Illustration 31. A modal window titled 'Nuevo Usuario' is open in the center. It contains fields for 'Nombre' (with placeholder 'Ingrese Nombre'), 'Apellidos' (placeholder 'Ingrese Apellidos'), 'Teléfono' (placeholder 'Ingrese Teléfono'), 'Correo Electrónico' (placeholder 'Ingrese Correo Electronico'), 'Usuario' (placeholder 'Ingrese Usuario'), 'Contraseña' (placeholder 'Ingrese Contraseña'), and 'Rol' (placeholder 'Ingrese Rol'). At the bottom of the modal is a 'Registrar' button. The background shows the list of users from Illustration 31.

FUENTE: Elaboración propia, en base a Gestor de Usuarios,2024

En la lista de usuarios también se podrá encontrar dos botones de acción los cuales ayudaran a editar y eliminar al usuario.

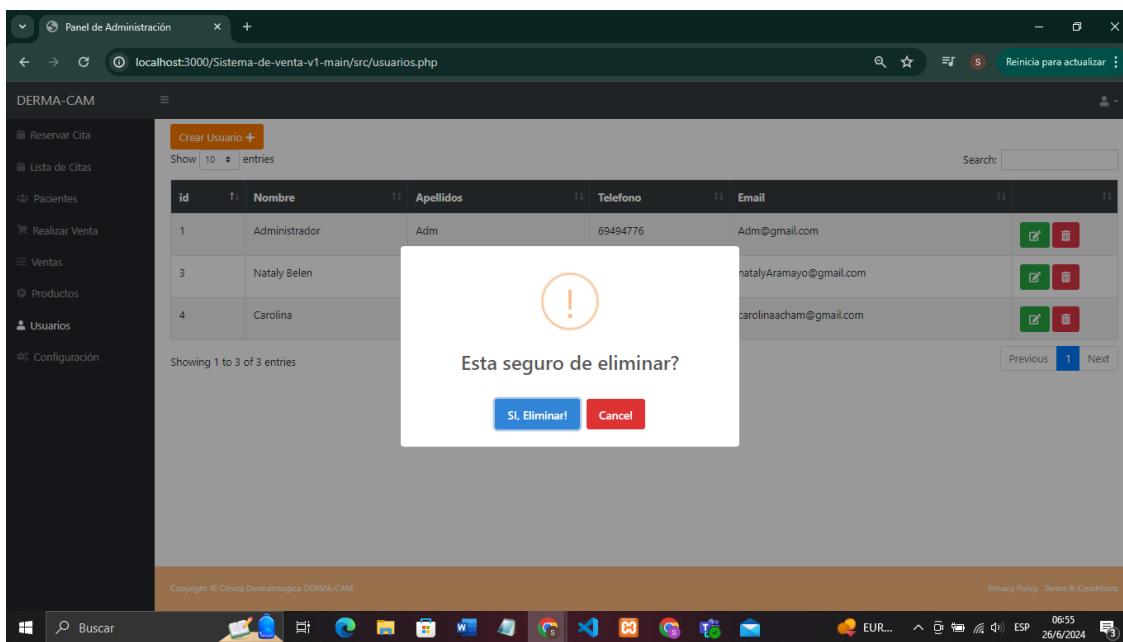
En la siguiente imagen se ilustra la funcionalidad de editar usuario. Parte del código en los anexos (ver Anexo III).

ILUSTRACIÓN N° 33: EDITAR USUARIO



FUENTE: Elaboración propia, en base a Gestor de Usuarios,2024

En la siguiente ilustración se puede observar la funcionalidad de eliminar usuario el cual, al oprimir el botón de acción, esta muestra una alerta para confirmar si en realidad desea eliminar el registro, Parte del código en los anexos (ver Anexo III).

ILUSTRACIÓN N° 34: ELIMINAR USUARIO

FUENTE: Elaboración propia, en base a Gestor de Usuarios,2024

3.3.3.4. Gestión de Pacientes

El desarrollo de esta función permite realizar la gestión de los pacientes los cuales, si bien pueden realizar su registro y gestionar su propia información, hay pacientes los cuales por diferentes razones se les dificulta el manejo del sistema pueden ser asistidos por un asistente de la clínica. En esta sección se podrá realizar la creación del paciente, la actualización de su información y la eliminación del paciente.

En la siguiente imagen se puede observar el gestor de pacientes desde la vista del usuario, el cual muestra la lista de pacientes y los diferentes botones de acción.

ILUSTRACIÓN N° 35: GESTOR DE PACIENTES

ID	Nombre	Apellidos	Fecha Nac.	CI	Teléfono	Email
1	Silver Luis	Aramayo Borges	1997-10-31	10540691	69494776	slab3110@gmail.com
2	Juan Carlos	Carpio Soto	1990-02-22	23654789	65478124	JuanCarpio@gmail.com
4	Jose	Martinez	1992-02-12	56987456	65478234	JoseM@gmail.com
5	Karen Joselin	Mendoza Terceros	1998-11-13	13289424	76915865	KarenMJ@gmail.com

FUENTE: Elaboración propia, en base a Gestor de Pacientes,2024

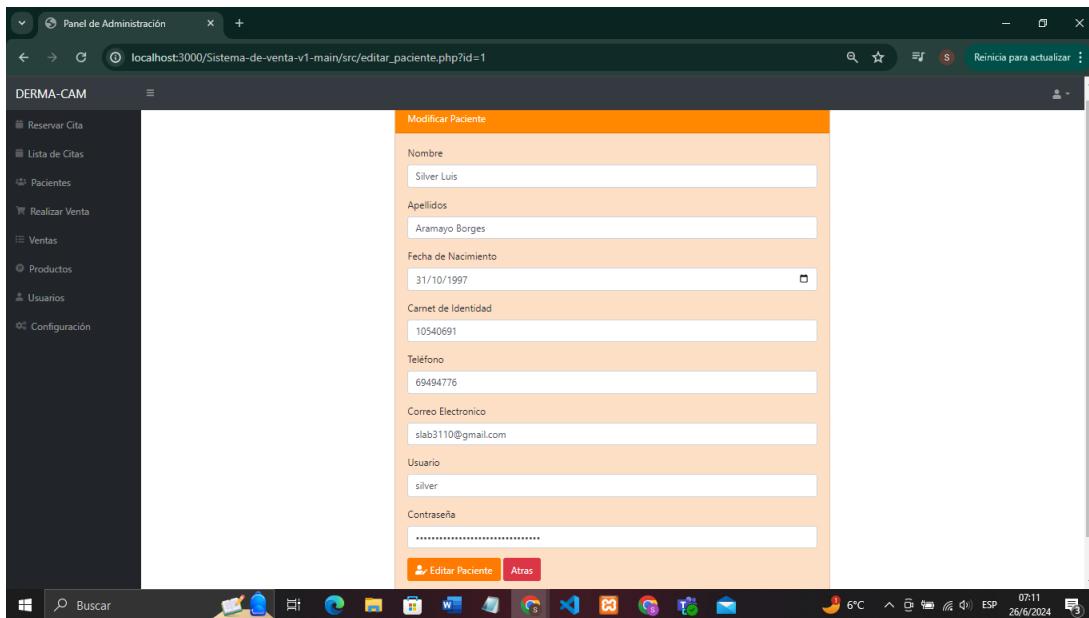
A continuación, se pude observar el formulario para crear un paciente desde la vista del usuario.

ILUSTRACIÓN N° 36: FORMULARIO CREACIÓN DE PACIENTE

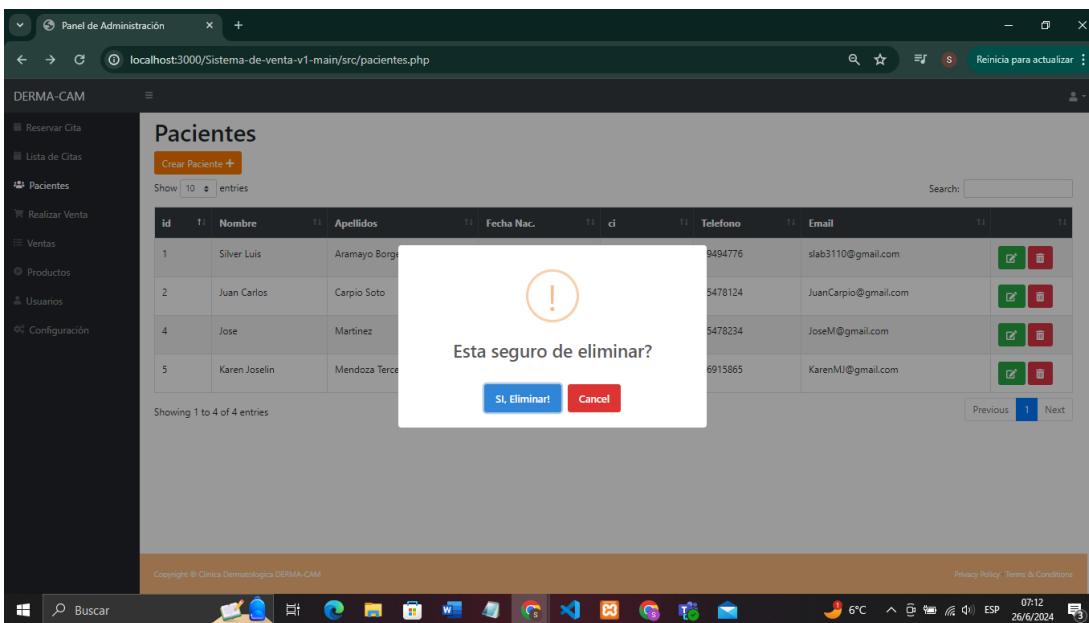
id	Nombre	Apellido
1	Silver Luis	Aramayo
2	Juan Carlos	Carpio
4	Jose	Martinez
5	Karen Joselin	Mendoza

FUENTE: Elaboración propia, en base a Gestor de Pacientes,2024

En las siguientes ilustraciones se pueden observar las funcionalidades de editar y eliminar pacientes.

ILUSTRACIÓN N° 37: EDITAR PACIENTE

FUENTE: Elaboración propia, en base a Gestor de Pacientes,2024

ILUSTRACIÓN N° 38: ELIMINAR PACIENTE

FUENTE: Elaboración propia, en base a Gestor de Pacientes,2024

3.3.3.5. Gestión de Citas

En este punto se puede ver la implementación del apartado de citas el cual se puede reservar citas tanto por los usuarios como los pacientes, llenando un formulario con la información necesaria como el nombre del paciente en el caso del usuario, en el caso del paciente se auto rellena con el inicio de sesión, también el nombre del médico, la fecha de la cita se podrá seleccionar en el calendario desplegable y por último según el día seleccionado se podrá escoger entre los horarios disponibles.

Para facilitar el ingreso de los datos del paciente y el medico se realiza una búsqueda de manera asincrónica al ingresar las 3 primeras letras del nombre. Parte del código en anexos (ver Anexo III).

ILUSTRACIÓN N° 39: AGENDAR CITA

The screenshot displays the 'Agendar Cita' (Schedule Appointment) form within a web browser window titled 'Panel de Administración'. The URL is 'localhost:3000/Clinica_Deramtológica/src/citas.php'. The main content area is titled 'Agendar Cita' and contains several input fields:

- Search fields for patients: 'Buscar Paciente:' (Nombre del paciente), 'Apellidos del Paciente', 'CI del Paciente'.
- Search fields for professionals: 'Buscar Profesional:' (Nombre del médico), 'Apellidos del Profesional', 'Cargo del Profesional'.
- Appointment details: 'Tipo de Cita:' dropdown set to 'Consulta', 'Fecha:' date input, 'Hora:' time input.
- Action button: 'Reservar' (Reserve).

The sidebar on the left lists various administrative functions: Reservar Cita, Lista de Citas, Pacientes, Realizar Venta, Ventas, Productos, Usuarios, and Configuración. The bottom of the screen shows a Windows taskbar with icons for search, file explorer, and other applications.

FUENTE: Elaboración propia, en base a Agendar Cita,2024.

Al realizar la cita se envía un mensaje WhatsApp al número del cliente confirmando la cita, esto mediante el api de meta que permite enviar mensajes desde el sistema con una plantilla creada en Meta for Developers.

En la siguiente imagen, se ilustra la plantilla usada para el mensaje de confirmación.

ILUSTRACIÓN N° 40: PLANTILLAS CREADA EN META

The screenshot shows the 'Message Templates' section in the Facebook Business Manager. The left sidebar has icons for messages, calls, files, and more. The main area is titled 'Plantillas de mensajes' (Message Templates) and shows a list of four active templates:

Nombre de la plantilla	Categoría	Idioma	Estado	Mensajes enviados	Mensajes abiertos	Motivo principal...	Última actualizaci...
citaenviada	Utilidad	English Cita Registrada Correctamen...	Activa: calidad per	0	0	—	29 jun 2024
citaregistrada	Utilidad	Spanish Cita registrada correctament...	Activa: calidad per	—	Menos de ...	0	—
registrado	Utilidad	Spanish Hola, tu cita ha sido reservad...	Activa: calidad per	0	0	—	29 jun 2024
hello_world	Utilidad	English (US) Welcome and congratulation...	Activa: calidad per	—	Menos de ...	0	—

At the bottom, it says 'Se muestran 4 plantillas de mensajes (Total de plantillas activas: 4 de 250)'.

FUENTE: Elaboración propia, en base a Meta for Developers,2024.

A continuación, se puede observar el mensaje recibido a través de WhatsApp, con el formato de la plantilla.

ILUSTRACIÓN N° 41: MENSAJE DE CONFIRMACIÓN DE CITA.

The screenshot shows a WhatsApp conversation with a contact named '+591 77485546'. The message received is a template from 'Clinica Dermatologica "Derma-Cam"'. It contains the following information:

Ubicación: <https://maps.app.goo.gl/DDdgqRe95XTqMeam> AYER

Clinica Dermatologica "Derma-Cam".
Cita Registrada Correctamente
Fecha: 01-07-2024.
Hora: 16:00.
Ubicación: <https://maps.app.goo.gl/DDdgqRe95XTqMeam> 4:19

HOY

Clinica Dermatologica "Derma-Cam".
Cita Registrada Correctamente
Fecha: 01-07-2024.
Hora: 16:00.
Ubicación: <https://maps.app.goo.gl/DDdgqRe95XTqMeam> 1:27

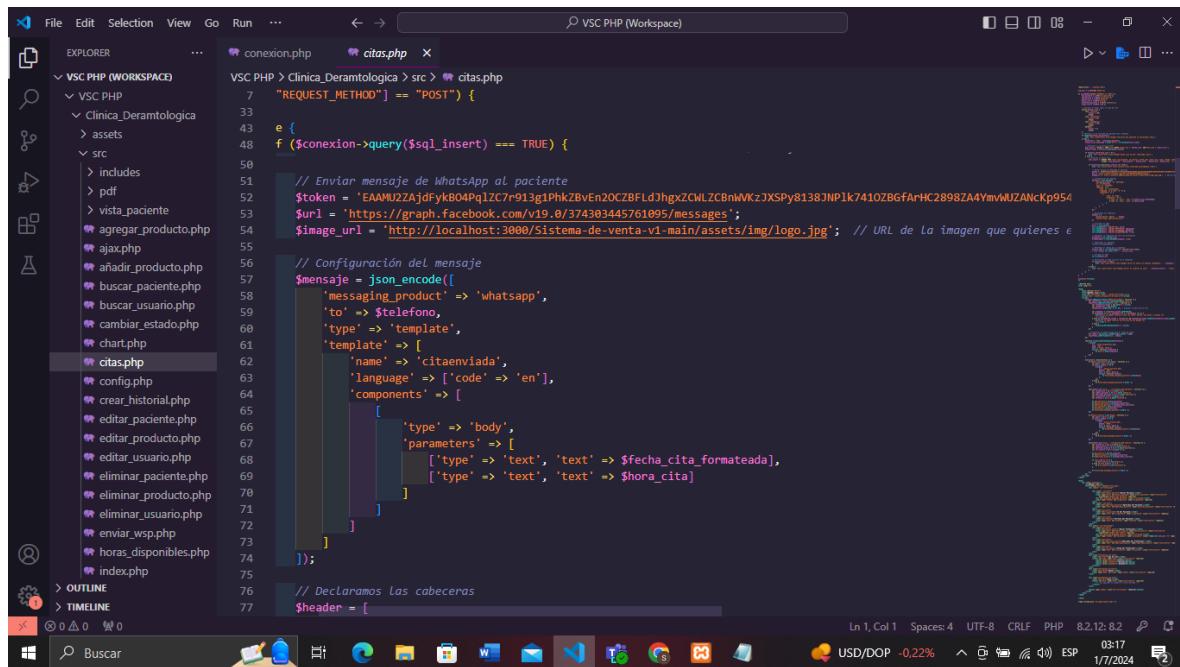
Clinica Dermatologica "Derma-Cam".
Cita Registrada Correctamente
Fecha: 01-07-2024.
Hora: 16:15.
Ubicación: <https://maps.app.goo.gl/DDdgqRe95XTqMeam> 2:54

Actualmente, esta empresa está usando un servicio seguro de Meta para administrar este chat. Haz clic para obtener más información.

FUENTE: Elaboración propia, en base WhatsApp,2024.

En la siguiente ilustración, se puede observar el código para el envío de la confirmación de la cita usando la API de meta, con el mensaje en Json que contiene los datos del paciente en este caso el número de teléfono y también el nombre de la plantilla usada con las variables que se agregarán a la plantilla para enviar el mensaje con la fecha y hora de la cita reservada.

ILUSTRACIÓN N° 42: CÓDIGO PARA ENVIÓ DEL MENSAJE



The screenshot shows a Microsoft Visual Studio Code (VS Code) interface with a dark theme. The left sidebar displays a file tree for a 'VSC PHP (WORKSPACE)' project, including files like 'conexion.php', 'citas.php', 'agregar_producto.php', 'ajax.php', 'anadir_producto.php', 'buscar_paciente.php', 'buscar_usuario.php', 'cambiar_estado.php', 'chart.php', 'citas.php', 'config.php', 'crear_historial.php', 'editar_paciente.php', 'editar_producto.php', 'editar_usuario.php', 'eliminar_paciente.php', 'eliminar_producto.php', 'eliminar_usuario.php', 'enviar_vsp.php', 'horas_disponibles.php', and 'index.php'. The main editor area contains PHP code for sending a WhatsApp message. The code includes a JSON message structure with fields like 'messaging_product' (set to 'whatsapp'), 'to' (set to '\$telefono'), 'type' (set to 'template'), and a 'template' object containing a 'name' (set to 'citaenviada'), 'language' (set to 'code' => 'en'), and 'components' (containing a single component with 'type' => 'body' and 'parameters' containing two text parameters: 'text' => \$fecha_cita_formateada and 'text' => \$hora_cita). The code is annotated with comments explaining the configuration of the message. At the bottom of the editor, there are tabs for 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', and '...', along with a search bar and a status bar showing file paths, line numbers, and other development information.

```

File Edit Selection View Go Run ...
VSC PHP (Workspace)
explorer conexiones.php citas.php
VSC PHP > Clinica_Dermatologica > src > citas.php
7 "REQUEST_METHOD" == "POST" {
33
e {
43 f {
48 if ($conexión->query($sql_insert) === TRUE) {
50
51 // Enviar mensaje de WhatsApp al paciente
52 $token = 'EAAMj2ZAidykB04Pq1ZC7r91sgtPhkZBvEn20CZBFldJhgxZCMZCBnWVzJXSPy8138JNPlk7410ZBGfArHC2898ZA4YmvMuzAnKp954';
53 $url = 'https://graph.facebook.com/v10.0/374303445761095/messages';
54 $image_url = 'http://localhost:3000/Sistema-de-venta-v1-main/assets/img/logo.jpg'; // URL de la imagen que quieres enviar
55
56 // Configuración del mensaje
57 $mensaje = json_encode([
58     'messaging_product' => 'whatsapp',
59     'to' => $telefono,
60     'type' => 'template',
61     'template' => [
62         'name' => 'citaenviada',
63         'language' => ['code' => 'en'],
64         'components' => [
65             [
66                 'type' => 'body',
67                 'parameters' => [
68                     ['type' => 'text', 'text' => $fecha_cita_formateada],
69                     ['type' => 'text', 'text' => $hora_cita]
70                 ]
71             ]
72         ]
73     ]
74 );
75
76 // Declaramos Las cabeceras
77 $header = [

```

FUENTE: Elaboración propia, en base Proyecto Clínica Dermatológica,2024.

En la lista de citas se podrán ver las citas reservadas, las cuales se pueden cancelar haciendo clic en el botón de acción cancelar, así liberando la hora para que pueda ser tomada por otro cliente. Presionando el botón de iniciar cita el medico inicia la consulta ingresando a un formulario para poder ingresar el diagnóstico y el respectivo tratamiento pasando a ser una cita realizada y actualizando los botones de acción en el cual aparecerá un botón para ver el historial de dicha cita. Parte del código de esta funcionalidad en los anexos (ver Anexo III).

ILUSTRACIÓN N° 43: LISTA DE CITAS

The screenshot shows a web-based application titled 'DERMA-CAM'. On the left, a sidebar menu includes 'Reservar Cita', 'Lista de Citas' (selected), 'Pacientes', 'Realizar Venta', 'Ventas', 'Productos', 'Usuarios', and 'Configuración'. The main content area is titled 'Lista de Citas' and displays a table of appointments. The columns are: Cod. (Appointment ID), Paciente (Patient), Fecha (Date), Hora (Time), Total (Total), Tipo (Type), Estado (Status), and Acciones (Actions). Two entries are listed:

Cod.	Paciente	Fecha	Hora	Total	Tipo	Estado	Acciones
56	Silver Luis Aramayo Borges	2024-07-01	16:00	80.00	reconsulta	Realizado	<button>Ver Historial</button> <button>Cancelar Cita</button>
57	Silver Luis Aramayo Borges	2024-07-01	16:15	200.00		Reservado	<button>Iniciar Cita</button> <button>Cancelar Cita</button>

Below the table, it says 'Showing 1 to 2 of 2 entries'. At the bottom right, there are 'Previous' and 'Next' buttons. The status bar at the bottom shows '9°C Despejado' and the date '1/7/2024'.

FUENTE: Elaboración propia, en base a Lista de Citas,2024

Al hacer clic en el botón iniciar cita, manda al usuario a este apartado donde se ingresa el diagnóstico y el tratamiento para el paciente. Parte del código se puede observar en los anexos (ver Anexo III).

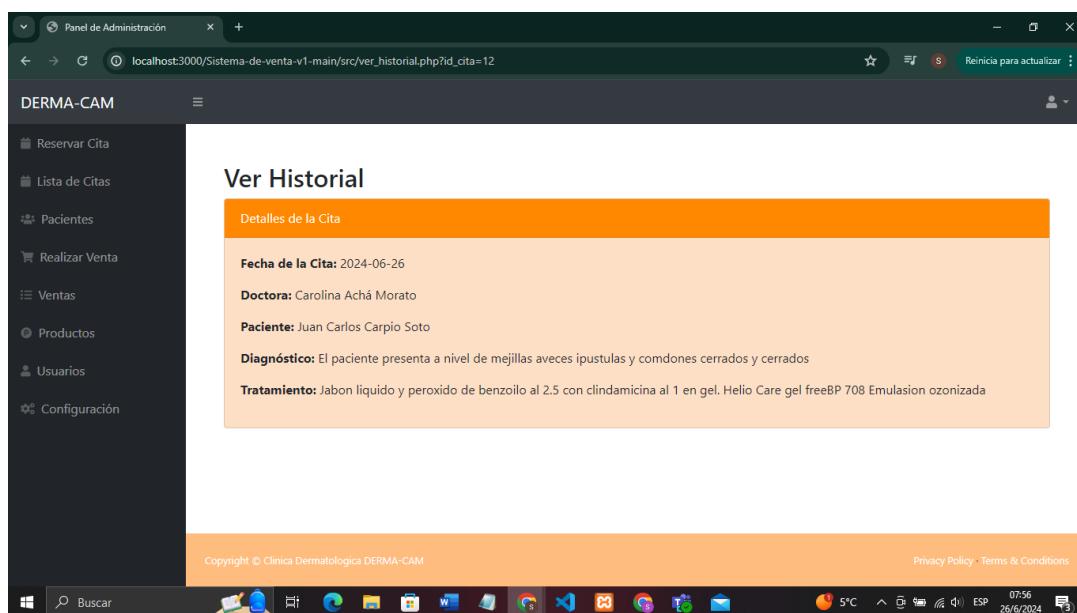
ILUSTRACIÓN N° 44: INGRESAR DIAGNÓSTICO Y TRATAMIENTO

The screenshot shows a web-based application titled 'Panel de Administración'. On the left, a sidebar menu includes 'Reservar Cita', 'Lista de Citas' (selected), 'Pacientes', 'Realizar Venta', 'Ventas', 'Productos', 'Usuarios', and 'Configuración'. The main content area is titled 'Crear Historial' and has two input fields: 'Diagnóstico:' and 'Tratamiento:', each with a text area. Below these fields is a large orange button labeled 'Guardar Historial'. The status bar at the bottom shows '5°C Despejado' and the date '26/6/2024'.

FUENTE: Elaboración propia, en base a Crear historial,2024

Al guardar el historial la cita pasa a realizarse y se puede ver el historial agregado a esa cita al pulsar el botón de ver historial que aparece tras guardar. Parte del código de esta funcionalidad en anexos (ver Anexo III).

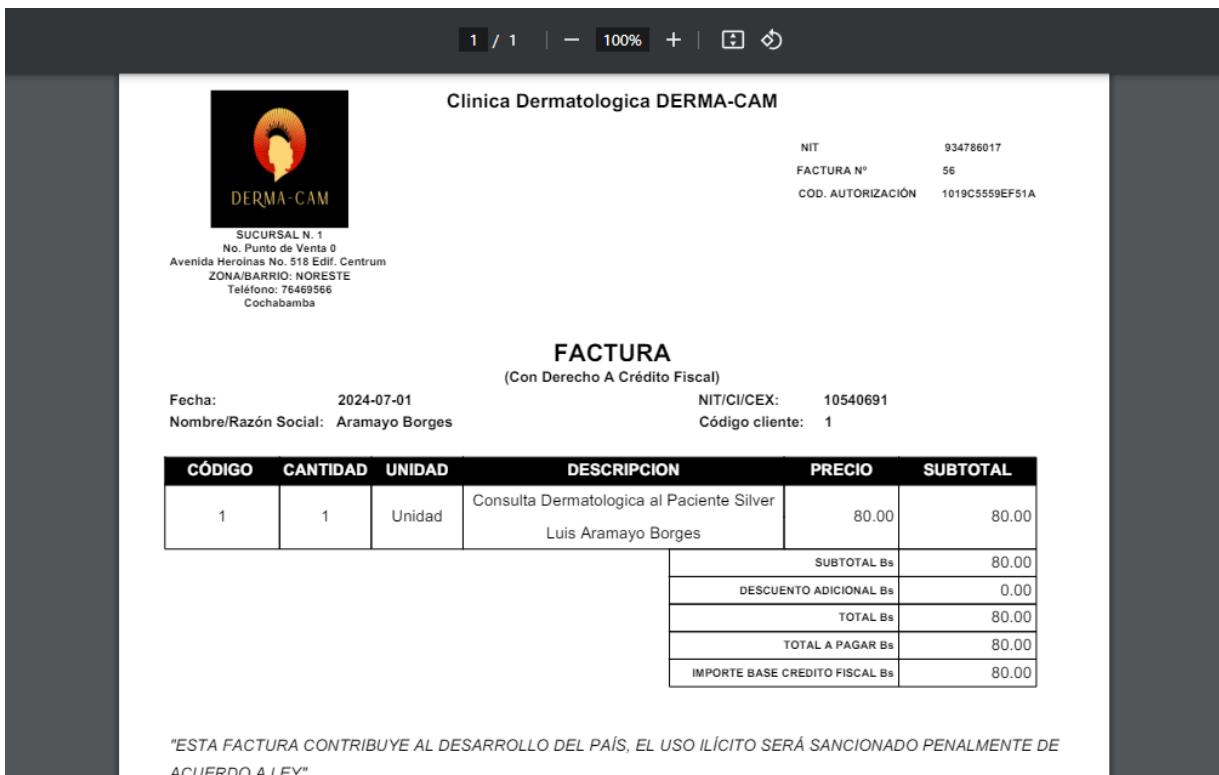
ILUSTRACIÓN N° 45: VER HISTORIAL DE LA CITA



FUENTE: Elaboración propia, en base a Ver historial, 2024.

También se puede encontrar un botón de acción para generar la factura de la cita con los datos del paciente y la descripción del servicio, dicha factura tiene el formato que impuestos nacionales exige.

En la siguiente ilustración, se puede observar la factura generada. Parte del código en los anexos (ver Anexo III).

ILUSTRACIÓN N° 46: FACTURA DE LA CITA GENERADA

FUENTE: Elaboración propia, en base a Generar Factura,2024.

3.3.3.6. Gestión de productos

En este punto se puede observar el panel para gestionar los productos, donde se puede agregar nuevos productos, aumentar el stock, editar y eliminar los productos que comercializa la clínica.

En la siguiente ilustración se puede observar el panel de gestión de los productos. Parte del código en los anexos (ver Anexo III).

ILUSTRACIÓN N° 47: GESTOR DE PRODUCTOS

Id	Nombre	Descripción	Precio	Cantidad	
2	EUSERIN AQUAPHOR	Pomada Reparadora, acelera la regeneración de la piel.	250.00	5	
3	CICATRICURE	Gel para aplicar a cicatrices y estriás.	90.00	10	

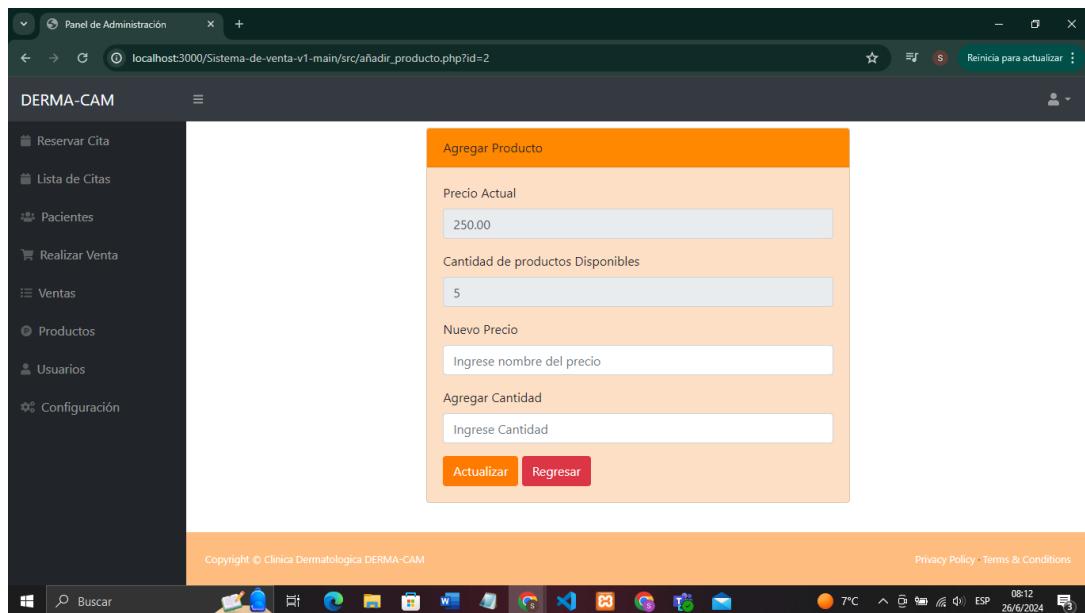
FUENTE: Elaboración propia, en base a Gestor de Productos,2024

Como todos los gestores tiene los botones de acción para realizar las funciones de crear, editar y eliminar, pero en este gestor tiene un botón que es para ingresar stock y también actualizar el precio del producto. Se puede observar parte del código de estas funcionalidades en los anexos (ver Anexo III).

ILUSTRACIÓN N° 48: CREAR PRODUCTO

Id	Nombre	Precio	Cantidad	
2	EUSERIN AQUA	250.00	5	
3	CICATRICURE	90.00	10	

FUENTE: Elaboración propia, en base a Crear Producto,2024

ILUSTRACIÓN N° 49: AGREGAR STOCK

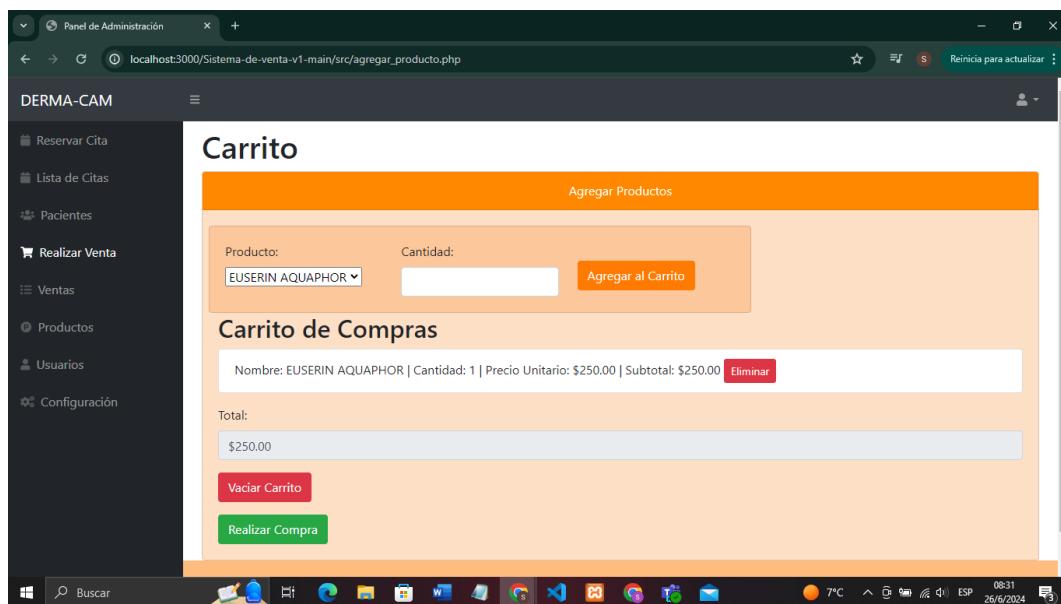
FUENTE: Elaboración propia, en base a Agregar stock,2024

3.3.3.7. Gestión de ventas

En este punto se podrá realizar la gestión de productos, como agregar y quitar productos al carrito, pasar a realizar la venta, buscar los datos del cliente también de una manera asíncrona ingresando tres letras de su nombre, realizar la venta y pasar a la lista de ventas en la cual estarán los datos de la venta y un botón de acción que genera la factura en pdf.

A continuación, se puede observar en la ilustración el carrito de compras, en el cual se agregan los productos y la cantidad de ellos para realizar una venta, donde se puede observar los productos agregados, los cuales también pueden ser eliminados por si ocurrió un error al agregarlos. También se observa el precio unitario, el subtotal y el total de la compra. Parte del código de esta funcionalidad en los anexos (ver Anexo III).

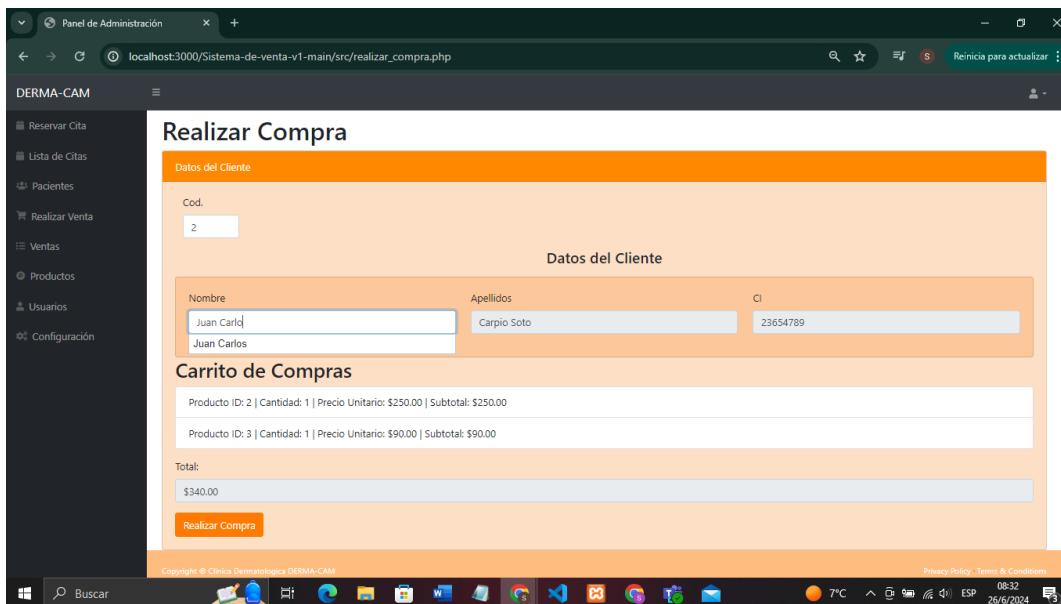
ILUSTRACIÓN N° 50: CARRITO DE COMPRAS



FUENTE: Elaboración propia, en base a Carrito de compras,2024

En realizar compra se realiza una búsqueda asíncrona del nombre del cliente para auto llenar su información y poder realizar la venta. Ver parte del código de esta funcionalidad en los anexos (ver Anexo III).

ILUSTRACIÓN N° 51: REALIZAR COMPRA



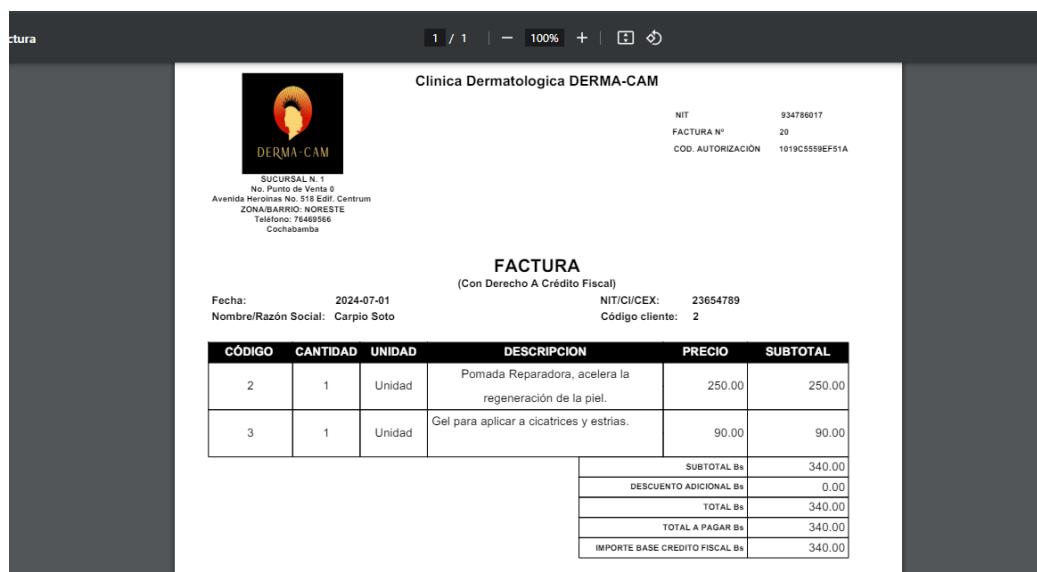
FUENTE: Elaboración propia, en base a Realizar compra,2024

ILUSTRACIÓN N° 52: LISTA DE VENTAS

#	Cliente	Apellidos	Total	Fecha	
17	Jose	Martinez	250.00	2024-06-26	
16	Silver Luis	Aramayo Borges	340.00	2024-06-26	
15	Jose	Martinez	340.00	2024-06-26	
14	Juan Carlos	Carpio Soto	250.00	2024-06-26	
13	Silver Luis	Aramayo Borges	0.00	2024-06-26	
12	Silver Luis	Aramayo Borges	250.00	2024-06-26	
11	Silver Luis	Aramayo Borges	250.00	2024-06-26	
10	Silver Luis	Aramayo Borges	340.00	2024-06-26	

FUENTE: Elaboración propia, en base a Realizar compra,2024

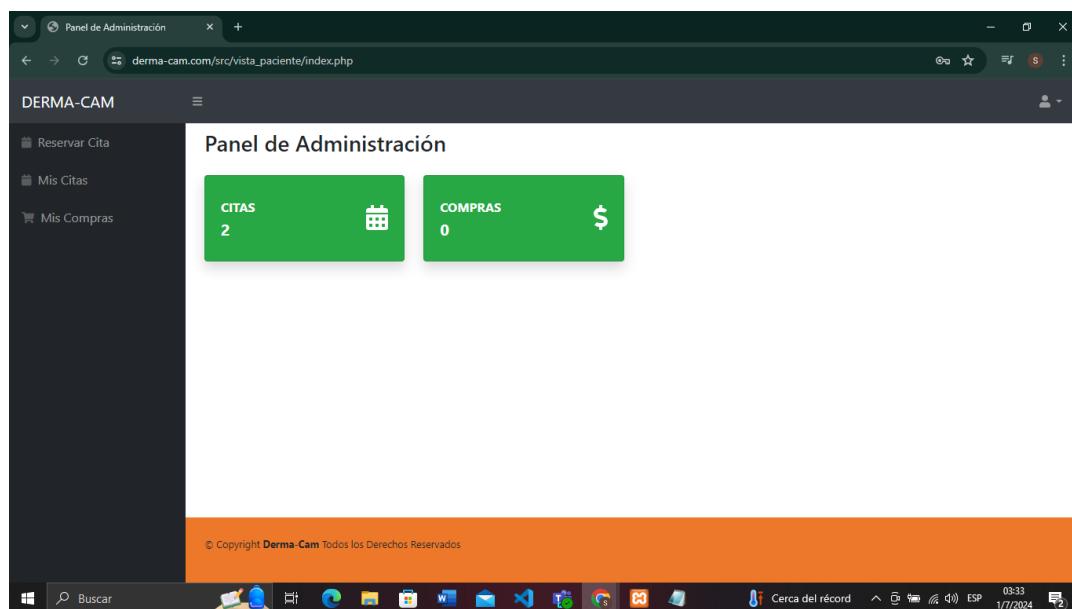
Pulsando el botón de acción de la venta generará un pdf de la factura en la cual se mostrará la información del cliente y los datos de los productos. Ver parte del código de esta funcionalidad en los anexos (ver Anexo III).

Ilustración N° 53: PDF DE LA FACTURA**FUENTE:** Elaboración propia, en base a Pdf factura,2024

3.3.3.8. Vistas para pacientes

Los pacientes una vez realicen su inicio de sesión tendrán acceso a su información personal en el apartado de panel de administración como ser sus citas, donde podrán ver información de sus citas y las facturas de cada cita y sus compras donde podrán ver las compras que realizaron en la clínica y su respectiva factura. Por otro lado, también pueden reservar sus propias citas, recibiendo también el mensaje de confirmación mediante WhatsApp. Parte del código en los anexos (ver Anexo III).

ILUSTRACIÓN N° 54: PANEL DE ADMINISTRACIÓN CLIENTE



FUENTE: Elaboración propia, en base Vistas del Cliente,2024

En el apartado de mis citas y compras solo podrá ver información que le compete al paciente como sus facturas.

ILUSTRACIÓN N° 55: CITAS VISTA PACIENTE

Fecha	Hora	Tipo	Estado	Acciones
2024-07-01	16:00	reconsulta	Realizado	Ver Historial
2024-07-01	16:15		Reservado	Cancelar Cita

FUENTE: Elaboración propia, en base Vistas del Cliente,2024

En la siguiente ilustración se puede observar la lista de compras desde la vista del paciente.

ILUSTRACIÓN N° 56: COMPRAS VISTA CLIENTE

#	Total	Fecha	Factura
21	340.00	2024-07-01	

FUENTE: Elaboración propia, en base Vistas del Cliente,2024

3.3.4. Adaptabilidad del sistema

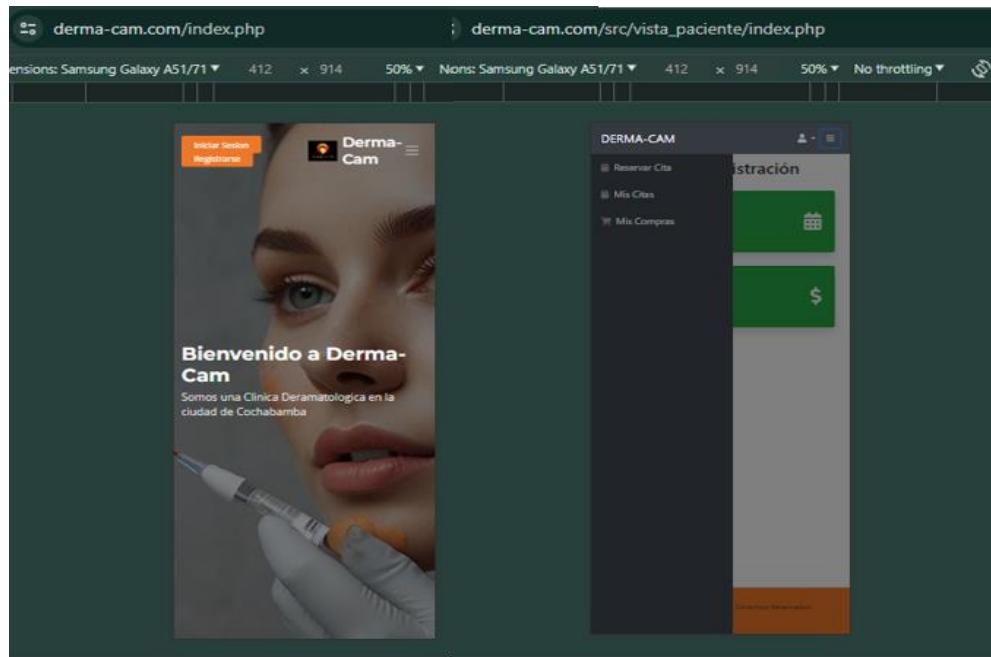
Este punto hace referencia a la capacidad del sistema de poder adaptarse a diferentes dispositivos como pc, laptop, tabletas y teléfonos inteligentes, garantizando el funcionamiento y visualización del sistema en diferentes buscadores.

➤ Diseño responsive del Sistema Web

En el sistema web se implementó el diseño responsive en cada una de las vistas, para así tener mejor adaptabilidad en diferentes tamaños de pantalla como en diferentes dispositivos que cuenten con un buscador.

En el sistema web, se implementó un diseño responsive utilizando el framework Bootstrap, que proporciona una serie de clases CSS y componentes que facilitan la creación de interfaces adaptables a diferentes tamaños de pantalla y dispositivos. Cada una de las vistas del sistema fue diseñada para ajustarse de manera óptima a pantallas de distintos tamaños, desde dispositivos móviles hasta monitores de escritorio, como se puede observar en la siguiente ilustración.

ILUSTRACIÓN N° 57: DISEÑO RESPONSIVE PARA MÓVILES



FUENTE: Elaboración propia, en base Diseño Responsive,2024

3.4. PRUEBAS DE VALIDACIÓN

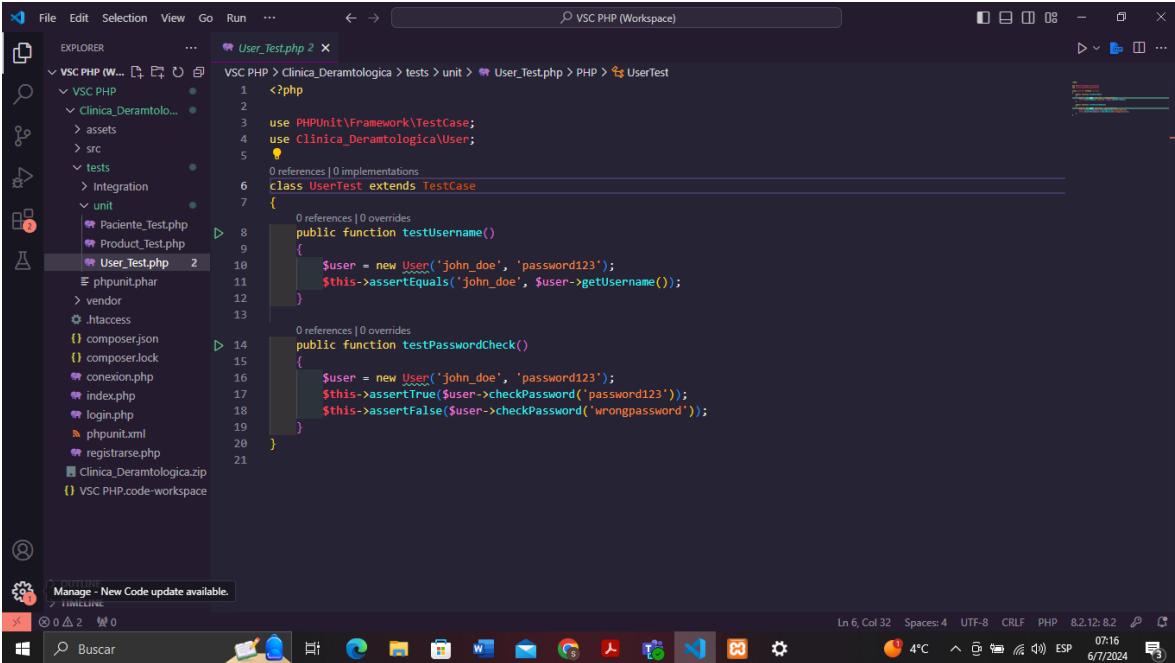
En esta parte del proyecto, se mostrarán los resultados de las pruebas realizada al sistema, las cuales se realizaron para verificar el correcto funcionamiento tanto de los componentes como del sistema en sí. En este paso se realizaron diferentes pruebas que ayudaron a identificar errores y funcionamientos inadecuados del sistema y así corregirlos.

3.4.3. Pruebas Unitarias

Durante el desarrollo del sistema de gestión para la clínica dermatológica, se implementaron pruebas unitarias utilizando PHPUnit, un framework de pruebas unitarias para PHP. Estas pruebas unitarias fueron fundamentales para asegurar la correctitud y calidad del código. Cada componente crítico del sistema, como usuarios, pacientes y productos, fue sometido a pruebas rigurosas. Por ejemplo, se verificó que la creación de usuarios y pacientes funcionara correctamente, así como la gestión de contraseñas y la integridad de los datos.

En la siguiente ilustración, se puede ver parte del código que se utilizó en las pruebas unitarias. El código mostrado realiza la primera prueba unitaria, validando la funcionalidad de creación de un nuevo usuario en el sistema.

ILUSTRACIÓN N° 58: CODIGO DE LAS PRUEBAS UNITARIAS



```
<?php
use PHPUnit\Framework\TestCase;
use Clinica_Deramologica\User;

class UserTest extends TestCase
{
    public function testUsername()
    {
        $user = new User('john_doe', 'password123');
        $this->assertEquals('john_doe', $user->getUsername());
    }

    public function testPasswordCheck()
    {
        $user = new User('john_doe', 'password123');
        $this->assertTrue($user->checkPassword('password123'));
        $this->assertFalse($user->checkPassword('wrongpassword'));
    }
}
```

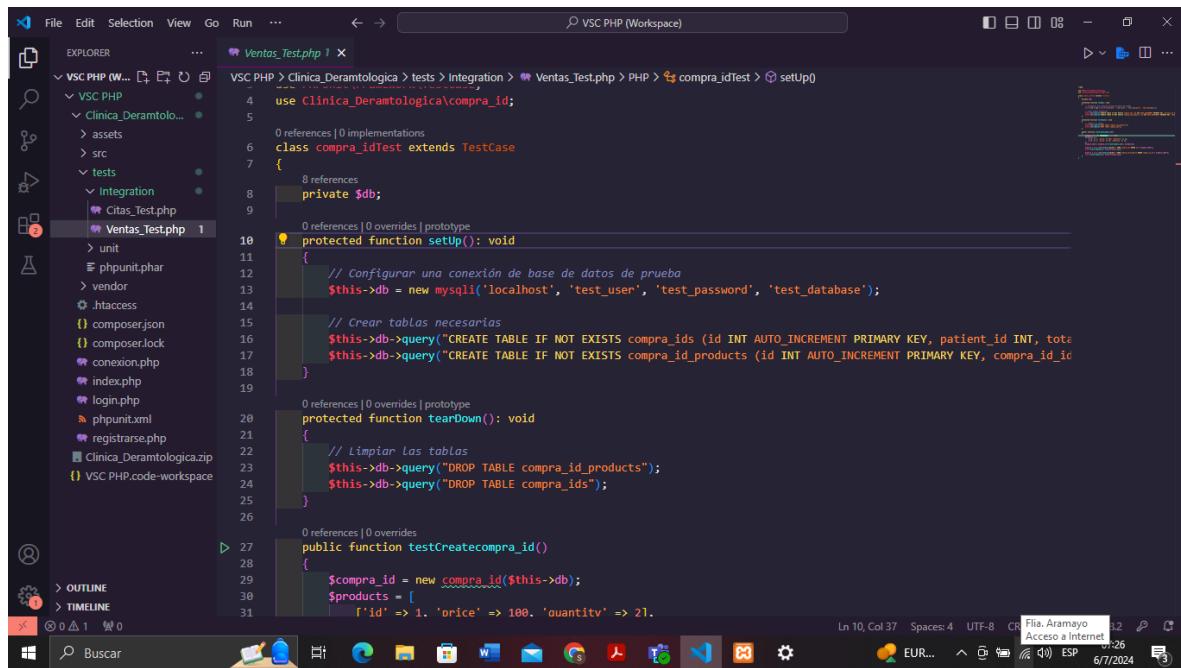
FUENTE: Elaboración propia, en base a las pruebas unitarias.

3.4.4. Pruebas de Integración

Durante la fase de desarrollo del sistema de gestión para la clínica dermatológica, se implementaron pruebas de integración para garantizar que los distintos módulos del sistema funcionaran de manera cohesiva y sin conflictos. Utilizando PHPUnit, se crearon casos de prueba que simularon interacciones reales entre los componentes del sistema, como la integración de la gestión de citas con la base de datos de pacientes y la funcionalidad de notificación a través de WhatsApp. Estas pruebas de integración fueron esenciales para verificar que los datos fluían correctamente entre los módulos y que las operaciones críticas, como la creación de citas y la actualización de información de los pacientes, se ejecutaban sin errores.

En la siguiente ilustración, se puede ver parte del código que se utilizó en las pruebas de integración. El código mostrado realiza la primera prueba de integración, validando el proceso completo de una venta en el sistema, desde la selección de productos hasta la confirmación de la venta.

ILUSTRACIÓN N° 59: CODIGO DE LAS PRUEBAS DE INTEGRACIÓN



The screenshot shows a Microsoft Visual Studio Code (VS Code) interface with a dark theme. The left sidebar displays a file tree for a 'VSC PHP' workspace, including files like 'Clinica_Deramtologica.php', 'index.php', 'login.php', 'registerse.php', and several test files such as 'Citas_Test.php', 'Ventas_Test.php', and 'compra_idTest.php'. The main editor area shows a portion of the 'Ventas_Test.php' file, which contains PHP code for a unit test class named 'compra_idTest' extending 'TestCase'. The code includes methods for setting up the database connection and tables, and a public method 'testCreatecompra_id()' that creates a new purchase object and adds products to it. The status bar at the bottom provides information about the current file ('Fila. Aramayo'), cursor position ('Ln 10, Col 37'), and date ('6/7/2024').

```

use Clinica_Deramtologica\compra_id;
class compra_idTest extends TestCase
{
    protected function setUp(): void
    {
        // Configurar una conexión de base de datos de prueba
        $this->db = new mysqli('localhost', 'test_user', 'test_password', 'test_database');

        // Crear tablas necesarias
        $this->db->query("CREATE TABLE IF NOT EXISTS compra_ids (id INT AUTO_INCREMENT PRIMARY KEY, patient_id INT, total DECIMAL(10, 2))");
        $this->db->query("CREATE TABLE IF NOT EXISTS compra_id_products (id INT AUTO_INCREMENT PRIMARY KEY, compra_id INT, product_id INT, quantity INT, price DECIMAL(10, 2))");

        // Limpiar las tablas
        $this->db->query("DROP TABLE compra_id_products");
        $this->db->query("DROP TABLE compra_ids");
    }

    public function testCreatecompra_id()
    {
        $compra_id = new compra_id($this->db);
        $products = [
            ['id' => 1, 'price' => 100, 'quantity' => 2];
    }
}

```

FUENTE: Elaboración propia, en base a las pruebas de integración.

3.4.5. Testim

Para realizar las pruebas end to end del sistema web se usó la aplicación Testim, la cual es una herramienta en base a inteligencia artificial la cual ayuda a la realizar pruebas automatizadas end to end ahorrando al desarrollador tiempo y costos, acelerando la creación de pruebas.

Testim es útil para realizar pruebas en aplicaciones web y móvil, realizando las pruebas por captura de la pantalla en video, mediante machine learning. También realiza pruebas en diferentes navegadores para comprobar la compatibilidad de la aplicación.

ILUSTRACIÓN N° 60: LOGO TESTIM



FUENTE: https://files.readme.io/e37ef02-small-docs_logo2x.png.

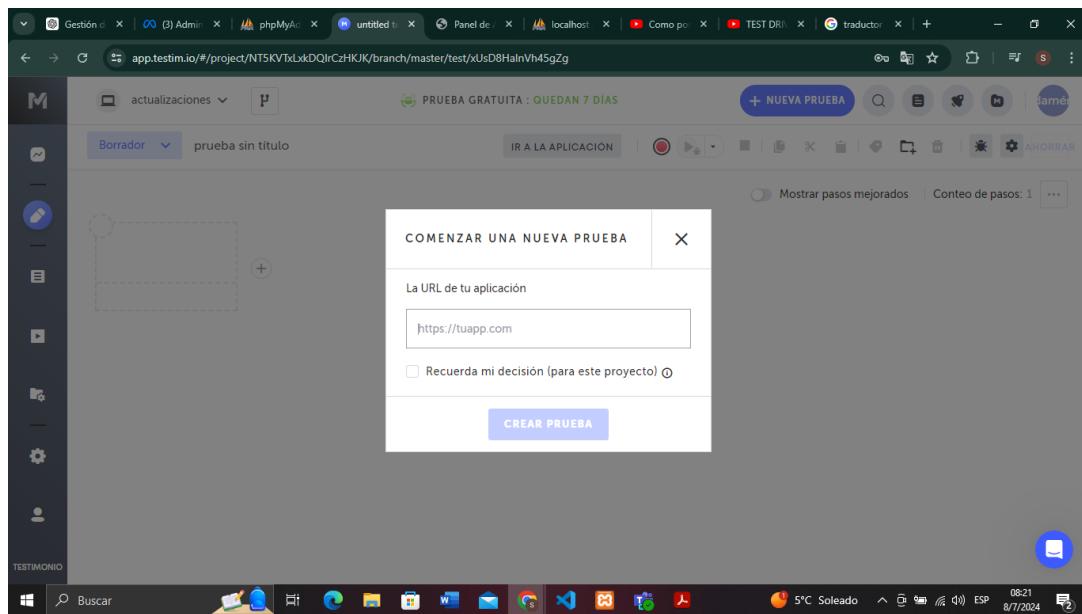
Los pasos para realizar pruebas con la ayuda de la herramienta Testim son los siguientes:

- **Crear una nueva prueba:** Ingresar al dashboard de Testim, hacer clic en nueva prueba y la herramienta crea un borrador para realizar la prueba.

ILUSTRACIÓN N° 61: CREAR UNA PRUEBA EN TESTIM

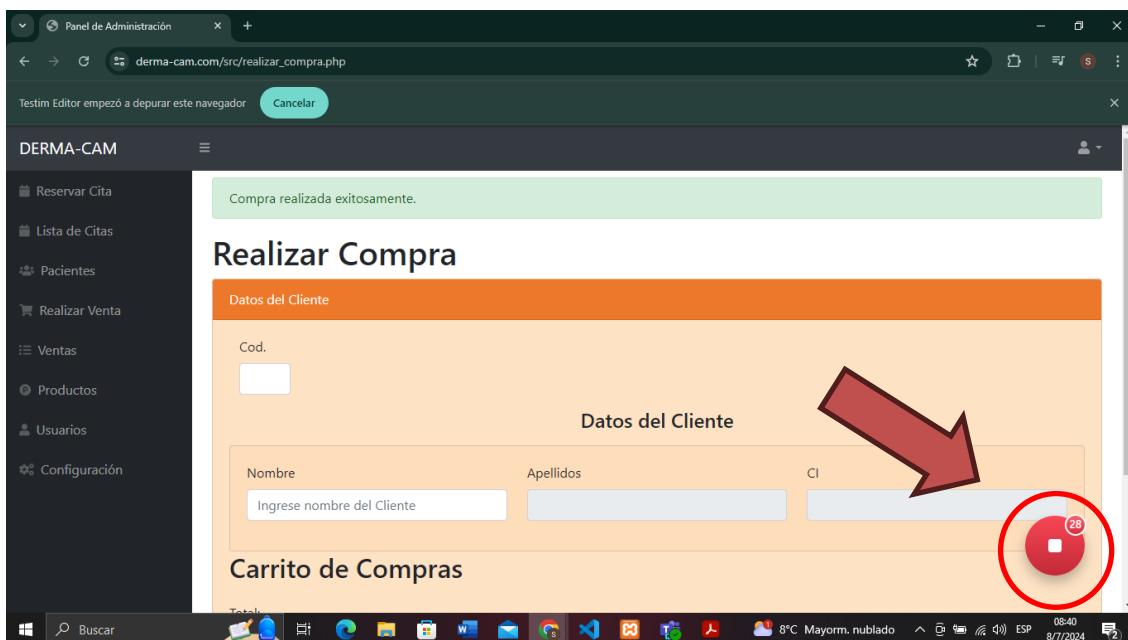
FUENTE: Elaboración propia, en base a Testim.

- **Comenzar la prueba:** En el panel del borrador se encuentra el botón de comenzar nueva prueba, con lo cual solicita la url de la aplicación o sistema, el cual puede ser tanto del localhost como del dominio del sistema ya desplegado, con este url inicia la grabación en la cual Testim registrara los pasos que se realizan para ejecutar una funcionalidad.

ILUSTRACIÓN N° 62: SOLICITUD DE LA URL DEL SISTEMA

FUENTE: Elaboración propia, en base a Testim.

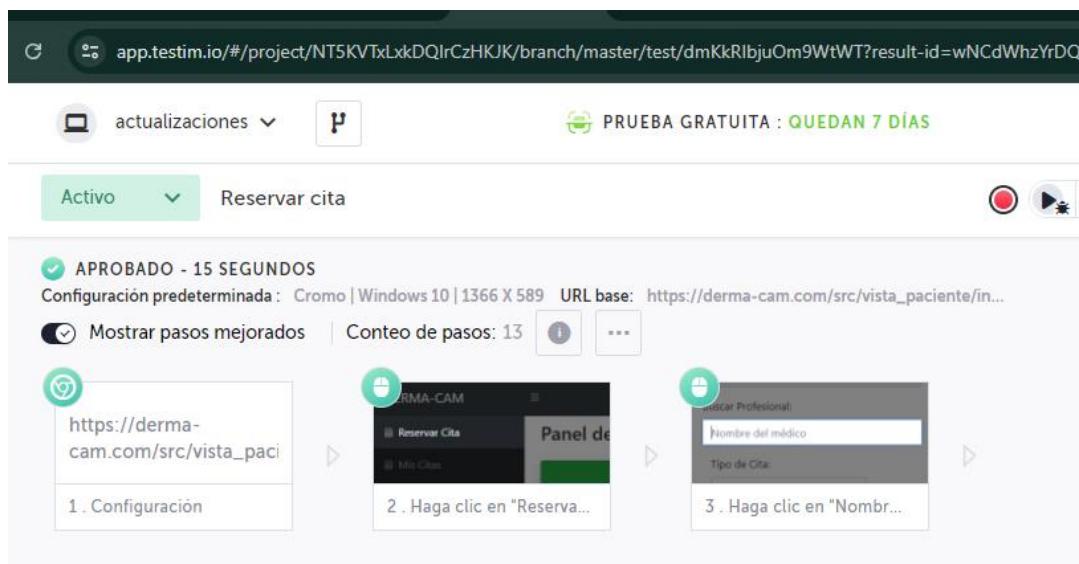
- **Recopilación de pasos:** Posteriormente al iniciar la grabación Testim, con la ayuda de la inteligencia artificial detecta los pasos que realizamos para cumplir con la funcionalidad,

ILUSTRACIÓN N° 63: RECOPILACIÓN DE PASOS

FUENTE: Elaboración propia, en base a Testim.

- **Ejecución de prueba:** Una vez recopilados los pasos que se necesitan para realizar una funcionalidad, se ejecuta la prueba para verificar su correcto funcionamiento.

ILUSTRACIÓN N° 64: EJECUCION DE LA PRUEBA APROBADA



FUENTE: Elaboración propia, en base a Testim.

3.4.5.1. Pruebas end to end (E2E)

Con la aplicación Testim se realizaron las pruebas end to end para evaluar el rendimiento y funcionamiento completo del sistema. Gracias a la aplicación se logró realizar diferentes simulaciones en todos los apartados del sistema, como la navegación en la landing page, inicio de sesión, registro de pacientes, navegación del paciente en su panel de administración, reserva de citas, realizar ventas, generación de facturas, etc.

3.4.5.2. Escenarios de Prueba

- **Navegación landing page:** Se realizó una prueba de la navegación de la landing page, comprobando las acciones que realiza.
- **Inicio y registro de usuarios y pacientes:** Se realizó el registro y el inicio de los actores que interactúan con esta función.

- Reserva de citas: Se probó la función de reserva de citas tanto de la parte del usuario como del paciente. Dando como resultado la correcta reserva.
- Agregar productos y realizar venta: Se puso a prueba el apartado de ventas dando como resultado que los productos se agregan de manera correcta, también la búsqueda y agregado de la información para realizar la venta.
- Generación de facturas: Se verificó la generación de las facturas tanto de ventas como de las citas.

En la siguiente ilustración se pueden observar algunas de las pruebas realizadas en la herramienta Testim.

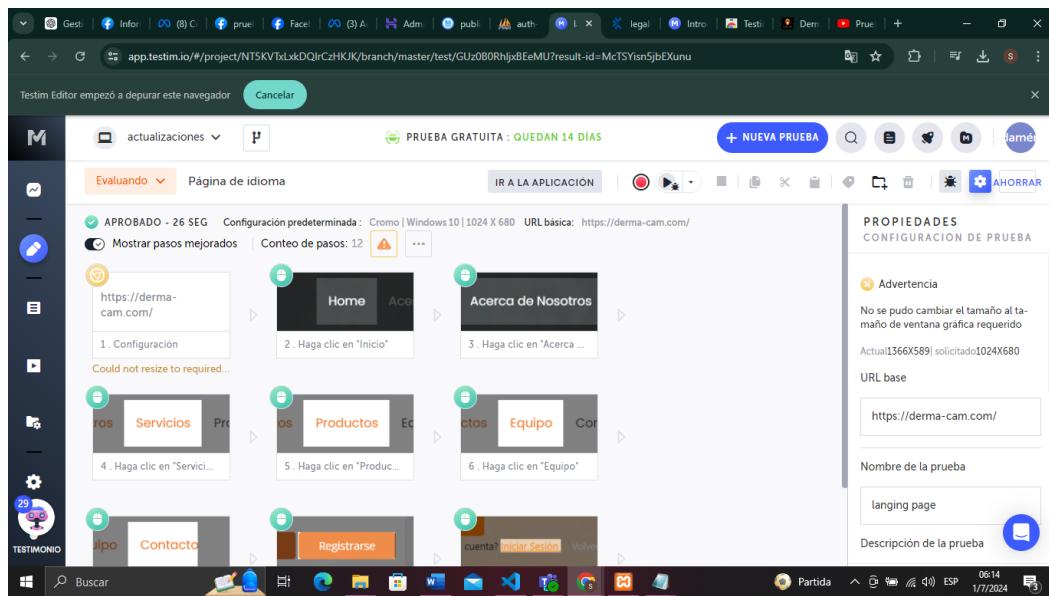
Ilustración N° 65: PRUEBAS GUARDAS EN TESTIM

The screenshot shows the Testim application interface. On the left, there's a sidebar with icons for 'actualizaciones', 'PRUEBAS' (selected), 'SUITES', 'PLANES', 'ETIQUETAS', and 'PASOS COMPARTIDOS'. The main area has a header with 'PRUEBA GRATUITA : QUEDAN 7 DÍAS' and a '+ NUEVA PRUEBA' button. Below this is a search bar and a 'guardar vista' button. The central part displays a table titled 'Biblioteca de pruebas (5)' with columns: NOMBRE, DUEÑO, ACTUALIZADO, AMABLE, ETIQUETA, ESTADO, and ÚLTIMAS CARRER. The table lists five test cases:

NOMBRE	DUEÑO	ACTUALIZADO	AMABLE	ETIQUETA	ESTADO	ÚLTIMAS CARRER
Generar PDF	Aramayo plateado	hace unos s...	prueba	Borrador	Borrador	[progress bar]
Menú e interacciones del Paciente	Aramayo plateado	Hace 7 días	prueba	Activo	Activo	[progress bar]
Navegación en la Landing Page	Aramayo plateado	hace 10 min...	prueba	Activo	Activo	[progress bar]
Realizar Venta	Aramayo plateado	hace un mi...	prueba	Activo	Activo	[progress bar]
Reservar cita	Aramayo plateado	Hace 6 min...	prueba	Activo	Activo	[progress bar]

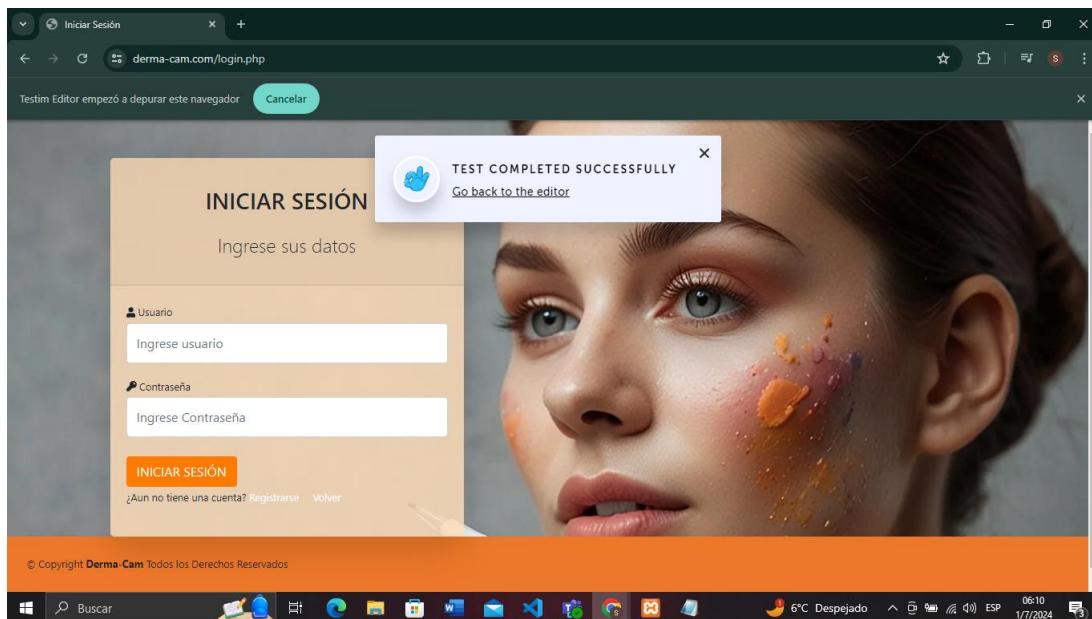
FUENTE: Elaboración propia, en base Pruebas en Testim,2024

A continuación, se puede observar los pasos que siguió la herramienta tras la grabación del ejemplo que se realizó.

ILUSTRACIÓN N° 66: PRUEBAS DESDE LA INTERFAZ DE TESTIM

FUENTE: Elaboración propia, en base Interfaz Testim,2024

En la siguiente ilustración se puede observar, que después de ejecutar la prueba en la cual la herramienta sigue los pasos grabados, el test fue completado satisfactoriamente.

ILUSTRACIÓN N° 67: MENSAJE DE TEST COMPLETADO

FUENTE: Elaboración propia, en base Interfaz Testim,2024

Testim ha demostrado ser una herramienta invaluable en este proyecto debido a su capacidad para automatizar pruebas de end to end de manera eficiente y efectiva. Su interfaz intuitiva y basada en inteligencia artificial permite crear y gestionar pruebas con facilidad, reduciendo significativamente el tiempo y esfuerzo necesarios para garantizar la calidad del software. Gracias a Testim, se pudieron identificar y resolver rápidamente errores y problemas potenciales antes de que llegaran a los usuarios finales, asegurando así una experiencia de usuario fluida y sin interrupciones.

3.5. DESPLIEGUE DEL SISTEMA

Este punto corresponde a la última fase de la metodología en cascada, la cual consiste en el despliegue del sistema en un entorno de producción para su posterior uso por parte de los usuarios finales. Durante esta fase, el sistema fue implementado en un hosting, permitiendo poner el sistema en funcionamiento de manera efectiva y asegurando que esté disponible para los usuarios de la clínica.

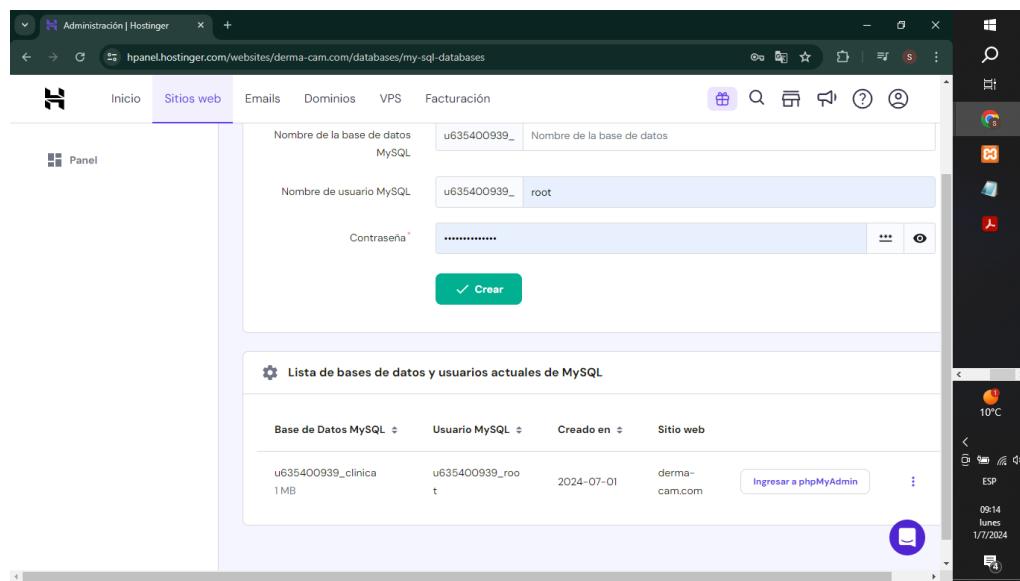
3.5.5. Hostinger

Para el despliegue del sistema web de la “Clínica Dermatológica DERMA-CAM”, se usó el proveedor de alojamiento Hostinger.

Hostinger es una de las mejores opciones para el despliegue del sistema, ya que este proporciona el hosting de los archivos, un dominio, certificado SSL y garantiza su disponibilidad en línea. Los pasos que se siguieron para el despliegue fueron:

- **Crear la base de datos:** Para esto se realizó la exportación de nuestra base de datos con la cual se creó la nueva en Hostinger.

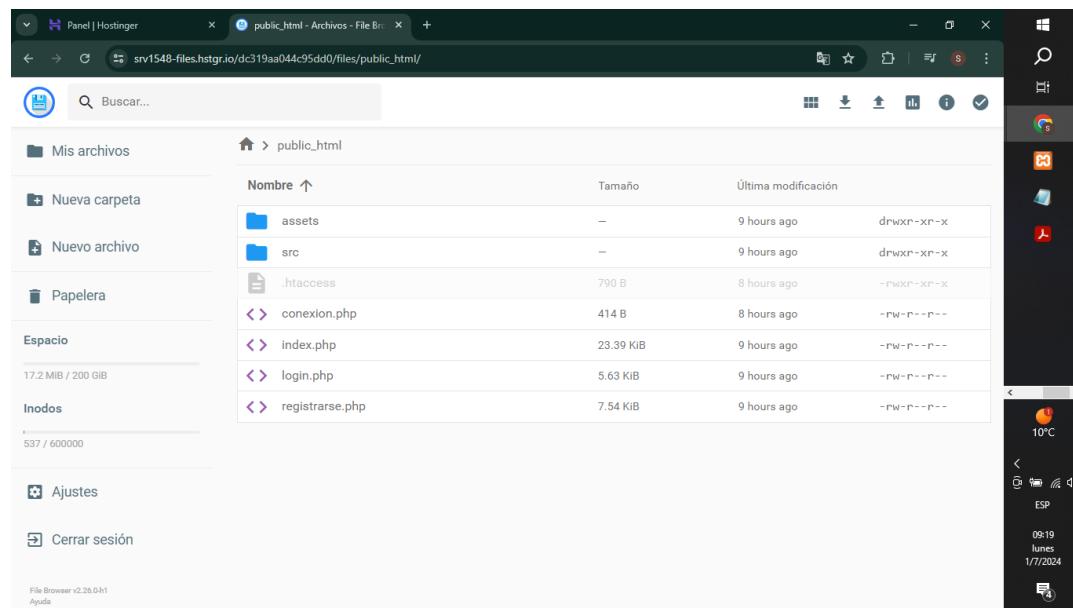
ILUSTRACIÓN N° 68: Base de Datos en Hostinger



FUENTE: Elaboración propia, en base Hostinger,2024

- **Subir archivos:** En este punto se subieron los archivos que conforman el sistema al administrador de archivos del hosting.

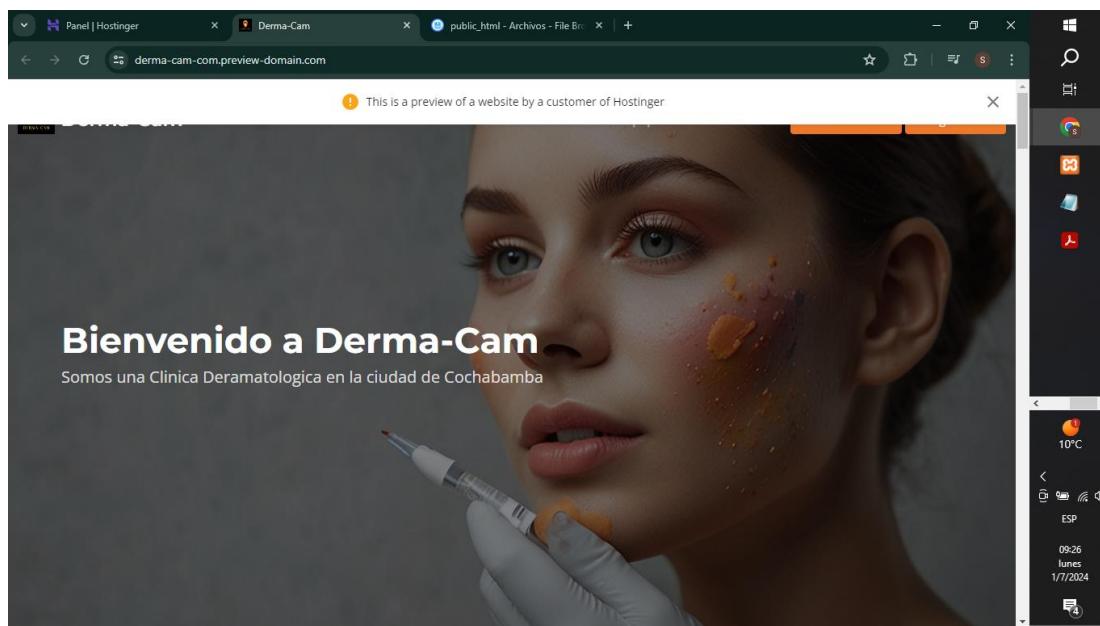
ILUSTRACIÓN N° 69: Archivos del Sistema en el Hosting



FUENTE: Elaboración propia, en base Hostinger,2024

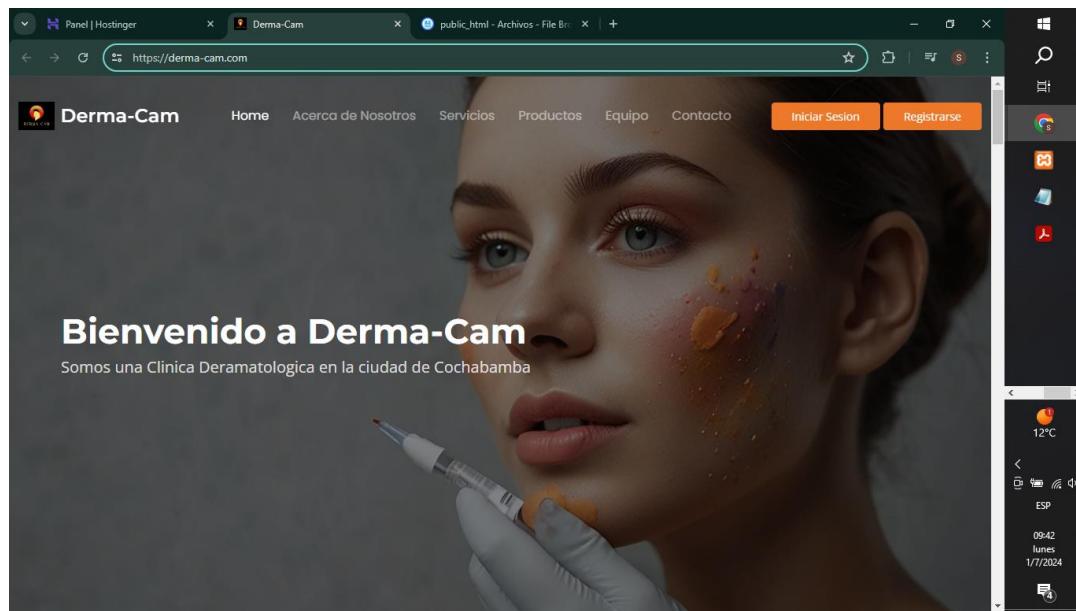
Posteriormente el servicio de hosting realizo el despliegue del sistema, permitiendo su disponibilidad y accesibilidad para los usuarios finales, esto de manera automatizada.

ILUSTRACIÓN N° 70: Vista previa del sistema en Hostinger



FUENTE: Elaboración propia, en base Vista Previa del Sistema,2024.

En este punto se conectó el dominio que se adquirió para el Sistema Web de la Clínica Dermatológica DERMA-CAM, el cual se personalizo de acuerdo con la clínica dándole carácter único y profesional. Con lo cual los usuarios pueden encontrar y visitar el sitio ingresando al dominio “derma-cam.com” desde cualquier navegador en cualquier dispositivo, lo cual mejoró la visibilidad de la clínica.

ILUSTRACIÓN N° 71: Sistema Desde el Domino

FUENTE: Elaboración propia, en base derma-cam.com,2024.

Por último, se presentó el certificado SSL, el cuales esencial para proteger la información de los usuarios, asegurar la autenticidad del sitio web, aumentar la confianza del usuario, cumplir con las normativas y garantizar la integridad de los datos. Gracias a este certificado la información que intercambian los clientes con el servidor del sistema Web “Derma-Cam” se encuentra encriptada, evitando que sean interceptadas.

ILUSTRACIÓN N° 72: Certificado SSL

FUENTE: Elaboración propia, en base derma-cam.com,2024.

CAPITULO IV

ANÁLISIS DE

VIABILIDAD

ANÁLISIS DE VIABILIDAD

4.1 VIABILIDAD TÉCNICA

Para el correcto funcionamiento de del sistema, web son necesarios los siguientes requerimientos técnicos, tanto para los usuarios como para los proveedores, estos están en base al optimo funcionamiento y el maximizar los recursos.

TABLA N° 19: REQUERIMIENTOS TÉCNICOS DE HARDWARE Y SOFTWARE PARA EL USO DEL SISTEMA

Requerimiento	Descripción
Computador de escritorio o portátil	Dispositivo por el cual el usuario puede ingresar al sistema mediante el navegador de su preferencia, también se puede ingresar mediante un dispositivo móvil que tenga navegador web.
Procesador	Dual-core o superior
Memoria RAM	Al menos 2 GB.
Sistema Operativo	Windows, macOS, Linux
Internet	Conexión estable al internet, para poder acceder al sistema web mediante cualquier buscador.

FUENTE: Elaboración Propia en base a Requerimientos Técnicos (2024).

TABLA N° 20: REQUERIMIENTOS TÉCNICOS DE HARDWARE Y SOFTWARE PARA EL FUNCIONAMIENTO DEL SISTEMA

Requerimiento	Descripción
Servidor Web	Servidor compartido donde se alojó el sistema, el cual contiene instalada la tecnología necesaria para su funcionamiento, como ejemplo están Apache o Nginx.
Procesador	CPU con al menos 2 núcleos.
Memoria RAM	Mínimo 4 GB, recomendado 8 GB o más.
Almacenamiento:	SSD de al menos 100 GB para mejorar la velocidad de acceso a los datos.

Requerimiento	Descripción
Sistema Operativo	Linux (preferido para servidores web), Windows Server o macOS.
PHP	Versión 7.4 o superior.
Base de Datos	MySQL 5.7 o superior, o MariaDB 10.3 o superior.
Dominio	Es el nombre único que identifica a un sitio web en Internet.

FUENTE: Elaboración Propia en base a Requerimientos Técnicos (2024).

4.2. VIABILIDAD ECONÓMICA

La viabilidad económica del proyecto es crucial para determinar el costo de este y así analizar si es rentable a largo plazo, con esto también poder realizar la toma de decisiones.

4.2.1. Estimación del costo del proyecto

Para realizar la estimación del costo se hizo uso del método Albrecht para el análisis de los puntos de función.

4.2.1.1. Identificación y Clasificación de los Componentes

➤ Entradas Externas (EE)

- Crear cita
- Actualizar cita
- Cancelar cita
- Registrar paciente
- Buscar paciente
- Buscar médico

➤ Salidas Externas (SE)

- Generar informe de citas
- Enviar confirmación de cita por WhatsApp
- Generar Factura

➤ Consultas Externas (CE)

- Consultar disponibilidad de citas

- Consultar historial de citas
 - Generar Factura
- **Archivos Lógicos Internos (ALI)**
- Tabla de pacientes
 - Tabla de citas
 - Tabla de usuarios
- **Archivos de Interfaz Externos (AIE)**
- integración con API de WhatsApp
 - Integración con sistema de facturación

La fórmula del método Albrecht para encontrar el punto de función es:

$$PF = TPF * (0.65 + 0.01 * \sum F_i \text{ } 14 \text{ } i = 1)$$

TABLA N° 21: PUNTOS DE FUNCION SISTEMA WEB

Descripción	Nro.	Sencilla	Media	Compleja	Total (PF)
Nro. de Entrada Externas	6	3	4	6	24
Nro. de Salidas Externas	3	4	5	7	15
Nro. de Consultas Externas	3	7	10	15	6
Nro. de Archivos Lógicos Internos	3	5	7	10	24
Nro. de Archivos de Interfaz Externos	2	3	4	6	10
TOTAL, PUNTOS DE FUNCIÓN NO AJUSTADOS					79

FUENTE: Elaboración Propia, en base a Puntos de Función (2024).

4.2.1.2. Puntos de Función Ajustado

Para realizar el Método Albrecht para el análisis de puntos función, se utiliza la Tabla de Factor de Ajuste, también conocida como el Factor de Ajuste de Complejidad. Este factor se utiliza para ajustar el valor bruto de los puntos de función basándose en 14 características de complejidad del sistema, según la siguiente escala.

TABLA N° 22: ESCALA DE INFLUENCIA FACTOR AJUSTE

0	1	2	3	4	5
Sin Influencia	Poca Influencia	Moderado	Medio	Significativo	Esencial

FUENTE: Elaboración Propia, en base a Escala de Factor Ajuste, (2024).

Según la escala se definió la influencia en el factor ajuste en las 14 características.

TABLA N° 23: AJUSTE DE LOS PUNTOS DE FUNCIÓN

Nro.	Característica de Complejidad	Influencia
1	Comunicación de datos	3
2	Procesamiento distribuido	2
3	Rendimiento	4
4	Configuración robusta de hardware	2
5	Volumen transaccional	3
6	Entrada de datos en línea	4
7	Eficiencia del usuario final	3
8	Actualización en línea	3
9	Complejidad lógica interna	2
10	Reusabilidad del código	1
11	Facilidad de instalación	3
12	Facilidad de operación	4
13	Múltiples sitios de operación	2
14	Facilidad de cambio	3
TOTAL		39

FUENTE: Elaboración Propia, en base a Ajuste de Puntos de Función, (2024).

Una vez encontrado el resultado de la sumatoria de los puntos de función y los puntos de función ajustados en base a la influencia se remplaza en la formula encontrando el punto de función para el desarrollo del sistema.

$$PF = 79 * (0.65 + 0.01 * 39)$$

$$PF = 82$$

Con el punto de función para el desarrollo hallado, se procede a realizar la estimación en horas hombre para poder determinar el costo del desarrollo del sistema.

4.2.1.3. Factor Productividad

El factor de productividad para el lenguaje de programación, en el caso del sistema web es PHP puede variar, pero un valor comúnmente aceptado es entre 10 y 15 horas por Punto de Función. Utilizaremos un valor promedio de 12 horas por Punto de Función.

TABLA N° 24: FACTOR PRODUCTIVIDAD

Lenguaje	Horas Puntos de Función Promedio	Líneas de código por Punto de Función
Lenguaje de 4ta Gen.	8	25

FUENTE: Elaboración Propia, en base al Factor Productividad, (2024).

4.2.1.4. Estimación Horas Hombre

En base a la información encontrada como los puntos de función para el desarrollo y las horas por puntos de función promedio según el lenguaje de programación se calcula las horas hombre para el desarrollo del sistema web.

- **HH:** Horas Hombre
- **PFA:** Puntos de Función Ajustados
- **HPFP:** Horas Puntos de Función Promedio
- **HD:** Horas Dia
- **DT:** Días de Trabajo
- **DM:** Días Mes
- **MT:** Mese de Trabajo

$$HH = PFA * HPFP$$

$$HH = 82 * 8$$

$$HH = 656 \text{ Horas Hombre}$$

Con el cálculo de las horas hombre se pasa a calcular la duración del proyecto.

$$HH / HD = DT$$

$$656 / 8 = 82 \text{ días de trabajo.}$$

$$DT / DM = MT$$

$82 / 21 = 3.90$ que es igual a 3 meses con 18 días de trabajo para desarrollar el sistema web de la Clínica Dermatológica “DERMA-CAM”.

El sueldo de los desarrolladores se obtuvo en base al sueldo mínimo de Bolivia, el cual es de 2500 Bs. desde el año 2024, el dato fue obtenido de la página del ministerio de economía y finanzas públicas del país.

TABLA N° 25: COSTO DEL DESARROLLO DEL PROYECTO

Horas/Persona/Mes	Tiempo de desarrollo (meses)	Numero de Desarrolladores (2500 Bs.)	Costo del Proyecto (Bs.)
168	3,9	1	5.500,00
Costo Total del Desarrollo del Proyecto en Bs.			9.750,00

FUENTE: Elaboración Propia, en base al Costo del Desarrollo del Proyecto, (2024).

4.2.2. Costos de Funcionamiento del Proyecto

En este punto se realizó el ajuste de los costos para el funcionamiento del sistema web para la Clínica Dermatológica “DERMA-CAM”, dividiendo los costos mensuales y anuales a pagar.

TABLA N° 26: COSTO MENSUAL DE FUNCIONAMIENTO

Recursos	Descripción	Costo (Bs.)
Servicio de Hosting	Servicio donde se aloja el proyecto para su funcionamiento	112,21
Internet	Plan de internet de 35Mbps	169,00
	Costo Total	291,21

FUENTE: Elaboración Propia, en base al Costos del Proyecto, (2024).

TABLA N° 27: COSTO ANUAL DE FUNCIONAMIENTO

Recursos	Descripción	Costo (Bs.)
Dominio	Domino por la cual el sistema será buscado e identificado en la web	61,05
	Costo Total	61,05

FUENTE: Elaboración Propia, en base al Costos del Proyecto, (2024).

4.2.3. Conclusión

Con este análisis se pudo llegar a la conclusión que, el costo para el desarrollo del sistema web para la Clínica Dermatológica DERMA-CAM en base a la metodología Albrecht, es de 9.642,90 Bs. En el cual solo se toma en cuenta el desarrollo de la del sistema.

Por otro lado, el costo mensual para el funcionamiento del sistema es de 291,21 Bs. En este se contempla el costo del Hosting y el Internet que es necesario para poder acceder al sistema web.

Y por último se tiene los gastos anuales para el funcionamiento del sistema el cual es de 61,05 Bs. Donde se contempla el costo del Dominio esencial para la identificación del sistema en los navegadores.

4.3. VIABILIDAD OPERATIVA

El 1 de Julio del 2024 a las 18:30 pm se hizo la presentación del sistema de web a la dueña de la Clínica Dermatológica DERMA-CAM, la Dra. Carolina Achá Morató, la cual junto a su asistente realizaron la prueba del sistema web desde sus equipos de cómputo en la clínica, mediante su navegador.

Posteriormente se aprobó el sistema para su uso en la clínica después de una capacitación de las funcionalidades que esta brinda, para así empezar a gestionar su información a través del sistema web.

CAPÍTULO V

CONCLUSIONES Y

RECOMENDACIONES

CONCLUSIONES Y RECOMENDACIONES**5.1. CONCLUSIONES**

Al finalizar la implementación del sistema web titulado “Implementación de un Sistema Web para mejorar la gestión y organización de la información en la Clínica Dermatológica DERMA - CAM” en la ciudad de Cochabamba durante la gestión 2024”, se alcanzaron con éxito todos los objetivos planteados. Gracias al cumplimiento de los objetivos específicos, se logró alcanzar el objetivo general, mediante la implementación de diversas funcionalidades en el sistema web.

El análisis exhaustivo de los requerimientos permitió identificar las necesidades específicas de la clínica dermatológica y definir claramente los componentes del sistema. Esto incluyó la identificación de las funcionalidades esenciales como la gestión de citas, el historial médico y la facturación. La recopilación de datos mediante entrevistas proporcionó una base sólida para el diseño del sistema, asegurando que se adaptara a las necesidades reales de los usuarios. Esta etapa crítica garantizó una comprensión profunda de los procesos internos de la clínica, lo que resultó en un sistema bien alineado con sus requerimientos operativos.

La fase de diseño fue crucial para estructurar los componentes del sistema de manera coherente y eficiente. Se crearon diagramas de casos de uso, secuencia, clases y componentes, lo que facilitó la visualización de la interacción entre los diferentes elementos del sistema. Este enfoque permitió identificar y resolver posibles problemas de integración antes de la fase de desarrollo, asegurando un flujo de trabajo más fluido y una implementación más rápida y efectiva. La planificación detallada en esta etapa fue fundamental para evitar contratiempos durante el desarrollo.

La implementación del sistema de acuerdo con los diseños y especificaciones establecidos resultó en un software funcional y robusto. Se utilizó PHP como lenguaje de programación principal y MySQL como sistema de gestión de bases de datos, lo que garantizó un rendimiento óptimo y una fácil escalabilidad.

CONCLUSIONES Y RECOMENDACIONES

Las pruebas exhaustivas, tanto unitarias como de integración, confirmaron que el sistema cumplía con los requisitos establecidos. Estas pruebas fueron fundamentales para identificar y corregir errores, mejorar la estabilidad del sistema y garantizar que todas las funcionalidades trabajaran en conjunto de manera armoniosa. La realización de pruebas en diferentes escenarios y condiciones operativas ayudó a garantizar que el sistema pudiera manejar adecuadamente las situaciones del mundo real.

El despliegue exitoso del sistema en un entorno de producción permitió a la clínica dermatológica comenzar a beneficiarse de inmediato de las mejoras en la gestión y organización de la información. La implementación en un servidor web facilitó el acceso a los usuarios finales y garantizó la disponibilidad del sistema. Además, se proporcionó capacitación al personal para asegurar una transición sin problemas y una adopción rápida del nuevo sistema.

5.2. RECOMENDACIONES

- Integrar la telemedicina como una funcionalidad del sistema, permitiendo a los pacientes realizar consultas médicas en línea. Esta funcionalidad es particularmente útil en situaciones donde el acceso físico a la clínica es limitado.
- Desarrollo de módulos adicionales como la gestión de inventarios y control de suministros médicos. Esto permitirá un control más exhaustivo y eficiente de los recursos disponibles en la clínica.
- Desarrollar una aplicación móvil complementaria al sistema web para facilitar el acceso a los servicios de la clínica desde dispositivos móviles. Esto aumentaría la accesibilidad y comodidad para los pacientes y el personal de la clínica.
- Implementar el uso de inteligencia artificial y análisis de datos para mejorar la toma de decisiones clínicas y personalizar los tratamientos para los pacientes.

CONCLUSIONES Y RECOMENDACIONES

- Implementar un sistema de feedback donde los pacientes puedan dejar comentarios y valoraciones sobre los servicios recibidos. Esto ayudará a identificar áreas de mejora y a mantener un alto nivel de calidad en los servicios.

- Mejorar la optimización para motores de búsqueda (SEO) del sistema web para aumentar la visibilidad en línea de la clínica y atraer a más pacientes potenciales a través de búsquedas orgánicas.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

- Agueldo, G., Aigneran, M., & Ruiz Restrepo, J. (2010). *La Sociología En Sus Escenarios*. Medellin: Univeridad de Antioquia.
- Becker, P. &. (2019). *Modern Development with Visual Studio Code*. Sebastopol: O'Reilly Media.
- Bergmann, S. (2014). *The PHPUnit Manual: Integration Testing with PHPUnit*. . Stuttgart: Sebastian Bergmann.
- Campos Monge, M., & Campos Monge, E. M. (2023). *Sistemas operativos sistemas informáticos y lenguajes de programación*. España: RA-MA Editorial.
- Del Águila Cano, I. (2022). *Fundamentos de Ingeniería de los Requisitos*. España: Editorial Universidad de Almería.
- Gómez, R. (2022). *La era del código*. España: Hub de Libros.
- Gonzales León, L. G. (2019). *Implementación de Sistema de Reservas de Citas Médicas en Línea*. Lima: Universidad San Ignacio de Loyola.
- HERNANDEZ, R., FERNANDEZ, C., & BAPTISTA, P. (2010). *Metodología de la Investigación*. Mexico: McGraw-Hill.
- Lawrence , S., & Atlee , J. (2006). *Software Engineering: Theory and Practice*. Reino Unido: Pearson Prentice Hall.
- Lozano Flores, R. (2014). *SISTEMA DE ADMINISTRACION Y CONTROL DE HISTORIALES CLINICOS PARA LOS CONSULTORIOS DE LA UMSA* . La Paz: Univesidad Mayor de San Andrés.
- Mardan, A. (2014). *Pro Express.js: Master Express.js: The Node.js Framework For Your Web Development*. Estados Unidos: Apress.
- Marqués, M. (2009). *Bases de datos*. España: Universitat Jaume I. Servei de Comunicació i Publicacions.
- Ministerio de Salud. (2014). *Manual de Usuario Sistema de Información Clínico Estadístico*. Sistema Nacional de información en Salud.
- Mohedano, J. (2013). *Iniciación a JavaScript*. España: Ministerio de Educación y Formación Profesional de España.
- Natsys. (2014). *Todo sobre MySQL*. Natsys.

BIBLIOGRAFÍA

- Organization, W. H. (2022). *Guía para la inversión en la implementación digital (DIIG): incorporación de intervenciones digitales en los programas de salud*. Suiza: World Health Organization.
- Penning, M. &. (2020). *Designing Interfaces with Figma: Learn to Design User Interfaces with Figma*. Birmingham: Packt Publishing.
- PERRIN, R. (2012). *Pocket Guide to APA Style* (Vol. 4). (C. Learning, Ed.) Boston, USA: Wadsworth.
- Postes de la Torre, J. A. (2010). *“DISEÑO Y ELABORACIÓN DE UN SISTEMA DE INFORMACIÓN PARA EL ANÁLISIS Y CONTROL DE HISTORIAS CLÍNICAS PARA LA CLÍNICA MATERNIDAD PADRE LUIS VARIARA*. Playas: UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA.
- Pressman, R. S. (2010). *INGENIERÍA DEL SOFTWARE 7ED, UN ENFOQUE PRÁCTICO*. México: McGraw-Hill.
- Quintana, G. (2014). *Aprende SQL*. España: Universitat Jaume I. Servei de Comunicació i Publicacions.
- Ramos, C. (2020). *Los alcances de una investigación*. Ecuador: CienciAmérica.
- Repullo Labrador, J. (2012). *Sistemas y servicios sanitarios*. España: Ediciones Díaz de Santos, S.A.
- Salvador Oliván, J. A. (1997). *Sistemas de informacion hospitalarios*. el C.M.B.D.
- Sánchez Rodríguez, H. A. (2009). *Diseño e implantación de sistemas de información y procesamiento de datos para empresas*. El Cid Editor | apuntes.
- Sánchez Rodríguez, H. A. (2009). *Diseño e implantación de sistemas de información y procesamiento de datos para empresas*. Santa Fe: El Cid Editor.
- Sarasa, A. (2016). *Introducción a las bases de datos NoSQL usando MongoDB*. España: Editorial UOC.
- Silva , W. (2018). *Bases de datos en SQL server*. Bogotá: Ecoe Ediciones.
- Singh, R. &. (2021). *Artificial Intelligence in Software Testing: Automation Tools and Techniques*. Berlin: Springer.
- Sommerville, I. (2011). *Software Engineering*. Boston: Addison-Wesley.
- Toharia Rabasco, J. P. (2015). *Lenguajes de marcas y sistemas de gestión de la información*. RA-MA Editorial.

BIBLIOGRAFÍA

- van Lamsweerde, A. (2014). *Requirements Engineering: From System Goals to UML Models to Software Specifications*. Reino Unido: Wiley.
- Vasconcelos Santillán, J. (2016). *Tecnologías de la información: (2 ed.)*. Grupo Editorial Patria.
- Welling, L. &. (2008). *PHP and MySQL Web Development*. Boston: Addison-Wesley.
- Zofío Jiménez, J. (2013). *Aplicaciones web*. España: Macmillan Iberia, S.A.

WEBLOGRAFÍA

OasisCom. (5 de abril de 2023). *Linkedin*. Obtenido de <https://es.linkedin.com/pulse/rips-qu%C3%A9-son-los-y-c%C3%B3mo-se-generan-oasiscom-cloud>

Borovskoy, D. (10 de octubre de 2023). *Linkedin*. Recuperado el 6 de marzo de 2024, de <https://www.linkedin.com/pulse/desarrollo-de-sistema-web-per%C3%BA-denis-borovskoy/?originalSubdomain=es>

Google Maps, B. (2 de Marzo de 2024). *Direccion de Google Maps Clinica Dermatologica "DERM-CAM"*. Obtenido de <https://www.google.com/maps/place/Clinica+Dermatologica+%22DERM-CAM%22/@-17.3922116,-66.1540827,18z/data=!4m6!3m5!1s0x93e373000f7bd427:0xe00dc59e897ffe13!8m2!3d-17.3919117!4d-66.15438!16s%2Fg%2F11vt89g93l?entry=ttu>

toximed. (16 de abril de 2017). *¿QUE ES GESTIÓN DE LA INFORMACIÓN?* Recuperado el 6 de marzo de 2024, de <https://instituciones.sld.cu/toximed/2017/04/16/que-es-gestion-de-la-informacion/>

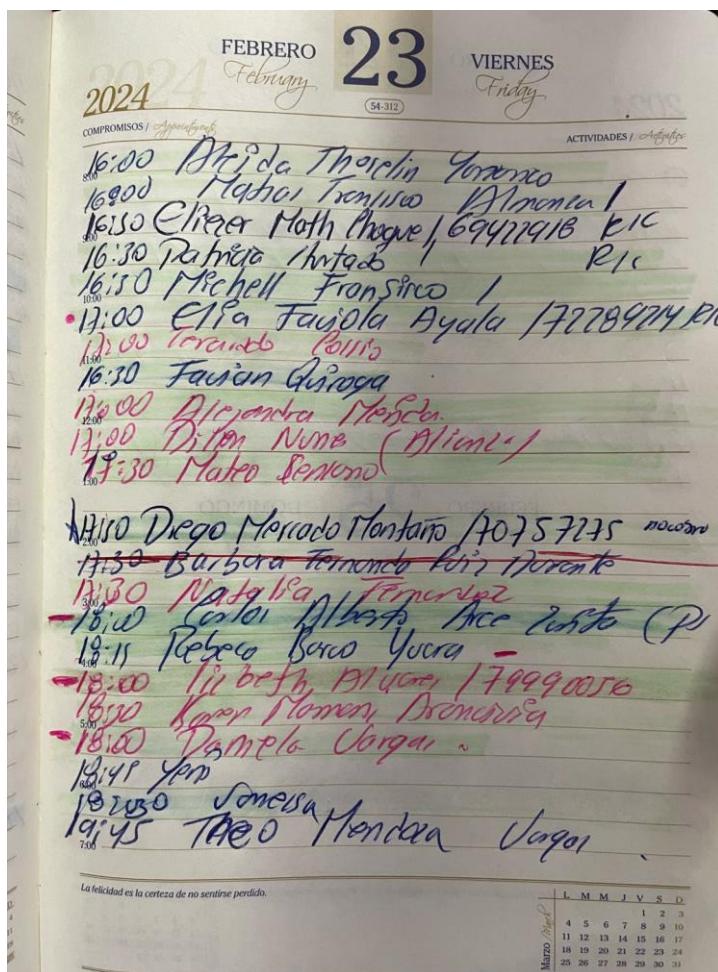
Safwana. (21 de Agosto de 2017). *bootstrapdash*. Recuperado el 26 de Junio de 2024, de <https://www.bootstrapdash.com/blog/use-bootstrap-with-php>

ANEXOS

Anexo I:

Situación Problemática

ILUSTRACIÓN N° 73: PROCESOS MANUALES EN AGENDA DE CITAS.



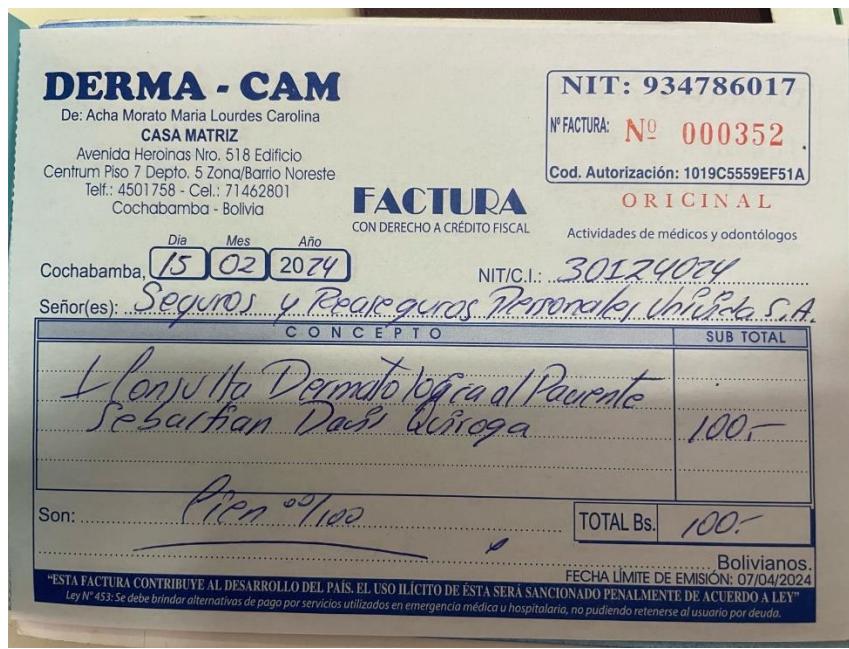
FUENTE: Elaboración propia, en base a problemática de la clínica.

ILUSTRACIÓN N° 74: PROCESOS MANUALES EN REGISTROS

Materia:	Fecha:
<u>Miercoles 31 de Enero 2024</u>	
Katherine Crespo	Pabellon Anexo -
Tatiana Muguerza Pastero	Pabellon 200 79377253
Sonia Tarea Balderrama Guzman	S.Nacion/ 35 77439768
Matthew Villacorta (Qe)	S.Nacion/ 35 77439768
Vicente Flores Flollo	QR S.Alonzo 42 70726889
querer querachí Capela	Instituto 150 78548120
Rodolfo Bustamante Vergara	Pabellon 200 68518771
Eliana Concepcion Sandoval Gómez	S.I.BIA 70 73589772 Qe
Christian Probencio Salazar (Qe)	S.Nacion/ 40 77690509
Yulio Asturias guzman Corbacho	Pabellon 80 79746908
Millon Mercado Frinande	Pabellon 200 65751209
Yisel Tenazas Juanto	Pabellon 200 77746998
Aldia Valenzuela Valenzuela	Pabellon 80 79957041
Katherine Rivero Vafeneque	Pabellon 180 79957041
Raul Fito Condor	Pabellon 200 79377973
17 Pacientes en total (Qe)	
5 Pac do 200 = 1000	1 Pac do 70 = 70
1 Pac do 180 = 180	4 Pac do 40 = 80
1 Pac do 150 = 150	1 Pac do 42 = 42
2 Pac do 80 = 160	2 Pac do 35 = 35
	7 Pac do 70
	220

FUENTE: Elaboración propia, en base a problemática de la clínica.

ILUSTRACIÓN N° 75: PROCESOS MANUALES EN LA FACTURACION

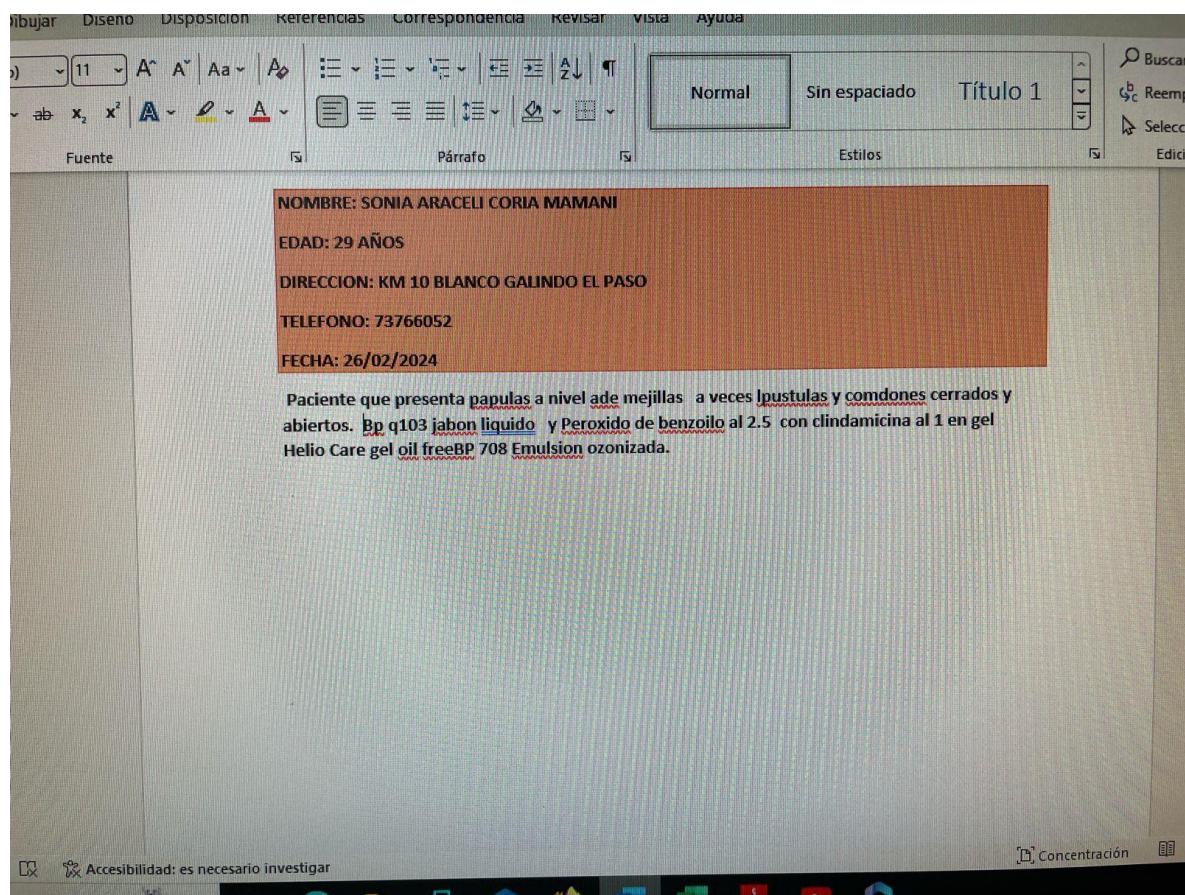


FUENTE: Elaboración propia, en base a problemática de la clínica.

ILUSTRACIÓN N° 76: DOCUMENTACION DUPLICADA.

Herramientas de búsqueda				JOSE CARLOS - Resultados de la búsqueda en historias		
				C:\Users\Acha\Escritorio\historias\HISTORIAS 20... S ALIANZA NOMBRE: JOSE CARLOS BARRIGA ARNEZ...	Tamaño: 16.3 KB	Autores: Acha
				C:\Users\Acha\Escritorio\historias\PACIENTES DE... S ALIANZA NOMBRE: JOSE CARLOS DESTRE MERCA...	Tamaño: 21.1 KB	Autores: Acha
				C:\Users\Acha\Escritorio\historias\HISTORIAS 20... S ALIANZA NOMBRE: JOSE CARLOS DESTRE MERCA...	Tamaño: 21.1 KB	Autores: Acha
				C:\Users\Acha\Escritorio\historias\PACIENTES DE... S ALIANZA NOMBRE: JOSE CARLOS CHAMBI VILLARROEL EDAD...	Tamaño: 14.7 KB	Autores: Acha
				C:\Users\Acha\Escritorio\historias\PACIENTES DE... S ALIANZA NOMBRE: JOSE CARLOS CHAMBI VILLARROEL EDAD...	Tamaño: 14.7 KB	Autores: Acha
				C:\Users\Acha\Escritorio\historias\HISTORIAS 20... S ALIANZA NOMBRE: JOSE CARLOS CHAMBI VILLARROEL EDAD...	Tamaño: 14.7 KB	Autores: Acha
				C:\Users\Acha\Escritorio\historias\PACIENTES DE... S ALIANZA NOMBRE: JOSE CARLOS CHAMBI VILLARROEL EDAD...	Tamaño: 14.7 KB	Autores: Acha
				C:\Users\Acha\Escritorio\historias\PACIENTES DE... S ALIANZA NOMBRE: JOSE CARLOS CHAMBI VILLARROEL EDAD...	Tamaño: 14.3 KB	Autores: Acha
				C:\Users\Acha\Escritorio\historias\PACIENTES DE... S ALIANZA NOMBRE: CARLOS RIOS LAZCANO EDAD: 47 AÑOS...	Tamaño: 14.3 KB	Autores: Acha

FUENTE: Elaboración propia, en base a problemática de la clínica.

ILUSTRACIÓN N° 77: ERRORES EN LA DOCUMENTACIÓN.

FUENTE: Elaboración propia, en base a problemática de la clínica.

Anexo II:

Entrevista a la Dueña de

la Clínica

ILUSTRACIÓN N° 78: ENTREVISTA CON LA DUEÑA DE LA CLÍNICA**ENTREVISTA PARA LA TOMA DE REQUERIMIENTOS**

Entrevista con la Dra. Carolina Achá Morató, Dueña de la Clínica Dermatológica DERMA-CAM

Entrevistador: Buenos días, Dra. Achá Morató. Gracias por tomarse el tiempo para esta entrevista. Para comenzar, ¿podría describirnos brevemente la historia y misión de la Clínica Dermatológica DERMA-CAM?

Dra. Achá Morató: Buenos días. Claro, la Clínica Dermatológica DERMA-CAM fue fundada hace 15 años con el objetivo de ofrecer servicios dermatológicos de alta calidad en la ciudad de Cochabamba. Nuestra misión es brindar atención especializada en el diagnóstico, tratamiento y prevención de enfermedades de la piel, siempre enfocándonos en el bienestar y la satisfacción de nuestros pacientes.

Entrevistador: Excelente. En cuanto al sistema web que estamos desarrollando, ¿cuáles son los principales problemas o necesidades que espera resolver con este nuevo sistema?

Dra. Achá Morató: El principal problema que queremos abordar es la gestión y organización de la información de nuestros pacientes y el personal de la clínica. Actualmente, utilizamos un sistema manual que resulta ineficiente y propenso a errores. Necesitamos un sistema que nos permita digitalizar y centralizar la información para mejorar la eficiencia y la accesibilidad de los datos.

Entrevistador: Entiendo. ¿Podría especificar qué funcionalidades considera esenciales para este sistema web?

Dra. Achá Morató: Por supuesto. Algunas de las funcionalidades esenciales serían:

Gestión de Pacientes: Un módulo donde podamos registrar, actualizar y consultar la información de los pacientes, incluyendo sus datos personales, historial médico, y citas.

Agenda de Citas: Una funcionalidad que permita a los pacientes programar y gestionar sus citas en línea, y que nuestro personal pueda visualizar y organizar dichas citas.

Gestión de Personal: Un módulo para administrar la información del personal de la clínica, incluyendo médicos, asistentes y otro personal administrativo.

Historial Clínico: Un sistema para registrar y acceder al historial clínico de los pacientes, incluyendo diagnósticos, tratamientos y recetas médicas.

Facturación: Un módulo que permita la generación de facturas, asegurando que todos los servicios prestados estén debidamente facturados.

Reportes: Herramientas para generar reportes sobre diversos aspectos del funcionamiento de la clínica, como número de citas, ingresos, entre otros.

FUENTE: Elaboración propia, en base a la entrevista realizada a la dueña de la clínica.

ILUSTRACIÓN N° 79: ENTREVISTA CON LA DUEÑA DE LA CLÍNICA

Entrevistador: Perfecto. En cuanto a los actores que interactuarán con el sistema, mencionó al personal de la clínica y a los pacientes. ¿Podría detallar cómo cada uno de estos actores utilizaría el sistema?

Dra. Achá Morató: Claro, los pacientes utilizarán el sistema principalmente para:

Registrar y actualizar su información personal.

Programar, cancelar y consultar citas médicas.

Acceder a su historial médico y ver los tratamientos recibidos.

Realizar pagos en línea y consultar facturas.

El personal de la clínica, por otro lado, tendrá roles más específicos:

Médicos: Podrán acceder y actualizar la información médica de los pacientes, registrar diagnósticos y tratamientos, y gestionar sus agendas de citas.

Asistentes: Ayudarán en la gestión de citas, registro de nuevos pacientes, y actualización de la información del paciente.

Personal Administrativo: Gestionará las finanzas de la clínica, generará reportes, y supervisará el correcto funcionamiento del sistema.

Entrevistador: Entendido. Para finalizar, ¿hay algún otro aspecto que considere importante incluir en el sistema o algún otro requerimiento específico?

Dra. Achá Morató: Solo enfatizar la importancia de la usabilidad. El sistema debe ser intuitivo y fácil de usar tanto para el personal como para los pacientes, minimizando la necesidad de formación extensa. También sería útil tener un sistema que pueda enviar recordatorios.

Entrevistador: Muchas gracias, Dra. Achá Morató, por su tiempo y por compartir todos estos detalles. Esta información será fundamental para el desarrollo del sistema.

FUENTE: Elaboración propia, en base a la entrevista realizada a la dueña de la clínica.

Anexo III:

Código del Desarrollo del Sistema

ILUSTRACIÓN N° 80: CODIGO DE REGISTRARSE

The screenshot shows the Visual Studio Code interface with the title bar 'VSC PHP (Workspace)'. The left sidebar shows a tree view with 'registrarse.php' selected. The main editor area displays the following PHP code:

```

42     <html Lang="es">
43         <body background="assets/img/fondo.jpg">
44             <div id="layoutAuthentication">
45                 <div id="layoutAuthentication_content">
46                     <main>
47                         ...
48                         ...
49                         ...
50                         ...
51                         ...
52                         ...
53                         ...
54                         ...
55                         ...
56                         ...
57                         ...
58                     </main>
59                 </div>
60             </div>
61         </body>
62     </html>
63 
```

The code continues with form fields for CI, date of birth, email, user, and password, followed by an alert message for errors and a submit button. The status bar at the bottom shows '17°C Parc nublado' and the date '8/7/2024'.

FUENTE: Elaboración propia, en base al código del sistema web.

ILUSTRACIÓN N° 81: CODIGO DE INICIAR SESIÓN

The screenshot shows the Visual Studio Code interface with the title bar 'VSC PHP (Workspace)'. The left sidebar shows a tree view with 'login.php' selected. The main editor area displays the following PHP code:

```

4 if ($_SERVER['REQUEST_METHOD'] == "POST") {
5     // Verificar en la base de pacientes
6     $sql_pacientes = "SELECT * FROM pacientes WHERE usuario = '$user' AND passw = '$pass'";
7     $resultado_pacientes = $conexion->query($sql_pacientes);
8
9     if ($resultado_usuarios->num_rows == 1) {
10         $dato = $resultado_usuarios->fetch_assoc();
11         $_SESSION['active'] = true;
12         $_SESSION['idUser'] = $dato['id_usuario'];
13         $_SESSION['nombre'] = $dato['nombre'];
14         $_SESSION['role'] = 'admin'; // Asignar rol de administrador
15         header('location: src/');
16     } elseif ($resultado_pacientes->num_rows == 1) {
17         $dato = $resultado_pacientes->fetch_assoc();
18         $_SESSION['active'] = true;
19         $_SESSION['idUser'] = $dato['id_paciente'];
20         $_SESSION['nombre'] = $dato['nombre'];
21         $_SESSION['role'] = 'paciente'; // Asignar rol de paciente
22         header('location: src/vista_paciente/index.php');
23     } else {
24         $alert = '<div class="alert alert-danger" role="alert">
25             Usuario o Contraseña Incorrecta
26         </div>';
27         session_destroy();
28     }
29 }
30 
```

The code handles POST requests to check user credentials against a database. It sets session variables for active status, user ID, name, and role based on the found user or patient. An alert message is displayed if the credentials are incorrect. The status bar at the bottom shows '17°C Parc nublado' and the date '8/7/2024'.

FUENTE: Elaboración propia, en base al código del sistema web.

ILUSTRACIÓN N° 82: CODIGO PANEL DE ADMINISTRACIÓN

```

21 <div class="row">
22   <a class="col-xl-3 col-md-6 mb-4" href="usuarios.php">
23     <div class="card border-left-primary shadow h-100 py-2 bg-warning">
24       <div class="card-body">
25         <div class="row no-gutters align-items-center">
26           <div class="text-xs font-weight-bold text-white text-uppercase mb-1">Usuarios</div>
27           <div class="h5 mb-0 font-weight-bold text-white"><?php echo $totalU; ?></div>
28         </div>
29         <div class="col-auto">
30           <i class="fas fa-user fa-2x text-gray-300"></i>
31         </div>
32       </div>
33     </div>
34   </a>
35 
36 
37 
38   <a class="col-xl-3 col-md-6 mb-4" href="pacientes.php">
39     <div class="card border-left-success shadow h-100 py-2 bg-success">
40       <div class="card-body">
41         <div class="row no-gutters align-items-center">
42           <div class="col mr-2">
43             <div class="text-xs font-weight-bold text-white text-uppercase mb-1">Pacientes</div>
44             <div class="h5 mb-0 font-weight-bold text-white"><?php echo $totalC; ?></div>
45           </div>
46           <div class="col-auto">
47             <i class="fas fa-users fa-2x text-gray-300"></i>
48           </div>
49         </div>
50       </div>
51     </div>
52   </a>
53 
```

FUENTE: Elaboración propia, en base al código del sistema web.

ILUSTRACIÓN N° 83: CODIGO GESTOR DE USUARIOS

```

47 <div id="nuevo_usuario" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="my-modal-title">
48   <div class="modal-dialog" role="document">
49     </div>
50   </div>
51 
52   <div class="table-responsive">
53     <table class="table table-hover table-striped table-bordered mt-2" id="tbl">
54       <thead class="thead-dark">
55         <tr>
56           <th>id</th>
57           <th>Nombre</th>
58           <th>Apellidos</th>
59           <th>Telefono</th>
60           <th>Email</th>
61           <th></th>
62         </tr>
63       </thead>
64       <tbody>
65         <tr>
66           <td><?php ?></td>
67           <td><?php ?></td>
68           <td><?php ?></td>
69           <td><?php ?></td>
70           <td><?php ?></td>
71           <td><?php ?></td>
72         </tr>
73       </tbody>
74     </table>
75   </div>
76 
77   <?php ?>
78   <?php
79     include ".../conexion.php";
80 
81     $query = mysqli_query($conexion, "SELECT * FROM usuarios ");
82     $result = mysqli_num_rows($query);
83     if ($result > 0) {
84       while ($data = mysqli_fetch_assoc($query)) {
85         <?php
86         <?php
87         <?php
88         <?php
89         <?php
90         <?php
91         <?php
92         <?php
93         <?php
94         <?php
95         <?php
96         <?php
97         <?php
98         <?php
99         <?php
100        <?php
101        <?php
102        <?php
103        <?php
104        <?php
105        <?php
106        <?php
107        <?php
108        <?php
109        <?php
110        <?php
111        <?php
112        <?php
113        <?php
114        <?php
115        <?php
116        <?php
117        <?php
118        <?php
119        <?php
120        <?php
121        <?php
122        <?php
123        <?php
124        <?php
125        <?php
126        <?php
127 
```

FUENTE: Elaboración propia, en base al código del sistema web.

ILUSTRACIÓN N° 84: CODIGO FORMULARIO PARA CREAR USUARIO

```

File Edit Selection View Go Run ...
VSC PHP (WORKSPACE) < - > VSC PHP (Workspace)
VSC PHP Clinica_Deramologica src vistas usuarios.php
    class="fas fa-plus"></i></button>
<div id="nuevo_usuario" class="modal fade" tabindex="-1" role="dialog" aria-labelledby="my-modal-title" aria-hidden="true">
<div class="modal-dialog" role="document">
<div class="modal-content">
<div class="modal-header bg-primary text-white">
<h5 class="modal-title" id="my-modal-title">Nuevo Usuario</h5>
<button class="close" data-dismiss="modal" aria-label="Close">&times;</span>
</button>
</div>
<div class="modal-body">
<form action="" method="post" autocomplete="off">
<?php echo isset($alert) ? $alert : '' ; ?>
<div class="form-group">
<label for="nombre">Nombre</label>
<input type="text" class="form-control" placeholder="Ingrese Nombre" name="nombre">
</div>
<div class="form-group">
<label for="apellidos">Apellidos</label>
<input type="text" class="form-control" placeholder="Ingrese Apellidos" name="ape_id="apellidos">
</div>
<div class="form-group">
<label for="telefono">Teléfono</label>
<input type="number" placeholder="Ingrese Teléfono" name="telefono" id="telefono" class="form-control">
</div>
<div class="form-group">
<label for="email">Correo Electrónico</label>
<input type="text" placeholder="Ingrese Correo Electrónico" name="email" id="email" class="form-control">
</div>
</form>
</div>
</div>
</div>
    
```

Ln 3, Col 15 Spaces: 4 UTF-8 LF PHP 8.2.12: 8.2 19:24 17°C Parc. nublado 8/7/2024

FUENTE: Elaboración propia, en base al código del sistema web.

ILUSTRACIÓN N° 85: CODIGO EDITAR USUARIO

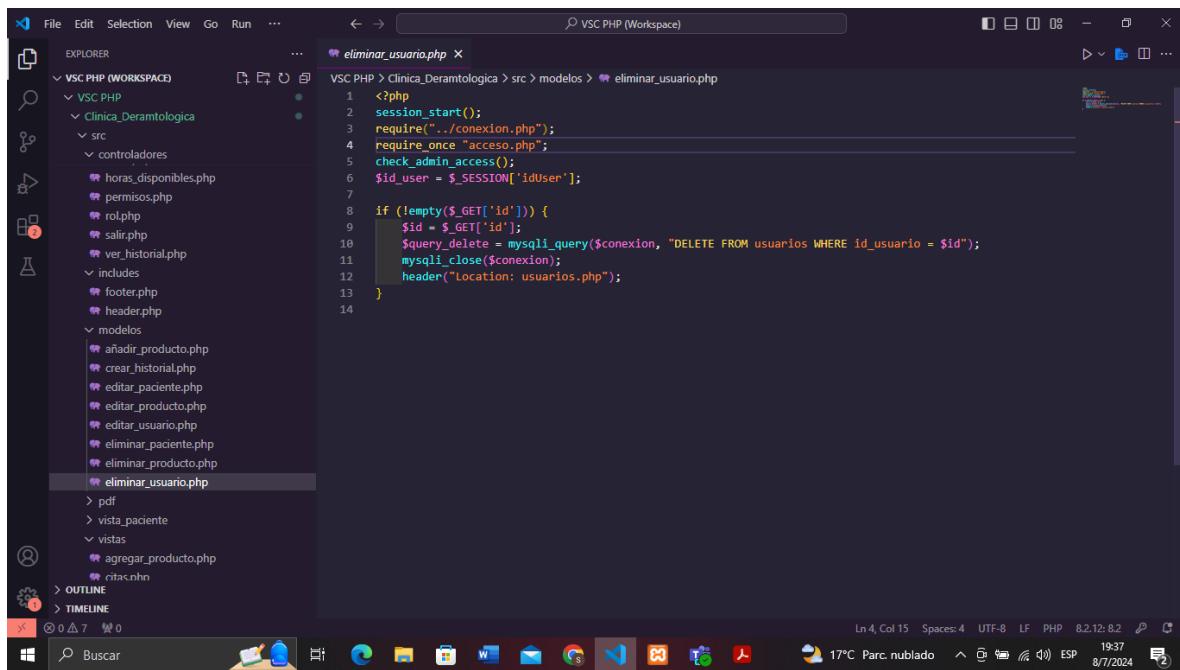
```

File Edit Selection View Go Run ...
VSC PHP (WORKSPACE) < - > VSC PHP (Workspace)
VSC PHP Clinica_Deramologica src modelos editar_usuario.php
    $id_usuario = $_REQUEST['id'];
    $sql = mysqli_query($conexion, "SELECT * FROM usuarios WHERE id_usuario = $id_usuario");
    $result_sql = mysqli_num_rows($sql);
    if ($result_sql == 0) {
        header("Location: usuarios.php");
    } else {
        if ($data = mysqli_fetch_array($sql)) {
            $id_usuario = $data['id_usuario'];
            $nombre = $data['nombre'];
            $apellidos = $data['apellidos'];
            $telefono = $data['telefono'];
            $email = $data['email'];
            $usuario = $data['usuario'];
            $passw = $data['passw'];
            $rol = $data['rol'];
        }
    }
    <div class="row">
        <div class="col-md-6 mx-auto">
            <div class="card">
                <div class="card-header bg-primary text-white">
                    Modificar Usuario
                </div>
                <div class="card-body">
                    <form class="" action="" method="post">
                        <?php echo isset($alert) ? $alert : '' ; ?>
                        <input type="hidden" name="id_usuario" value=<?php echo $id_usuario; ?>>
                        <div class="form-group">
                            <label for="nombre">Nombre</label>
                            <input type="text" placeholder="Ingrese Nombre" class="form-control" name="nombre">
                        
```

Ln 3, Col 15 Spaces: 4 UTF-8 LF PHP 8.2.12: 8.2 19:25 17°C Parc. nublado 8/7/2024

FUENTE: Elaboración propia, en base al código del sistema web.

ILUSTRACIÓN N° 86: CODIGO ELIMINAR USUARIO



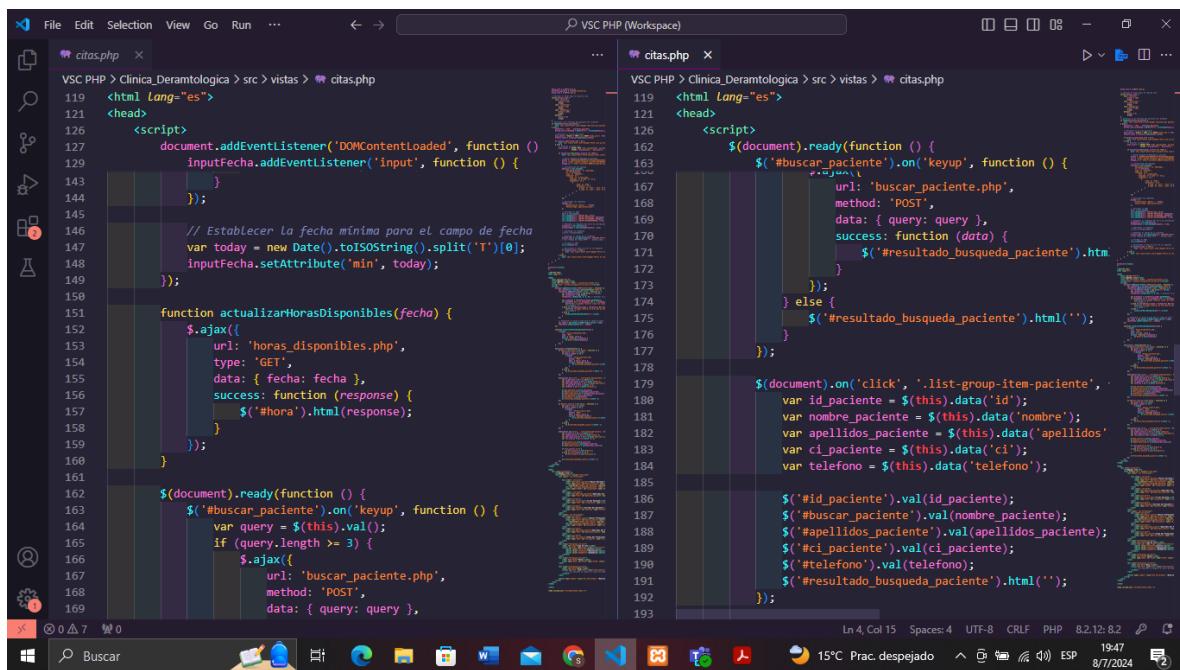
```

1 <?php
2 session_start();
3 require("../conexion.php");
4 require_once "acceso.php";
5 check_admin_access();
6 $id_user = $_SESSION['idUser'];
7
8 if (!empty($_GET['id'])) {
9     $id = $_GET['id'];
10    $query_delete = mysqli_query($conexion, "DELETE FROM usuarios WHERE id_usuario = $id");
11    mysqli_close($conexion);
12    header("Location: usuarios.php");
13 }
14

```

FUENTE: Elaboración propia, en base al código del sistema web.

ILUSTRACIÓN N° 87: CODIGO AGENDAR CITAS



```

119 <html Lang="es">
120 <head>
121     <script>
122         document.addEventListener('DOMContentLoaded', function () {
123             inputFecha.addEventListener('input', function () {
124                 // Establecer la fecha mínima para el campo de fecha
125                 var today = new Date().toISOString().split('T')[0];
126                 inputFecha.setAttribute('min', today);
127             });
128
129             function actualizarHorasDisponibles(fecha) {
130                 $.ajax({
131                     url: 'horas_disponibles.php',
132                     type: 'GET',
133                     data: { fecha: fecha },
134                     success: function (response) {
135                         $('#hora').html(response);
136                     }
137                 });
138
139             }
140
141             $(document).ready(function () {
142                 $('#buscar_paciente').on('keyup', function () {
143                     var query = $(this).val();
144                     if (query.length > 3) {
145                         $.ajax({
146                             url: 'buscar_paciente.php',
147                             method: 'POST',
148                             data: { query: query },
149                             success: function (data) {
150                                 $('#resultado_busqueda_paciente').html(data);
151                             }
152                         });
153                     }
154                 });
155             }
156         });
157     </script>
158 
```

FUENTE: Elaboración propia, en base al código del sistema web.

ILUSTRACIÓN N° 88: CODIGO LISTA DE PACIENTES

```

10 <div class="container mt-5">
11   <table class="table table-striped table-bordered" id="tbl">
12     <tbody>
13       <?php while ($row = $result->fetch_assoc()) { ?>
14         <tr>
15           <td>
16             <?php
17               if ($row['estado'] == 1) {
18                 echo "Reservado";
19               } elseif ($row['estado'] == 2) {
20                 echo "Realizado";
21               } else {
22                 echo "Cancelado";
23               }
24             </?php >
25           </td>
26           <td>
27             <?php
28               if ($row['estado'] == 1) { ?>
29               <button class="btn btn-success" onclick="crearHistorial(<?php echo $row['id_cita']; ?>)">Nuevo</button>
30               <button class="btn btn-danger" onclick="cambiarEstado(<?php echo $row['id_cita']; ?>)">Cancelar</button>
31             <?php } elseif ($row['estado'] == 2) { ?>
32               <button class="btn btn-info" onclick="verHistorial(<?php echo $row['id_cita']; ?>)">Ver Historial</button>
33             <?php } else { ?>
34               <span class="text-danger">Cancelada</span>
35             <?php } ?>
36           </td>
37         </tr>
38       </tbody>
39     </table>
40   </div>

```

FUENTE: Elaboración propia, en base al código del sistema web.

ILUSTRACIÓN N° 89: CODIGO CREAR HISTORIAL DE CITA

```

5 if ($_SERVER["REQUEST_METHOD"] == "POST") {
6   // Insertar el historial
7   $query_historial = "INSERT INTO historial (id_cita, diagnostico, tratamiento)
8   VALUES ('$id_cita', '$diagnostico', '$tratamiento')";
9   if ($conexion->query($query_historial) === TRUE) {
10     // Actualizar el estado de la cita a "En curso"
11     $query_update = "UPDATE citas SET estado = '2' WHERE id_cita = '$id_cita'";
12     $conexion->query($query_update);
13
14     echo '<div class="alert alert-success" role="alert">Historial y tratamiento creados exitosamente.</div>';
15     header('Location: listar_citas.php');
16     exit();
17   } else {
18     echo '<div class="alert alert-danger" role="alert">Error al crear el historial: ' . $conexion->error . '</div>';
19   }
20
21   $conexion->close();
22 }
23
24 <div class="container mt-5">
25   <h2>Crear Historial</h2>
26   <form method="post" action="crear_historial.php">
27     <div class="form-group">
28       <label for="diagnostico">Diagnóstico:</label>
29       <textarea id="diagnostico" name="diagnostico" class="form-control required"></textarea>
30     </div>
31     <div class="form-group">
32       <label for="tratamiento">Tratamiento:</label>
33       <textarea id="tratamiento" name="tratamiento" class="form-control required"></textarea>
34     </div>
35     <input type="hidden" name="id_cita" value=<?php echo $_GET['id_cita']; ?>>">
36   </form>
37 </div>

```

FUENTE: Elaboración propia, en base al código del sistema web.

ILUSTRACIÓN N° 90: CÓDIGO VER HISTORIAL

The screenshot shows the Visual Studio Code interface with the file 'ver_historial.php' open in the editor. The code is written in PHP and displays a patient's medical history. It includes error handling for missing appointment IDs and displays details like date, doctor, patient name, diagnosis, and treatment.

```

if (isset($_GET['id_cita'])) {
    $resultado = $conexion->query($query);
    $historial = $resultado->fetch_assoc();

    if (!$historial) {
        echo '<div class="alert alert-danger" role="alert">No se encontró el historial para esta cita.</div>';
        exit();
    } else {
        echo '<div class="alert alert-danger" role="alert">ID de cita no proporcionado.</div>';
        exit();
    }
}

$conexion->close();
}

<div class="container mt-5">
    <h2>Ver Historial</h2>
    <div class="card">
        <div class="card-header bg-primary text-white">
            Detalles de la Cita
        </div>
        <div class="card-body">
            <p><strong>Fecha de la Cita:</strong> <?php echo $historial['fecha_cita']; ?></p>
            <p><strong>Doctora:</strong> <?php echo $historial['usuario_nombre'] . ' ' . $historial['usuai']; ?></p>
            <p><strong>Paciente:</strong> <?php echo $historial['paciente_nombre'] . ' ' . $historial['pa'; ?></p>
            <p><strong>Diagnóstico:</strong> <?php echo $historial['diagnostico']; ?></p>
            <p><strong>Tratamiento:</strong> <?php echo $historial['tratamiento']; ?></p>
        </div>
    </div>
</div>

```

FUENTE: Elaboración propia, en base al código del sistema web.

ILUSTRACIÓN N° 91: CÓDIGO GERNERAR FACTURA PARA CITAS

The screenshot shows the Visual Studio Code interface with the file 'generar_cita.php' open in the editor. The code generates a PDF invoice for an appointment. It uses the FPDF library and MySQL queries to fetch appointment and patient data.

```

require_once 'fpdf/fpdf.php';

// Verificar que el archivo fpdf.php exista
if (!file_exists('fpdf/fpdf.php')) {
    die('Error: No se encontró el archivo fpdf.php.');
}

// Obtener el ID de la cita y el ID del paciente desde la URL
$id_cita = $_GET['v'];
$id_paciente = $_GET['c'];

// Crear un nuevo documento PDF
$pdf = new FPDF('P', 'mm', 'letter');
$pdf->AddPage();
$pdf->SetMargins(10, 10, 10);
$pdf->SetTitle("Factura");
$pdf->SetFont('Arial', 'B', 12);

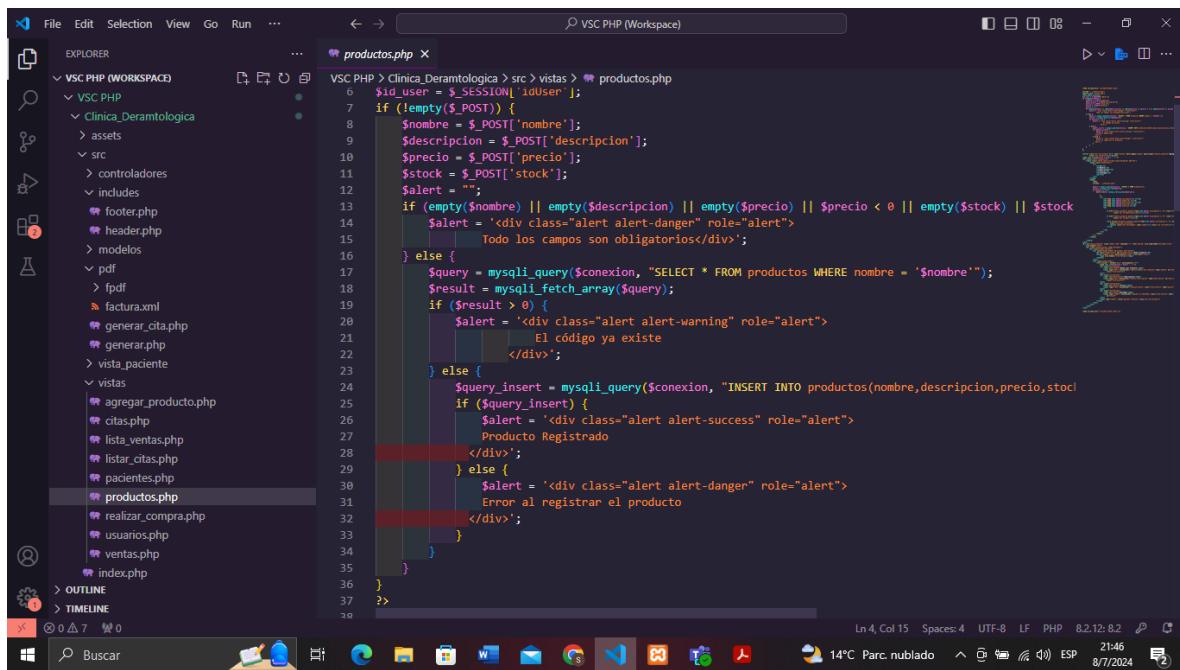
// Obtener datos de la configuración de la clínica
$config = mysqli_query($conexion, "SELECT * FROM configuracion");
if (!$config) {
    die('Error en la consulta de configuración: ' . mysqli_error($conexion));
}
$datos = mysqli_fetch_assoc($config);

// Obtener datos del paciente
$pacientes = mysqli_query($conexion, "SELECT * FROM pacientes WHERE id_paciente = $id_paciente");
if (!$pacientes) {
    die('Error en la consulta de pacientes: ' . mysqli_error($conexion));
}
$datosC = mysqli_fetch_assoc($pacientes);

```

FUENTE: Elaboración propia, en base al código del sistema web.

ILUSTRACIÓN N° 92: CÓDIGO DE GESTOR DE PRODUCTOS



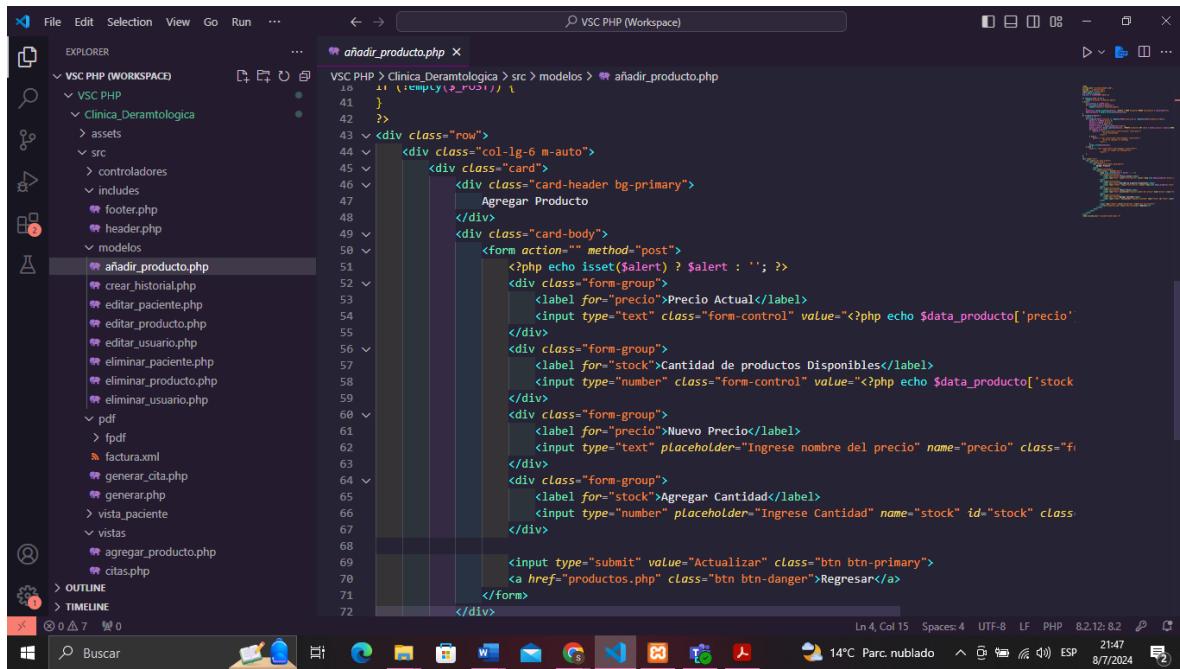
```

File Edit Selection View Go Run ...
products.php x
VSC PHP (Workspace)
VSC PHP
Clinica_Deramologica
assets
src
controladores
includes
footer.php
header.php
modelos
pdf
fpdf
factura.xml
generar_cita.php
generar.php
vista_paciente
vistas
agregar_producto.php
citas.php
lista_ventas.php
listar_citas.php
pacientes.php
productos.php
realizar_compra.php
usuarios.php
ventas.php
index.php
OUTLINE
TIMELINE
Buscar
14°C Parc nublado 21:46
8/7/2024
Ln 4, Col 15 Spaces: 4 UTF-8 LF PHP 8.2.12: 8.2
if (!empty($_POST)) {
    $id_user = $_SESSION['idUser'];
    $nombre = $_POST['nombre'];
    $descripcion = $_POST['descripcion'];
    $precio = $_POST['precio'];
    $stock = $_POST['stock'];
    $alert = "";
    if (empty($nombre) || empty($descripcion) || empty($precio) || $precio < 0 || empty($stock) || $stock < 0) {
        $alert = '<div class="alert alert-danger" role="alert">
                    Todo los campos son obligatorios</div>';
    } else {
        $query = mysqli_query($conexion, "SELECT * FROM productos WHERE nombre = '$nombre'");
        $result = mysqli_fetch_array($query);
        if ($result > 0) {
            $alert = '<div class="alert alert-warning" role="alert">
                        El código ya existe
                    </div>';
        } else {
            $query_insert = mysqli_query($conexion, "INSERT INTO productos(nombre,descripcion,precio,stock) VALUES('$nombre','$descripcion','$precio','$stock')");
            if ($query_insert) {
                $alert = '<div class="alert alert-success" role="alert">
                            Producto Registrado
                        </div>';
            } else {
                $alert = '<div class="alert alert-danger" role="alert">
                            Error al registrar el producto
                        </div>';
            }
        }
    }
}

```

FUENTE: Elaboración propia, en base al código del sistema web.

ILUSTRACIÓN N° 93: CODIGO DE AGREGAR STOCK



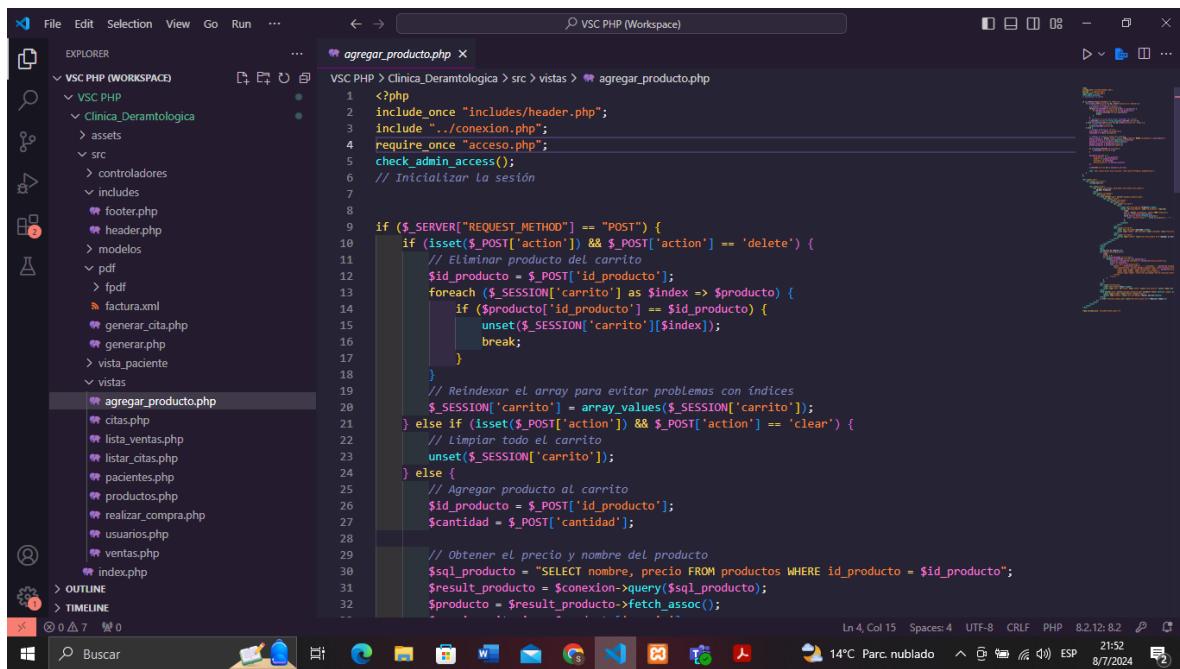
```

File Edit Selection View Go Run ...
añadir_producto.php x
VSC PHP (Workspace)
VSC PHP
Clinica_Deramologica
assets
src
controladores
includes
footer.php
header.php
modelos
añadir_producto.php
crear_historial.php
editar_paciente.php
editar_producto.php
editar_usuario.php
eliminar_paciente.php
eliminar_producto.php
eliminar_usuario.php
pdf
fpdf
factura.xml
generar_cita.php
generar.php
vista_paciente
vistas
agregar_producto.php
citas.php
OUTLINE
TIMELINE
Buscar
14°C Parc nublado 21:47
8/7/2024
Ln 4, Col 15 Spaces: 4 UTF-8 LF PHP 8.2.12: 8.2
if (!empty($_POST)) {
    $alert = '';
    if (empty($_POST['precio'])) {
        $alert = '<div class="alert alert-danger" role="alert">
                    Precio Actual
                </div>';
    } else {
        $data_producto['precio'] = $_POST['precio'];
    }
    if (empty($_POST['stock'])) {
        $alert = '<div class="alert alert-danger" role="alert">
                    Cantidad de productos Disponibles
                </div>';
    } else {
        $data_producto['stock'] = $_POST['stock'];
    }
    if (!empty($alert)) {
        echo $alert;
    } else {
        $query_update = mysqli_query($conexion, "UPDATE productos SET precio = '$data_producto[precio]', stock = '$data_producto[stock]' WHERE id = '$id'");
        if ($query_update) {
            $alert = '<div class="alert alert-success" role="alert">
                            Nuevo Precio
                        </div>';
        } else {
            $alert = '<div class="alert alert-danger" role="alert">
                            Error al actualizar el producto
                        </div>';
        }
    }
}

```

FUENTE: Elaboración propia, en base al código del sistema web.

ILUSTRACIÓN N° 94: CÓDIGO DE CARRITO DE COMPRAS



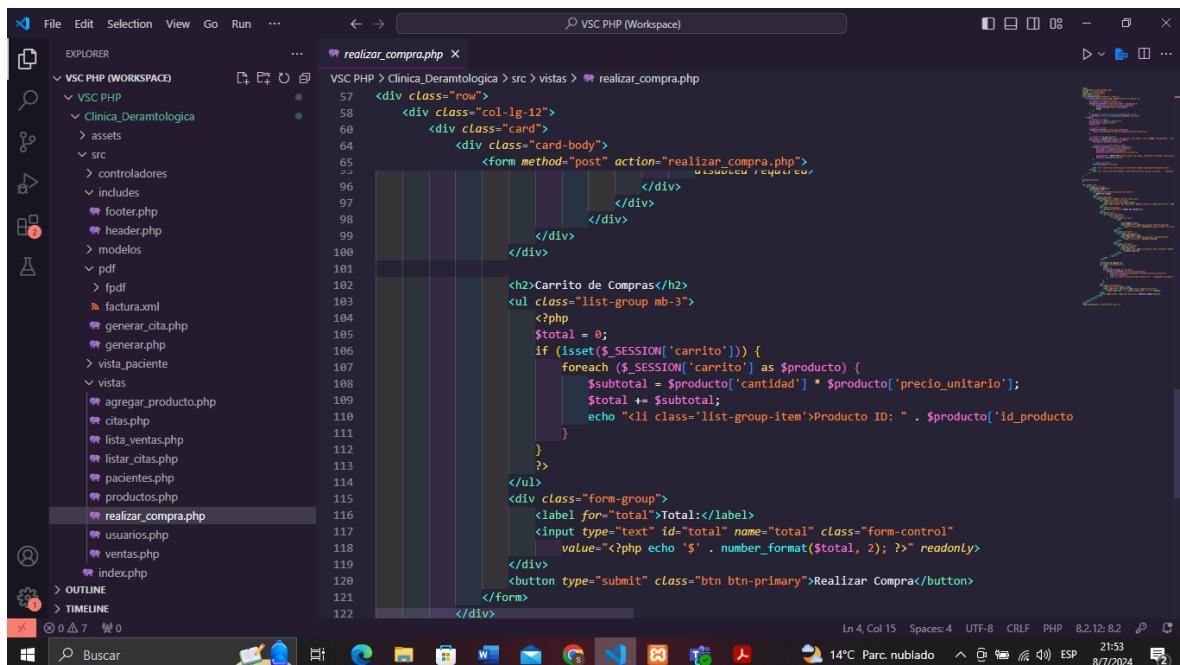
```

1 <?php
2 include_once "includes/header.php";
3 include "../conexion.php";
4 require_once "acceso.php";
5 check_admin_access();
6 // Inicializar la sesión
7
8
9 if ($_SERVER["REQUEST_METHOD"] == "POST") {
10    if (isset($_POST['action']) && $_POST['action'] == 'delete') {
11        // Eliminar producto del carrito
12        $id_producto = $_POST['id_producto'];
13        foreach ($_SESSION['carrito'] as $index => $producto) {
14            if ($producto['id_producto'] == $id_producto) {
15                unset($_SESSION['carrito'][$index]);
16                break;
17            }
18        }
19        // Reindexar el array para evitar problemas con índices
20        $_SESSION['carrito'] = array_values($_SESSION['carrito']);
21    } else if (isset($_POST['action']) && $_POST['action'] == 'clear') {
22        // Limpiar todo el carrito
23        unset($_SESSION['carrito']);
24    } else {
25        // Agregar producto al carrito
26        $id_producto = $_POST['id_producto'];
27        $cantidad = $_POST['cantidad'];
28
29        // Obtener el precio y nombre del producto
30        $sql_producto = "SELECT nombre, precio FROM productos WHERE id_producto = $id_producto";
31        $result_producto = $conexion->query($sql_producto);
32        $producto = $result_producto->fetch_assoc();

```

FUENTE: Elaboración propia, en base al código del sistema web.

ILUSTRACIÓN N° 95: CÓDIGO DE REALIZAR COMPRA

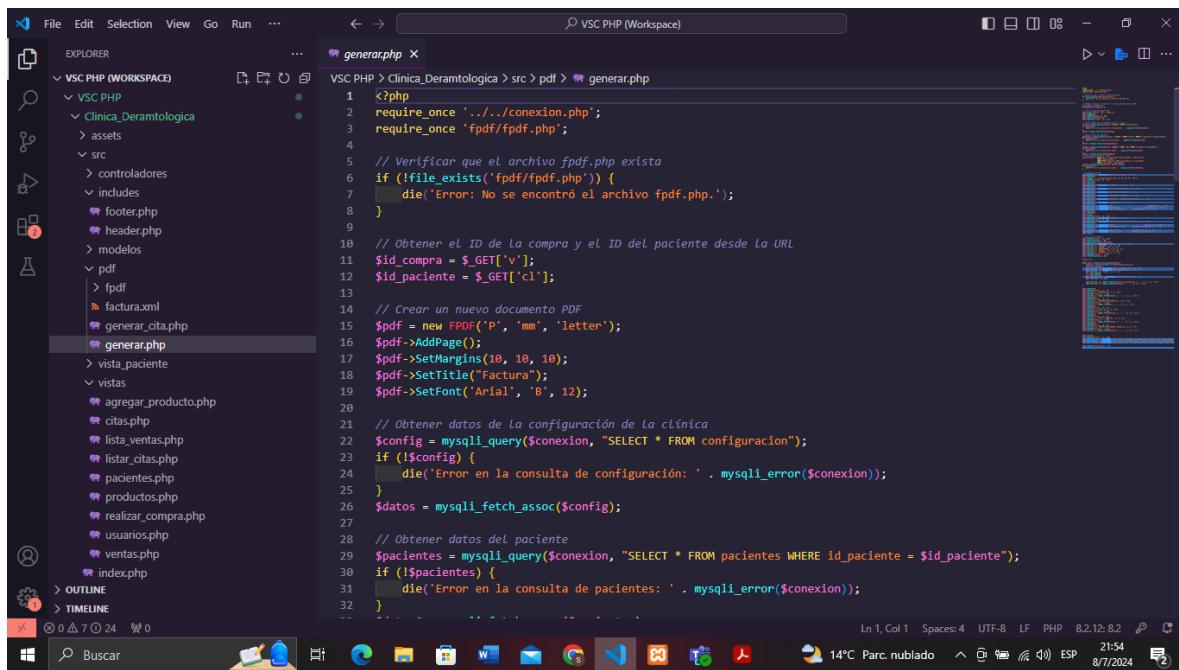


```

57 <div class="row">
58     <div class="col-lg-12">
59         <div class="card">
60             <div class="card-body">
61                 <form method="post" action="realizar_compra.php">
62                     <div>
63                         <div>
64                             <div>
65                                 <div>
66                                     <div>
67                                         <div>
68                                             <div>
69                                                 <div>
70                                                     <div>
71                                                         <div>
72                                                             <div>
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122

```

FUENTE: Elaboración propia, en base al código del sistema web.

ILUSTRACIÓN N° 96: CÓDIGO DE FACTURA DE PRODUCTOS

The screenshot shows the Visual Studio Code (VS Code) interface with the following details:

- File Path:** VSC PHP > Clínica_Dermatologica > src > pdf > generar.php
- Code Content:** The code is written in PHP and uses the FPDF library to generate a PDF. It includes logic to check if the FPDF file exists, set margins, and retrieve data from a MySQL database to fill the PDF template.
- Explorer View:** Shows the project structure under "VSC PHP (WORKSPACE)" with files like factura.xml, generacion.php, and various PHP scripts for patient management.
- Status Bar:** Displays system information such as temperature (14°C), battery level (Parc. nublado), and date/time (8/7/2024).

FUENTE: Elaboración propia, en base al código del sistema web.